Proceedings
of the 1st International Conference on
Robotics in Education

Editors: Peter Hubinský and Richard Balogh

Sept. 16 – 17, 2010
Bratislava, Slovakia

Published by:
Slovak University of Technology in Bratislava
http://rie2010.stuba.sk

ISBN 978-80-227-3353-3

Credits:
Cover design: David Jablonovský
Cover photo: Peter Tomcsányi
LaTeX editor: Richard Balogh
using LaTeX's 'confproc' class

Printed by Tribun EU, Ltd. (http://www.librix.eu)
September 2010

International Programme Committee

Chair: Peter Hubinský, Slovak University of Technology in Bratislava, Slovakia

Members:

- José Carlos Alves, University of Porto, Portugal
- Simon Coupland, De Montfort University in Leicester, United Kingdom
- Gerhard Gruhler, Heilbronn University, Germany
- Robert John, De Montfort University in Leicester, United Kingdom
- Ladislav Jurišica, Slovak University of Technology in Bratislava
- Ivan Kalaš, Comenius University in Bratislava, Slovakia
- Dariusz Marchewka, AGH-University of Science and Technology, Kraków, Poland
- Martin Mellado, Universidad Politécnica de Valencia, Spain
- Fritz Schmöllebeck, FH Technikum Wien, Austria
- Gerald Steinbauer, Graz University of Technology, Austria
- František Šolc, Brno University of Technology, Czech Republic
- Osman Tokhi, University of Sheffield, United Kingdom
- Gurvinder S. Virk, Massey University Wellington, New Zealand
- Anton Vitko, Slovak University of Technology in Bratislava

International Organizing Committee

- Richard Balogh (SK)
- Pavel Petrovič (SK)
- Lukáš Palkovič (SK)
- František Duchoň (SK)
- Alexander Hofmann (AT)
- Roland Stelzer (AT)
- Karim Jafarmadar (AT)

List of Reviewers

- Richard Balogh (SK)
- Simon Coupland (UK)
- František Duchoň (SK)
- Gerhard Gruhler (DE)
- Peter Hubinský (SK)
- Štefan Chamraz (SK)
- Ľubomír Janovický (SK)
- Robert John (UK)
- Ladislav Jurišica (SK)
- Marián Kľúčik (SK)
- Václav Králík (SK)
- Andrej Lúčny (SK)
- Dariusz Marchewka (PL)
- Martin Mellado (ES)
- Roman Murár (IR)
- David Obdržálek (CZ)
- Lukáš Palkovič (SK)
- Janka Pekárová (SK)
- Pavel Petrovič (SK)
- Boris Rohaľ-Ilkiv (SK)
- Gerald Steinbauer (AT)
- Roland Stelzer (AT)
- František Šolc (CZ)
- Anton Vitko (SK)

Welcome from Peter Hubinský, Conference Chair

Ladies and gentlemen,

it is my pleasure to welcome you at the 1st International Conference on Robotics in Education which is held under the patronage of the Institute of Control and Industrial Informatics at the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava in cooperation with other Slovak and Austrian companies and institutions.

This conference is a part of the Slovak-Austrian cross-border project CENTROBOT, which is funded by the EFRE program *Creating the future*. Organizers would also like to appreciate the help and support of sponsors of the conference.

International character of the conference is underlined by the fact that members of programme committee are from ten different countries and participants are coming from over dozen different countries from all over the world. We would like to express our thanks all those who submitted papers and reviewers for their responsible work.

Especially we are appreciating invited lecturers namely Ms. Jessica Uelmen from USA, Dr. Ansgar Bredenfeld from Germany and Mr. Peter Ducháček from the ABB Robotic training centre in Slovakia.

Proceedings of this conference are published on the CD-ROM and also some printed booklets are available in request. Selected contributions will be published also in the magazine ATP Journal Plus.

The 1st international conference on Robotics in Education offers you also some joined events. You are warmly invited to the final presentation of the project Centrobot in Vienna, you can view beautiful robots on the accompanying exhibition and for those, who are interested about the outdoor mobile robotic contest: we host also Robotour – robotika.cz outdoor delivery challenge.

I wish you a pleasant stay here in Bratislava and I am looking forward to see you in our comprehensive sessions.

Peter Hubinský

Conference Programme

Day 1: September 16, 2010

Invited lectures

Robotics in Education

Robotic Contests

Hardware for Robotics

Day 2: September 17, 2010

Invited lectures

Robotics in Education

The Roberta® Initiative

Dr. Ansgar Bredenfeld, Thorsten Leimbach
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS
53754 Sankt Augustin, Germany
roberta-zentrale@iais.fraunhofer.de

Abstract—**This paper gives a short overview of the Roberta Initiative – an approach to raise especially but not only girls´ interest in STEM (science, technology, engineering and math). Roberta comes with a didactic concept that uses robot construction kits in combination with a specialized didactic material and course format. Roberta teacher trainings and a European dissemination network are integral parts of the Roberta Initiative in order to establish a sustainable activity to raise girls´ interest in technical topics and in the end the portion of female engineers in Europe.**

Keywords-component; educational robotics, robot construction kits, robot courses, teacher training

I. Introduction

By designing, constructing, and programming robots, children can experience that working with technology is a creative and interesting but not a trivial process. The resulting hands-on learning environment helps them to acquire knowledge in computer science, technology and engineering. Additionally, constructing and programming robots in a teamwork setting is an ideal instrument to train those types of competences and soft skills that are essential for dealing with technical development processes. Many educational robotics activities - robot courses or robot competitions - rely on the fascination of mobile robots.

With Roberta the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) addresses the lack of (female) engineers in Germany and other European countries by raising children's´ interest in technical professions. Roberta uses robot courses as a creative learning environment to teach knowledge in computer science, technology and engineering in an integrated, holistic way. The robot courses are tailored in a gender-balanced way, i. e. the didactic approach selects themes and experiments that are more interesting for girls but do not exclude boys. This is the specific approach and strength of the Roberta Initiative in comparison to other educational robotics offerings.

The Roberta Initiative comprises several elements which in combination constitute the basis for a sustainable activity to raise girls´ interest in technical topics.

- *Gendered didactic material* as a resource based on which certified teachers can design and assemble robot courses
- The *Roberta teacher trainings* as hands-on introductions to the employment of the Roberta didactic material. In addition, they are the entry to join the Roberta teacher network.
- A *network of Roberta Regional Centres* to promote the ideas of Roberta on a regional scale and to provide certified Roberta teachers and Roberta courses in a region.

The following sections give an overview on these constitutive elements of the Roberta Initiative.

II. Roberta Didactic Material

In Roberta courses children first learn to get familiar with the building and programming of robots (simple task). The next step is to learn how to use different kind of sensors and programming languages (compound tasks). Simple tasks and compound tasks impart basic knowledge to construct, program and test a robot. Roberta experiments are based on the knowledge gained from Roberta simple tasks and compound tasks.

The structure of a Roberta experiment is as follows: After an introduction and explanation to a real world theme, usually taken from biology or nature, the concepts and structure observed in the real world are abstracted to a robotics experiment. Thus, the course participants have to analyze and understand a real world phenomenon. The next task is to model the phenomenon and to map it to a robotic experiment. This requires to analyze and to really understand a theme and its phenomenon. It is essentially a research step that is implicitly performed by the Roberta course participants. The intention behind is, that the course participants get a deeper understanding of a system and not only of a small part of it. Teaching to think on system level is a key motivation for our approach. Since Roberta courses are performed in a team environment, the course setting may be regarded as a simulated research and engineering process.

An example from the Roberta material is the experiment »Dance of the Bees«. The Roberta material gives general information on the theme and suggests experiments. An example for the theme »Dance of the Bees« is to understand and subsequently model the behavior of the bees in different situations. A first experiment suggested is to build a robot that implements a bee dance for nectar collection. The next step is to develop a robot bee that implements a behavior to guard the beehive. The Roberta didactic material delivers ideas to develop further experiments for a given theme. The course participants are encouraged to develop and realize their own ideas using the knowledge they gained from the simple and compound tasks and the Roberta experiments suggested in the material.

Roberta themes and experiments concentrate in particular on nature and biology. Other examples presented in the Roberta material are Gaits (two legged, six legged), Maze and Ants (construction of an ant, ant trail). These themes are definitely more appealing and attractive for girls than for example soccer robots or fast driving vehicles.

An important concept of the organization and structure of the didactic material is a clear-cut separation of the experiments from their concrete implementation using a specific robot construction kit. The didactic approach is deliberately independent of a concrete robotic product. It can be adapted to new construction kits appearing on the market. A suitable robot construction kit has of course to provide functionality like actuators, sensors, programmable control and robot communication. At present, Roberta tasks and experiments have been adopted to the LEGO Mindstorms construction kits RCX and NXT and their programming environments. Some experiences with a robotics product of Fischertechnik have been made by one of our Roberta Regional Centres.

III. Roberta Teacher Training

To ensure the quality of the Roberta concept, Roberta courses may only be delivered by certified Roberta teachers. As prerequisite Roberta teacher candidates have a didactical and preferably technical background. They have to pass a two days training delivered by Roberta coaches. The Roberta teacher training gives a hands-on introduction to the robots, the didactic material and the course concept. Special emphasis is on gender-oriented course design of mono-educative and mixed courses as well as on the creation of an open research-oriented learning environment. A certified Roberta teacher gets a login to the Roberta portal [1]. It provides the technical infrastructure to get access to additional didactic material and is the platform to get in contact with other Roberta teachers.

Fraunhofer IAIS trains and approves Roberta coaches who in turn train and certify Roberta teachers. All Roberta coaches have many years of experience as Roberta teacher. Furthermore they have outstanding expertise in didactic and educational robotics activities at universities or schools. Many of them are very active in coaching of robotic teams participating in robot competitions. Quality assurance, feed-back analysis and continuous improvement of the teacher training is one of the key elements in the development strategy of the Roberta Initiative.

IV. Roberta Regional Centres

The Roberta Regional Centres coordinate the courses in their regions and support the Roberta teacher associated to them. Furthermore, upon demand, they lend out construction kits to their Roberta teachers. For each newly founded Roberta Regional Centre there is a certain number of Roberta teachers being trained and certified by Roberta coaches.

At present, 23 Regional Centres have been established in Germany. Usually, they are hosted at universities active in robotics and/or teacher education. During the project Roberta-Goes-EU, 12 Roberta Regional Centres have been established in Austria, Italy, Sweden, Switzerland and the UK.

V. Results and Evaluations

We outline some figures to be suggestive of the current status of the Roberta Initiative.

- Several hundred teachers have been trained to be certified Roberta teachers in Germany and Europe.
- Several thousands children participated in Roberta courses. In 2009, at least 5000 children (60 % girls) participated in registered Roberta courses in Germany.
- Roberta courses are often the entry for girls to set-up robotic teams that participate - successfully - in robot competitions, like RoboCupJunior or FIRST Lego League.

Part of the didactic material and the training concept was evaluated by the University of Bremen [2,3] during the funded projects ‘Roberta’ and ‘Roberta-Goes-EU’ by getting feed-back from several hundreds of the Roberta course participants within an age range between 10 and 16 years. The analysis of the feed-back showed very similar results for Germany and Europe. In general, participation in a Roberta course significantly improves the self-confidence of girls in their own technical skills. This positive effect is slightly better if no boys are attending the same course. Nevertheless, the evaluation shows that boys are not distracted by the material, even though it was originally designed for girls. Based on these evaluations, Roberta courses are also open for boys. It is up to the Roberta teacher to decide this for a particular Roberta course. We asked the participants on their future interest in courses and got the following figures, again similar in Germany and Europe:

- 94 % enjoyed the courses
- 88 % would recommend it to friends
- 74 % would attend further courses

Acknowledgment

The Roberta course concept and its didactic material were developed in the German research project “Roberta – girls discover robots” [4] funded by the German Federal Ministry of Education and Research (2002-2006). The EU funded project “Roberta-Goes-EU” (2005-2008) adapted the results of the German project to be used in additional European countries.

References

[1] www.roberta-home.de (Roberta Portal)

[2] A. Wiesner-Steiner, H. Wiesner, H. Schelhove: „Technik als didaktischer Akteur. Robotik zur Förderung des Technikinteresses“, In: Hochschulinnovation: Gender-Initiativen in der Technik, Bremen, 2005

[3] H. Schelhowe, H. Schecker: Abschlussbericht. Wissenschaftliche Begleitung des Projekts "Roberta - Mädchen erobern Roboter", Bremen, 2005

[4] M. Müllerburg, U. Petersen, G. Theidig, “Mit Robotern spielend lernen: Das Projekt Roberta“, In: ROBOTIK 2004, (VDI-Berichte 1841)

ABB® Robotics Training Centre in Trnava

Peter Ducháček
ABB, s.r.o.
Dúbravská cesta 2, 841 04 Bratislava, Slovakia
peter.duchacek@sk.abb.com

Abstract — **In the last 20 years robotics all over the world experienced "boom" in applicability and in installations - the amount of robots in Slovakia increased by 2500 robots in the last 10 years. There were only few customers who ventured at investing into robotization before the year 2000, but the robots became common production device in the last decade. The quick lining up of robotics called forth the education also in this area of automatization. Seeing that ABB is always aiming to be close to the customer therefore the division of robotics established a robotics training centre in 2004.**

Keywords - robot; industrial robot; training; course

I. Training Centre

The training centre upraised from the need of the production factory PSA in Trnava to train its operators, maintenance technicians and programers. We trained more than 400 people since the beginning from all over Slovakia. We have started with 4 types of training and today we have training for almost all products supplied by us. The most requested training are basic programming and retraining for the new robots generations. In the basic training we teach the majority of operators and line users to get familiar with robots. The most important part of training is to get to know the robot and not to be afraid of it. The industrial robot is a machine just like any other and it does only what we teach it and what we ask it for. It often is needed to dismantle the fear or imaginings that the robot can decide for itself as can be seen in science fiction movies.

The second very important task of a trainer is to motivate the students. We experienced quite a few times that trainees did not even know why they were attending the training and when would they be actually working with robots. It is the trainers' job to engage everybody into the course in a way that is interesting, practical and even funny.

The length of the basic training - programming - is 5 days and we do not like to cut it down although it appears to be long to the customers (mostly managers, not the real trainees ☺). We know from our own experience that when a people do not get familiar with a robot in our centre, they have problems with using it in practice and they either keep calling us all over or they come back for retraining. We act up to the principle: "It is better to experience once than to hear twice."

The practice and hands-on training are necessary. To try and try it over. We make efforts that every student has a robot assigned to him to spend a maximum time with it. The students get several tasks daily to try some basic functions and to get ready also for the more complicated tasks later on. We choose tasks that they can often meet in real practice and as similar as possible to the application of that special customer. Sometimes the consumer brings a program he uses with him and we go through it with him and explain particular functions. This way they return to their workplace trained directly for their application. And because it is known that one learns best from his own mistakes, we try to simulate the most common errors within the course so that the students remember them and do not repeat them.

At the end of the training there is a test for the students to check up their knowledge. We allow them to use all accessible resources which motivates them to take notes during the course which they can use then while testing.

One of the tools of the training is Robot Studio. It is a simulating software of the ABB robots in which all the robot functions are fully simulated. It is possible to choose a type of a robot, to insert 3D models of the robotic cell and to simulate the movements, communication, cycle time. We offer Robot Studio for free for schools, that is how ABB supports robotic knowledge in students.

The trainers are the members of our team who are not only trainers - they are real service men who take turns to train the customers from time to time. That is how we keep the training and the service together and how we can take through the practical experiences from the service and projects to the training - which makes the best contribution for the customer.

The trainees can also try how the remote service works. This function can be activated in a robot and then it is possible to revise the robot status from any place in the world. The Remote Service transmits the data about the system and the system functionality and malfunction through GPRS network.

Next to the training centre is Pilot Site where robotic links are being assembled, started up and tested before delivering and the final start up at customers'. While testing the links we also train our costumers for these links which is the best training they can get. During the tuning of the robotic link many problems appear which enables to get to know the link as good as it gets.

Considering the number of trainees and the variety of training portfolio ABB is happy for taking a big part in increasing robotics literacy in Slovakia.

II. Characteristics of Training Centre

Sample of training:

- Basic operators training
- Basic programming
- Advance programming
- Electrical maintenance
- Mechanical maintenance
- Robot Studio - offline programming
- Application software

Robots in training centre:

- IRB 2000 M93
- IRB 2400L M2000
- IRB 6600-2.55/175 M2000 Automotive version
- IRB 6620 M2004
- IRB 140 M2004

Applications in training centre:

- Manipulation
- Flex Finishing
- Arc Welding
- Spot Welding
- Pick Master 3 - robot with camera

References

[1] http://www.abb.com/robotics

[2] http://www.abb.com/product/9AAC910011.aspx

[3] Ducháček, P.: Robotika ABB v nových priestoroch. In: ABB Spektrum. Vol. 10 (2008) , No 2., ABB s.r.o., Bratislava, 2008. Available on-line: <http://www02.abb.com/global/skabb/skabb104.nsf/0/dbb98f5ac710f71cc1257444002ba6c5/$file/Spektrum0208.pdf>

[4] Myjavec, R.: Bližšie k zákazníkovi. Tréningové a testovacie centrum robotiky ABB v Trnave. In: AT&P journal, Vol. 13 (/2006), No. 1, Bratislava. Available on-line: http://www.atpjournal.sk/ asopisy/atp_06 / pdf/atp-2006-01-18.pdf

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Design and Validation of a Robotic System to Interactively Teach Geometry

Lorenzo Riano and Martin McGinnity
Intelligent Systems Research Centre
University of Ulster
Londonderry, UK

***Abstract*—Learning geometry can be significantly improved if the student interacts with shapes and their transformations. We present the design and validation of a robotic system that teaches geometry by natural interaction. The robot is able to draw arbitrary shapes in the environment by means of its own movement. It is also able to detect and track the student movements, representing them as geometrical shapes and reproducing them.**

We will show four experiments where a robot is able to draw mathematical shapes, including an affine transformation, and two experiments where the robot reproduces trajectories a student previously showed it.

Fig. 1. The Scitos-G5 during a demo

I. Introduction and System Overview

In the late sixties Seymour Papert invented Turtle Graphics and LOGO [1], a programming language for geometric drawing as a tool for basic programming and geometry teaching. A real or simulated robot, shaped as a turtle, is programmed to move in a $2D$ space and to draw lines while moving. The main reason for its development and success is that it places the learner in an interactive environment where he/she can experiment with the geometry involved in drawing a figure, while receiving direct visual feedback from the turtle movements [2].

It is commonly assumed that the teaching of geometry should contribute to the learning of, among others, "the movement between theoretical objects and their spatial representation" [3]. Observing an abstract shape being drawn by a real robot should therefore significantly contribute to a student's understanding of geometry. Other concepts such as affine transformations can be more efficiently learnt by a student if they observe their effects on an arbitrary shape or observethe effects of varying parameters [4].

In this paper we present a robotic system that interactively teaches geometry according to the guidelines above. In particular, it describes shapes in the environment by means of its own motion. To the best of our knowledge this is the first application of a real robot to teach geometry by natural interaction.

Such interaction starts from the student, who defines a geometric shape. This can be accomplished in two ways: i) by using the mathematical description of a shape, or ii) by moving in the environment, thus describing a shape via his/her own movements. Once a shape has been chosen, the robot shows what its approximation of it looks like, using a simulated trajectory. The student can vary parameters such as the smoothness of the approximation and the speed at which the robot should travel. All of these parameters have an intuitive interpretation, and the student can immediately visualise the effect of changing them by observing how much the proposed robot trajectory matches the original shape.

When the student is satisfied with the proposed trajectory, the robot starts moving along it. This way it is virtually "drawing" a shape in the space by moving in the environment, providing a visual feedback to the student. Once the robot stops moving the student can manipulate the shape by using affine transformations, and observe again the robot moving along it. This way the student can "see" geometry, and he/she can interact with it by manipulating figures and observing the effect on the robot's movements.

The main target audience of the proposed system are primary school pupils, who are learning the basic concepts of geometry. In this case observing the shapes being drawn by the robot's movements can improve their understanding of the subject [3]. Our system can as well be used in an undergraduate robotics course, where the students are presented with problems of path planning and trajectory following, and they can observe the results of varying several parameters on the physical robot itself.

In order to carry on the above task the robot requires several components, namely i) people detection and tracking, ii) trajectory extraction and iii) path following. The rest of the paper describes in more detail the proposed system and the experimental results that validate the system.

l.riano@ulster.ac.uk, tm.mcginnity@ulster.ac.uk

This research is supported by the Centre of Excellence in Intelligent Systems (CoEIS) project, funded by Northern Ireland Integrated Development Fund and InvestNI.

This paper is dedicated to the memory of Prof. Ulrich Nehmzow, whose inspiration and ideas led to the development of this work.

TABLE I

CONFUSION MATRIX FOR (LEFT) THE VIOLA-JONES CLASSIFIER AND (RIGHT) WHEN COMBINED WITH THE RBFN, FOR A PERSON PRESENT OR NOT PRESENT.

	Present	Not Present
Detected	98%	24%
Not detected	2%	76%
	Present	Not Present
Detected	78%	0%
Dot detected	22%	100%

II. PEOPLE DETECTION AND TRACKING

An effective people detection and tracking system is the one that minimises both false positives and negatives, i.e. it does not mistake objects in the environment as human.

1) Face detection: In order to reliably detect people in the environment, we used the Viola and Jones approach to face detection [5]. This classifier is scale and light conditions invariant.

In order to measure its performance, we conducted two experiments, the first one with a person always facing the robot camera, the second one without any person in the environment. During the first experiment the robot was following the person, while during the second one it was randomly moving. Both experiment lasted around 15 minutes each, and the results are summarised in the confusion matrix shown in Table I, left.

These results show that, although the Viola-Jones has an outstanding true positives rate of 98%, it performs poorly in false negatives with a rate of 24%. A second drawback of the Viola-Jones approach is that a person has to directly face the robot to be detected. This is a major problem for our application as we want a person to describe a trajectory in the space, and this would not be easy if he/she has to face all the time the robot. A third drawback of the Viola-Jones approach is that it can detect people only up to about $2m$ away from the camera when using wide angle lens. We will describe a solution to this problem in section II-3 with the introduction of the particle filter for tracking.

2) RBFN for legs detection: The Viola-Jones approach described above relies on vision to detect faces. A robot is usually equipped with several sensors, which can be combined to make a classifier stronger. In the past the laser sensor has been used to detect people using their legs, either as the sole sensor [6] or together with a camera [7], [8]. In most of the previous works, in order to train a laser based classifier a huge set of legs laser scans had to be manually constructed and labeled [6], or the authors created an ad-hoc classification algorithm [7]. Our approach is to use the face detection algorithm described before to train a Radial Basis Function Network (RBFN) classifier [9] for legs detection. This way the training process is completely automated.

In order to train a RBFN to classify leg patterns in laser scans, we collected the training data using the Viola-Jones face detection algorithm described above. Specifically, we had the robot running for 20 minutes with a person constantly in front of it, while recording the laser scan readings in a set C_1. As we stated before, the detection rate of the face detection algorithm is 98%, so almost all the laser scan readings refer to legs. The camera had been calibrated so that for every pixel it is possible to calculate the angle θ between that pixel and the camera itself. Considering that both the camera and the laser are vertically aligned, θ identifies a unique laser reading, taken at angle θ, in a whole laser scan. For every face detected, its centroid is extracted and the corresponding angle θ is calculated. Every laser scan in C_1 is then clipped between angles $\theta-\pi/6$ and $\theta+\pi/6$. This way C_1 is a training set for the class 1, "legs in a laser scan", composed by 60-dimensional vectors. We then built a second set of laser scans C_0 by letting the robot randomly move in an environment with no people in it for 20 minutes. This set represents all the laser scans with no legs in it. Both C_0 and C_1 are then been used to train the RBFN.

The face detector and the legs detector have been combined to create a new people detection algorithm, that outputs 1 if both Viola-Jones and the RBFN detect a person. The new people detector confusion matrix is summarised in Table I, right. It can be seen that the true positives detection rate dropped from 98% to 78%, but the number of false positives is now 0%, thus solving the problem with the Viola-Jones only approach. Once both classifiers agree that a person has been detected, the robot can switch to laser classification only. This solves the problem of a person having to face the robot camera all the time.

The only drawback of this approach is that the training set C_1 contains legs patterns which are only closer than about $2m$, because this set was "created" by the Viola-Jones detector. This means that the RBFN legs classifier can detect people only up to about $2m$.

3) Particle filter: The 2m limit described before does not allow a person to move arbitrarily in the environment, which is necessary to describe shapes. In order to solve this problem we employed a particle filter. An excellent review of this technique is in [10], while an application of it to people tracking using a laser sensor is in [11].

A particle filter requires a big number of particles to work reliably [12]. As a computational trade-off, in our application each particle does not try to detect people, but it tracks only a single laser scan. For this reason the adopted tracker relies heavily on a people detector that does not report false positives.

We tested the particle filter in several experiments using 2000 particles. In the worst case the tracker failed after 4 minutes of continuous operations, but in the average it lasted around $10m$. Moreover, the particle filter is able to track a person movements up to 8 meters away from it.

III. TRAJECTORY FOLLOWING

An arbitrary shape is represented by the robot as a parametric B-Spline. A motor controller that integrates feedforward and feedback signals is used to drive the robot along that shape. In the following we will give an overview of both B-Splines and the proposed controller.

A. B-Splines

A Basic Spline (B-Spline) is a parametric curve often used for interpolation and regression [13]. They have been widely used in robotics to approximate trajectories [14], [15], [16].

Given $m+1$ real numbers $t_0 \leq t_1 \leq \cdots \leq t_m$ and $m+1$ *control points* $p_i, i = 0, \cdots, m$, a B-Spline of degree n is a parametric curve $S(t)$ composed of a linear combination of basis functions $b_{i,n}$ of degree n, as given in (1).

$$S(t) = \sum_{i=0}^{m} p_i b_{i,n}(t) \tag{1}$$

where $b_{i,n}(t) \neq 0$ only if $t_{i-1} \leq t < t_i$. This makes a B-spline piecewise defined, i.e. the basis functions are non-zero only in a closed interval. In order to obtain a C^2 smooth function, usually a cubic polynomial is used as a basis function $b_{i,3}$. When used to approximate a parametric curve, B-Splines are defined as in (2).

$$S(t) \equiv (S_x(t), S_y(t)), t = 0, \cdots, 1 \tag{2}$$

where $S_x(t)$ and $S_y(t)$ are two B-Splines.

A B-Spline can be used to approximate a function $f(x)$ represented as a finite set of points (x_i, y_i). In this case the sum of squares error (3) is zero.

$$\sum_{i=1}^{N} (y_i - S(x_i))^2 \tag{3}$$

When the data points (x_i, y_i) are noisy, the interpolation requirement is not reasonable. In this case the smoothing spline in (4) is preferred [17].

$$\sum_{i=1}^{N} (y_i - S(x_i))^2 + \lambda \int_{x_1}^{x} N \left[\ddot{S}(x)\right]^2 dx = \min \tag{4}$$

where $\lambda \geq 0$ is a *smoothing factor*. When $\lambda = 0$, we obtain the interpolation again. The higher λ, the less the spline is constrained to pass through the data points (x_i, y_i). In the following we will use λ to generate smooth approximations of people trajectories.

B-Splines are *affine invariant*, i.e. if an affine transformation is applied to a B-spline curve, the result can be constructed from the affine images of its control points. This is very important in our application as we want the robot to show the effect of geometric transformations on shapes. Moreover, the derivatives of a B-Spline can be analytically calculated. This will come at hand for the development of the controller.

B. Controller

The trajectory tracking for mobile robots is characteristically a nonlinear problem. Several solutions have been proposed in the past. However, in most of them only simulation results are presented [18], [19], [20], or they require complex calculations that are not feasible in a real-time controller [21], [22]. In this work we decided to adopt a feedforward-feedback control law: the robot follows the trajectory described by the B-Spline, and at the same time tries to minimise a distance error using a PID controller.

1) Feedforwad control law: The motion of a unicycle robot can be described by the system of differential equations in (5).

$$\begin{cases} \dot{x} = v \cos\theta \\ \dot{y} = v \sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{5}$$

where x, y, θ are the robot position and orientation, and v, ω are the robot linear and angular speeds. If we consider a time interval dt sufficiently small, then we can approximate the robot movement as piecewise circular, where the radius ρ of the circle in the time interval $[t, t+dt]$ is in (6).

$$\rho = \frac{v(t)}{\omega(t)} \tag{6}$$

Any parametric curve $(x(t), y(t))$ can be approximated the same way by a set of arcs, whose curvature κ is in (7).

$$\kappa = \frac{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}} \tag{7}$$

Considering that the curvature of a circle of radius ρ is constantly equal to the reciprocal of the radius, we can rewrite (6) as in (8).

$$\frac{v(t)}{\omega(t)} = \frac{1}{\kappa} \tag{8}$$

Moreover, the angular coefficient of the tangent to a point (x_0, y_0) of a parametric curve is given in (9), which is also the instantaneous linear speed v of a point moving along that curve.

$$v = \sqrt{\dot{x}(x_0) + \dot{y}(y_0)} \tag{9}$$

By substituting (7) and (9) into (8), we obtain the feedforward control law for the robot angular speed in (10), which is a function of the B-Spline derivatives and the robot linear speed.

$$\omega_f = \frac{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|}{\dot{x}^2 + \dot{y}^2} v \tag{10}$$

As outlined in the introduction, the user can select two parameters, namely the time approximation dt and the robot speed v. Once the user has selected them, the robot shows a simulated trajectory. This is created by using a first order Euler approximation of (5), where ω is given by (10) and dt, v are supplied by the user. This is useful to check if the robot is theoretically able to follow a given trajectory. However, as (5) does not take into consideration the robot dynamics, this method is necessary but not sufficient to guarantee a correct trajectory following.

2) Feedback control law: The role of the feedback controller is to correct the robot when it deviates from the desired trajectory. This deviation can be calculated by estimating the distance between the robot and the trajectory. As there is no analytical solution to this problem, we used the Newton method to find a zero of the function

Given a point (p, q) and a parametric curve $(x(t), y(t))$, the squared distance between the point and any other point on the curve is given in (11).

$$\Delta x(t)^2 + \Delta y(t)^2 \tag{11}$$

where we defined $\Delta x(t) = x(t) - p$ and $\Delta y(t) = y(t) - p$. To find the minimum of (11) we differentiate and equate to zero, as in (12)

$$2\Delta x(t)\dot{x} + 2\Delta y(t)\dot{y} = 0 \tag{12}$$

The Newton method is an iterative method to find the zeros of a function $f(t)$. It starts with a guess t_0 and every step it updated the candidate solution as in (13)

$$t_i \leftarrow t_{i-1} - \frac{f(t)}{\dot{f}(t)} \tag{13}$$

By applying (12) to (13) we obtain the iteration step (14) to find the minimum distance between a point (p, q) and a curve trajectory.

$$t_i \leftarrow t_{i-1} - \frac{\Delta x\dot{x} + \Delta y\dot{y}}{\dot{x}^2 + \dot{y}^2 + \Delta x\ddot{x} + \Delta y\ddot{y}} \tag{14}$$

During our experiments we found that the Newton method requires only a few iterations to converge to a solution, so it is usable in a real-time controller.

If the robot position is (x_r, y_r) and the closest point on the desired trajectory is (x_d, y_d), then we can define the angular error as in (15).

$$e_\theta = \arctan\left(\frac{y_d - y_r}{x_d - x_r}\right) \tag{15}$$

The feedback control law is a PID with the error in (15) to steer the robot. The final robot controller is given in (16)

$$\begin{cases} v = const \\ \omega = \omega_f + \omega_b \end{cases} \tag{16}$$

where ω_f is the output of the feedforward controller (10) and ω_b is the output of the PID when supplied with the angular error (15).

IV. Experimental results

All the experimental results have been produced using a Scitos-G5 robotics platform equipped with a laser range finder and a camera (Figure 1). The trajectories have been recorded using the Vicon tracking system, that allows tracking an object with an accuracy of $1mm$. Every trajectory, before being replicated by the robot, has been rotated and translated so that it starts from the robot position and it is aligned with the robot x axis.

We performed two groups of experiments: in the first group the user asks the robot to follow a pure geometric shape, while in the second group the user defines a shape by moving in the environment. For every shape we show the corresponding one in the robot frame of reference, the simulated approximation (see section III-B1) and the real shape as produced by the robot movements. Table II shows the parameters and the statistics for every shape.

For every shape we calculated the approximation error as the mean absolute distance between the user-provided trajectory and the approximated one. The errors for each experiment are summarised in Table II. Note that if we were using the mean error the results would have been close to zero, as they are equally positive and negative (cf. Figures 2,3 and 4). However, the mean absolute error presents a clearer picture of the worst case scenarios.

TABLE II
Parameters and statistics for the tested trajectories.

	v	dt	λ	error [m]
Cosine	0.3	0.08	0	0.17
Spiral	0.2	0.08	0	0.21
Square	0.3	0.08	0.01	0.2
Sheared square	0.3	0.08	0.01	0.19
Trajectory 1	0.4	0.1	2	0.30
Trajectory 2	0.4	0.1	2	0.28

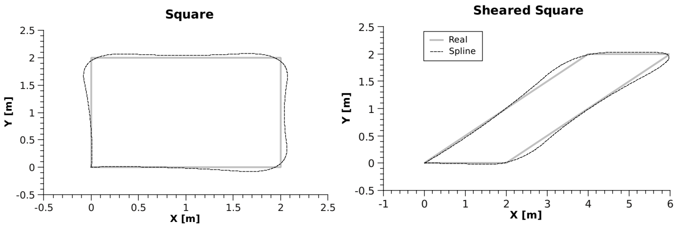


Fig. 2. The square and the sheared square used during our experiments.

3) Mathematical shapes: We tested four mathematical shapes: i) a cosine, ii) a spiral, iii) a square and iv) a sheared square. Figure 2 shows the last two shapes together with the spline approximation. The approximations of the cosine and the spiral match exactly the original data, so they are not shown here. Figure 3 shows the simulated and the real robot trajectories for all the mathematical shapes.

4) User produced shapes: We tested two user produced shapes. Both of them have been produced by the user moving in front of the robot while being tracked. Figure 4 top row shows the trajectories as observed by the tracker and the corresponding spline approximation. Figure 4 bottom row shows the simulated and the real robot trajectories for both the observed shapes.

5) Discussion: Among the mathematical shapes, both the square and the sheared square have discontinuities at the corners. For this reason the splines deviate significantly from the desired shapes when around the corners. This is due to the “smooth” nature of splines, which are not suitable to approximate a piecewise linear curve like a square. In order to avoid sharp turns at the corners, we opted for a slightly smoothed approximation with a λ parameter of 0.01.

The average error along all of the mathematical shapes is low, as shown in Table II. The only time the robot noticeably deviated from the desired trajectory was at the beginning of the spiral shape, as shown in Figure 3, top right. This is due to the initial high curvature of the spiral, which is not reproducible by the robot given its physical constraints. However, the feedback control law quickly corrected the error and drove the robot back on the desired trajectory. The same graph highlight the differences between the real trajectory and the simulated one, and the role physical constraints play in a robot controller.

The sheared square has been produced by shearing the square by $1m$ along the x axis. This shows how our proposed system can be used to interactively teach affine transformations.

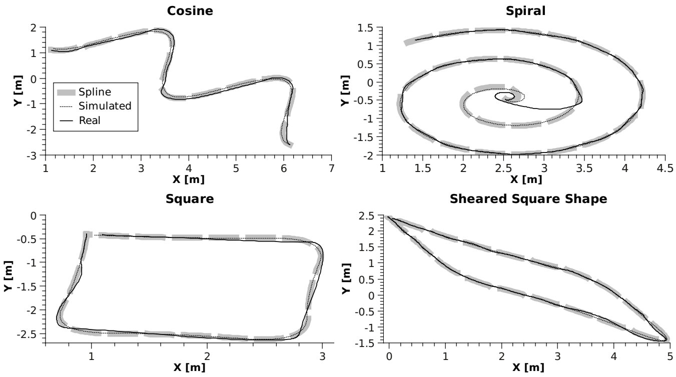


Fig. 3. The mathematical shapes with the corresponding simulated and real robot trajectories. All the graphs are rotated to match the robot frame of reference.

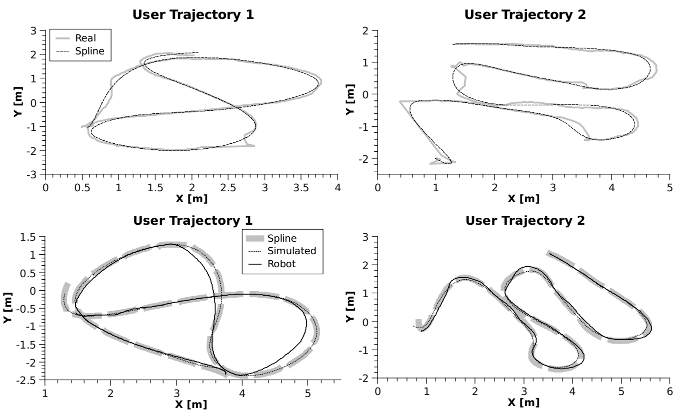


Fig. 4. (Top row) The user produced trajectories observed by the tracker, and the corresponding spline approximations. (Bottom row) The simulated and real robot trajectories. All the graphs are rotated to match the robot frame of reference.

The trajectories observed by the tracker are noisy and discontinuous in several points, as shown in Figure 4. Trajectory 2 benefited the most from the parameter λ, with the effect of obtaining a smooth non interpolating curve. This is reflected in the higher errors both the simulated and the real trajectories exhibit.

V. Conclusions and Future Work

In this paper we presented a robotic system to interactively teach basic geometry. The interaction with a student happens when the user selects a shape, and when the robot shows what the shape will look like when it will reproduce it, given the parameters chosen by the student. At the end of the interaction the robot describes the shape in the environment by moving along it.

The proposed system is composed of two main parts: a passive one, which detects and tracks people movements, and an active one, which plans the robot movements and drives it along a shape. We showed with several experiments that the system is reliably able to carry on both tasks.

In this paper we focused mainly on the application an on the results. In the future we will perform a survey among students and teachers to identify the main points where this system can be improved. We will also modify the controller so that it will include the robot physical constraints.

References

[1] H. Abelson and A. A. Disessa, *Turtle geometry: The computer as a medium for exploring mathematics*. The MIT Press, 1986.

[2] T. Fujita, K. Jones, and S. Yamamoto, "Geometrical intuition and the learning and teaching of geometry," in *Topic Study Group on the teaching of geometry at the 10th International Congress on Mathematical Education*, 2004, p. 411.

[3] C. Laborde, C. Kynigos, K. Hollebrands, and R. Strasser, "Teaching and learning geometry with technology," in *Handbook of research on the psychology of mathematics education: Past, present and future*, 2006, p. 275304.

[4] K. Jones, "Issues in the teaching and learning of geometry," in *Aspects of Teaching Secondary Mathematics: perspectives on practice*, 2002, pp. 121–139.

[5] P. Viola and M. J. Jones, "Robust Real-Time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[6] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Proc. of the int. conf. on robotics & automation*, 2007.

[7] N. Bellotto and H. Hu, "Multisensor-Based human detection and tracking for mobile service robots," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART B: CYBERNETICS*, vol. 39, no. 1, p. 167, 2009.

[8] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, "People tracking and following with mobile robot using an omnidirectional camera and a laser," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, p. 557562.

[9] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE (special issue: Neural Networks I: Theory and Modeling)*, vol. 78, no. 9, p. 14811497, 1990.

[10] A. Doucet, N. D. Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[11] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *The International Journal of Robotics Research*, vol. 22, no. 2, p. 99, 2003.

[12] S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, vol. 21, no. 4, p. 93, 2000.

[13] C. de Boor, *A Practical Guide to Splines*. Springer-Verlag, 1978.

[14] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the DARPA grand challenge," *Journal of field Robotics*, vol. 23, no. 9, p. 661692, 2006.

[15] S. Gulati and B. Kuipers, "High performance control for graceful motion of an intelligent wheelchair," in *IEEE International Conference on Robotics and Automation*, 2006, p. 39323938.

[16] H. Eren, C. C. Fung, and J. Evans, "Implementation of the spline method for mobile robot path control," in *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE*, vol. 2, 1999, pp. 739–744 vol.2.

[17] P. H. C. Eilers and B. D. Marx, "Flexible smoothing with b-splines and penalties," *Statistical Science*, vol. 11, no. 2, pp. 89–121, 1996.

[18] E. Guechi, J. Lauber, and M. Dambrine, "On-line moving-obstacle avoidance using piecewise bezier curves with unknown obstacle trajectory," in *Control and Automation, 2008 16th Mediterranean Conference on*, 2008, pp. 505–510.

[19] J. H. Hwang, R. C. Arkin, and D. S. Kwon, "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, p. 14441449.

[20] B. V. G., H. S. A., and J. D. S., "Auto guided vehicle control using expanded time b-splines," in *Systems, Man, and Cybernetics, 1994. 'Humans, Information and Technology'., 1994 IEEE International Conference on*, vol. 3, 1994, pp. 2786–2791 vol. 3.

[21] L. Lapierre, D. Soetanto, and A. Pascoal, "Nonsingular path following control of a unicycle in the presence of parametric modelling uncertainties," *International Journal of Robust and Nonlinear Control*, vol. 16, no. 10, p. 485504, 2006.

[22] C. G. Bianco, A. Piazzi, and M. Romano, "Smooth motion generation for unicycle mobile robots via dynamic path inversion," *Trans. Robot. Automat*, vol. 17, p. 413422, 2001.

Lessons learnt with LEGO Mindstorms: from beginner to teaching robotics

Martina Kabátová
Dept. of Informatics Education
FMFI Comenius University
Bratislava, Slovakia
kabatova@fmph.uniba.sk

Janka Pekárová
Dept. of Informatics Education
FMFI Comenius University
Bratislava, Slovakia
pekarova@fmph.uniba.sk

Abstract—**In this paper we describe several lessons of educational robotics at different level of robotics experience. First one focuses on developing basic skills needed to successfully control a robotic model. The other one uses advanced programming skills to provide the communication between two programmable bricks. We also describe organization of the project in brief. LEGO Mindstorms kit was used as a learning platform for all activities. Addressing differences between beginners and experienced students, we list a few tips and recommendations how to execute quality robotic lessons.**

Keywords: robotics education, LEGO NXT, programming, guidance, project

I. Introduction

Robotics presents an attractive introduction to the object-oriented programming or higher programming languages (see [2], [5], [10]), but it can be also used in the lower levels of education. Dealing with real robots has a high motivational effect – students visualize their robot as a toy [11] which behavior can be set according to the scenario where it is used. The experiences with robots are tangible although their design requires much abstract thinking. Finally, they enable rich varieties of interdisciplinary projects. Therefore we consider robotics to be a powerful tool for developing thinking. We pay special attention to the preparation of pre-service teachers who can enrich their teaching repertoire by the robots' use. We have been realizing the robotic seminar for them for several years. We try to explore the effective way for constructing students' comprehension of robotics. We often find our methodology similar to different courses worldwide.

II. How robotics is taught

Carnegie Mellon Robotics Academy [3] offers a special robotic course for educators. Here they learn more about:

- MOTION and CALCULUS (What is a robot?; Mindstorms hardware; Movement and rotations; Size, distance and movement; Abstract bridges; Challenge: go as close as possible)
- ROBOTIC SENSORS (Measure – Plan – Execute strategies overview; Touch, ultrasonic, light and sound sensor; View Menu; 'Wait for' block; Limits of the measurements; Tasks with sensors)
- DECISION MAKING (Repetition; Obstacle detection; Cycle, condition and conditional cycle; Line following; Setting the ride through obstacles; Iterative solution of the problems; Challenge: ride through obstacles)
- EDUCATIONAL ROBOTICS (Possibilities; Challenges; Robotics in your school)

Different approach to the educational robotics can be found in TERECOP project [1]. Teachers learn how to teach robotics in a constructionist way. Besides the basics of programming the kits, participants also get informed about constructionist learning philosophy and project-oriented classes. They analyze and assess the robotic model, suggest own assignment for their students and think over the organization of robotics projects. In the core lessons of the project they learn to measure with sensors and control the motors of the robot using the basic program blocks. Moreover, they learn how **to check if robot works** in the way prescribed in assignment and how **to modify the program** in order to fit assignment needs.

MIT Lifelong kindergarten applies four principles into their leisure time robotics workshops for children and families [13]: (a) Focus on **Themes** (not just Challenges); (b) Combine **Art and Engineering**; (c) Encourage **Storytelling** and (d) Organize **Exhibitions** (rather than Competitions). Authors of this learning approach aim to make robotics attractive to the as broad range of people as possible.

[9] puts emphasize on creating **cooperative learning environment** where **small groups of students** maximize each other's learning while working on robotics projects. In the proposed curriculum students work with ready model at first, in order to understand possibilities and limitation of robotic kit. They use programs prepared in advance, learn to modify them and after that try to design a functional robotic model on their own. Similar approach is presented in [14]. **Challenges** are taken in the end of each lesson in order to **ensure understanding** to the core concepts of the lessons.

Youth Engaged in Technology programme also combines team building activities with demonstration of programming instruction to the robot [6]. This course further contains necessary math to calculate gear ratios. Several exercises are focused on building and mechanical components of robots. Open-ended challenge completes the course syllabus.

[8] has decided to give his robotic course the form of open lab where students can spend **much time** on solving various

robotic related tasks which encourages students to be more creative in their design and robot implementation. The assignments are close-ended and clearly defined, although the author recognizes the potential of the **open-ended projects in combination with the contests**.

III. Seminar Robotic kits in education in detail

We have been applying constructionist ideas and principles ([12]; [13]) in our seminar practice:

- **learning by doing**, **hands on activities** through own experiences – students build and program robotic model and test its functionality;
- **authentic success in finding the problems and their solutions** – students decide how the robot will work and choose their way how to achieve this, they select the topic for their project and explore programming possibilities;
- **the hard fun** and **playful learning** – robotic kits are in fact the toys, but solving some tasks with them can be quite complicated, the atmosphere on the seminar is relaxed and we try to help students learn in an entertaining way;
- **creative learning** by designing and inventing is included in the creating the robotic model;
- **combination of digital technologies** as a building material together **with art materials** – students can decorate their robots, make a costume for their models or produce some coulisses;
- **enough time** – the syllabus of the seminar is quite flexible, we can spend more time on activities that last „too long";
- **freedom to make mistakes** and learn from them – students get space for their own, independent solutions in which they go wrong sometimes, we try to reveal the core of problem in common dialogue and help them to solve it;
- **teamwork**, **collaboration**, distribution of roles within the group, common work on problem solving – students learn how to organize teamwork, some assignments cannot be completed individually (e.g. to prepare the robot for the contest);
- **learning together** – it is not possible for us – the teachers – to be prepared for the whole range of troubles that can happen, we also solve novel tasks and learn new things together with our students.

The syllabus of the seminar keeps balance between closed tasks having the only solution and open-ended projects:

TABLE I. The syllabus of the seminar

Week	Topic of the seminar	Tasks, solution, teamwork
1	**Programming without computer?**	Simple closed tasks, one solution, small group
	Creating simple program through NXT brick interface	
2	**Introduction to Mindstorms programming**	Simple closed tasks, one solution, small group
	Move, Display and Wait for block, Cycle, Condition	
3 - 4	**More on Mindstorms programming**	Simple closed tasks, one solution, small group
	Procedures and variables; Parallel processes	
5	**Experiments with sensors**	Single task mixed from small group work and common activity with all students
	Read/Write to file, variables	
6	**Communication between robots**	Advanced tasks for small groups collaboration
	Send/Receive message	
7 – 9	**Our Project 1. Preparation for the robotic contest as an alternative**	Open-ended project, many possible solutions, small group
10 – 12	**Our Project 2. Exhibition and presentation of the models**	Open-ended project, many possible solutions, small group builds single model for the common topic

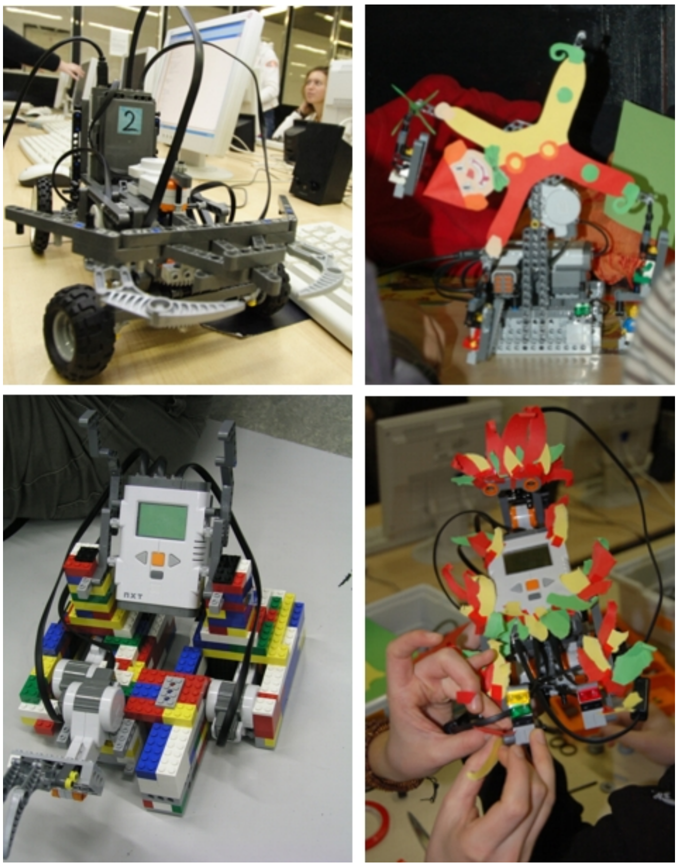


Figure 1. Our Project – examples of outputs

We have sketched the way of organizing open–ended projects in [7]. Briefly, during previous terms students:

- built and programmed robotic elevator controlled by touch sensors (see also section V),

- suggested and realized robotic models for the Space and Playground topics,
- designed and completed moving pieces of a Spooky castle, Amusement park and Intelligent house.

We encourage students to do **pedagogical reflection** of their own robotic design. The part of robotic assignment is also a teacher checklist where they advice fictional teacher how they can realize similar robotic project – how much time is needed, special needs of hardware, previous programming skills, common problems and their solution. Although documentation of the work in this way isn't very popular among students, they are able to produce high quality description of their robotic device.

In the following sections we describe three lessons: (a) an initial lesson in Week 2 where students work in the programming environment for the first time; (b) communicating between robots in Week 6 and (c) design of the project-based activity The Lift. We want to show the continual progress from simple close-ended tasks that help the beginner to develop basic programmer skills to change the robot's behavior to more open-ended tasks that require more creativity and open space for more qualitatively different solutions. We think it is necessary to give participants the possibility to **design, build and program the robotic model on their own**. This brings additional effect: the pre-service teachers experience some common problems when designing the robots and can more effectively give advice to their own students in future.

IV. Lesson one: controling the robot

The students use the standard model of the robot (as presented in Robot Educator section of Mindstorms programming environment). We provide them by the set of clearly defined, simple tasks so that they get to know the possibilities of the iconic programming environment used to program the robot. During previous terms we noticed several problems and misconceptions common to various groups of students. Let's have a closer look at some of them.

Robo – archeologist Assignment: Choose one of the letters written on the stone.

- Teach your robot to move in the outline of the letter.
- Teach it to move in the outline of crocodile teeth:

In the very first assignment which should be solved in Mindstorms programming environment the students express high expectations of robot's possibilities. Many of them ask whether the robot should follow the black outline of the letter and some students even don't ask and try to program it. The main idea of the assignment is in fact much simpler: they should learn to use Move block and use the sequence of this type of block to produce the track of the robot in a fixed dimensions (we advise the students the blocks needed in solution on the edge of the assignment).

In trying to create the crocodile teeth track students find out that the angle used in Move block isn't the same angle in which the robot turns. The angle stands for the rotations of the motor – 180° means half of one rotation. Solution of the task is then often based on many trial-and-error experiments with the settings of Move block in order to create a desired outcome.

Our students are the experienced programmers when beginning to attend the seminar. At this point they suddenly find out that programming the real robots differs from programming virtual ones (e.g. a turtle in Logo programming language) – they have to consider physical aspects of the model as well as the properties of the environment. When they program a robot that should move after whistling, some of them face the problem that robot will start moving immediately after the program launches (as the noise in room is often high). They discover the need to calibrate the sensors.

We always encounter students who try to program a continuous motion of the robot (no matter how outer conditions are). They soon question why the program containing only one Move block set to Unlimited steps doesn't work as they expected. This is the other difference between programming real and virtual creatures that needs to be explained explicitly and perhaps demonstrated in more guided instructional way.

Students learn to set robot's behavior depending on outer conditions – values measured by sensors in several real-life tasks, for instance:

Robo-racer Assignment: Your robot is waiting for the start-the-race signal. When hearing it, it will move forward.

- Teach it to take its run – to increase its speed.
- It will quickly go forward. It will stop when it finds the (black) line marking the finish of the race circuit.
- After achieving the finish line, the robot will turn all around because of the joy from victory.
- Each racer will smile after turning around – there will be a smiling face on its display.

The lesson finishes by the task requiring a partial disassembling of the robot. We were inspired by [4] in its assignment – the challenge was to increase the robot's speed so that it will move faster than the programming environment allows by default:

The Thief Assignment: Try to achieve as fast motion of the robot as possible. Find at least two different solutions.

Hint: You may need to modify the construction of your robot slightly. **Notice how the power of the motor transfers to the wheels.**

Besides experimenting with the gears and program settings students should find the answers to these questions:

- What will happen if you enlarge the cog wheel connected to the motor and use smaller cog wheel connected to the wheel?
- What will happen in opposite case?
- How much load can the robot push if you use various types of cog wheels?

We motivated students by the short movies in [4] and discussed the answers to the questions with them. They had chance to create hypothesis and test it immediately in real conditions. This task was appreciated also by two girls which showed no previous interest in mechanical issues of robotics and they were proud of themselves that they found the arguments supporting their opinions. Finding solution to this task requires lot of time (because of the need to assembling new driving mechanism for the robot) and the task should be included as the last piece of the lesson because of the need to re-build the robot construction. It also inspires future teachers to think in an interdisciplinary way – they think over the math hidden in mechanics and construction of the robotic models.

V. Lesson Two: The Robots Communicate

As soon as the students manage the basics of programming language, one lesson is dedicated to the communication between two (or more) NXT bricks. The bricks have Bluetooth and can be programmed to send and receive various messages. Unfortunately the process of connecting two bricks has some flaws. We found out that creating a connection should be guided as much as possible as the process is nothing near intuitive and detailed guidance can significantly reduce mistakes and errors that have nothing to do with controlling the robots.

Once the bricks are successfully connected the teams shall write programs to send and receive messages. In first simple introductory assignment one brick sends a word and the other brick displays the word on its display. We used to give another introductory assignment but we realized that it is not easy if it should be done correctly (the task is to send a Morse signal and the other brick should beep the message out).

The final task is to program a car and its remote controller.

Remote-control car Assignment:

Team A: Create a program that will send control messages to the car. Use NXT buttons on your brick as a remote controller.

Team B: Create a program to receive messages and move the car according to them.

Both teams: Negotiate the message content and how to interpret them. Think about car controlling - what is the desired behavior that will be suitable for a race?

The programs contains loop, if-statement and reaction to the NXT buttons or blocks to move the motors. There are several good solutions either with variables or without using them. We encourage the students to find their own solution and we only correct their errors once they ask us. The most common solution for receiving program contains two or more nested if-statements reacting to various messages from the other NXT brick.

Students should also design the behavior of the car, once it is successfully controlled. After few takeouts they realize it is important to make the car respond in certain way to be able to navigate it around racing circuit. This task is very closely connected with actual environment and usability of their model.

This lesson culminates in a competition. We measure the time needed to finish the racing circuit and give penalty seconds for bumping into circuit borders.

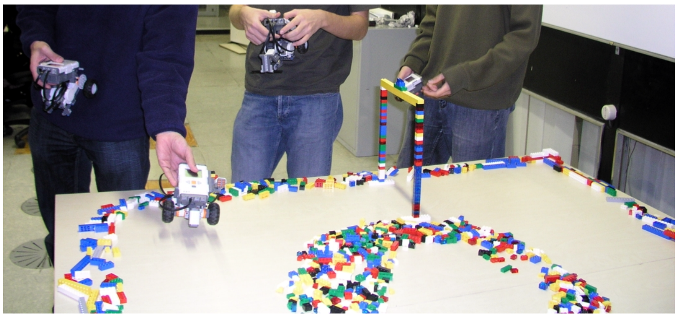


Figure 2. Testing the communication between robots on the racing circuit

Realizing short contest with remote-control cars is our reaction to recent feedback from students who have missed challenges and opportunities for the competitions besides the exhibitions (organizing exhibitions is strongly encouraged in [13]). Still, some groups of the students are not interested in the possibility to compare with the others at all. We assume that teachers should provide opportunities for exhibitions as well as for the contests in robotics classes, in order to suit the learning style of most students.

VI. Our Project: a LIFT as Fast as Possible

In the winter term of 2007 we introduced the open ended project-like activity named The Lift Model. At the end of this term the activity was rated as most popular among the students.

The assignment for the students:

Design the structure and devices of functional lift,

- build the frame tower, the cabin and some mechanism that will pull the cabin,
- program the sensors to act as lift controls (e.g. the touch sensor can be a button).

Your task is to build a functional model of a lift. You will use LEGO NXT kit, sensors and programming language. You decide how the frame will look like and how the lift will be controlled. Your model should be able to go up and down as the user needs. Note that

- the frame should be high enough,

- the frame should be steady and should not lean to the sides,
- the cabin should not tilt or spin, it should have the most stable position,
- if the cabin is too heavy, use right gears to lift it although at lesser speed.
- Challenge: try to build the fastest lift possible, is your lift faster than models built by other groups?

The material for the students included also some reference photographs of various lifts, pictures of the lift that was built by us and a list of LEGO bricks we have used in our model. We couldn't bring the actual LEGO model since all kits were used by the students and there was no spare kit for the model. According to [13], for students it is very inspirational to see sample models, they have more ideas and identify the problems they are about to solve easier. As there is often a problem with material and many teachers don't have spare kits we suggest using photographs and videos instead.

Originally we expected the project to take 3 lessons (each lasts 90 minutes), but in the process we realized 4 lessons are needed to finish all the work and to present the models.

Three out of four teams finished their models so they were fully functional. Two of those models used double motors at top of frame to pull the cabin. The third solution was a single motor placed at the base of tower. As we anticipated the highest tower was also the most unstable. It is noteworthy that all teams placed some LEGO figures or other decorations on their models (one of the girls even brought her own bricks from home). This indicates that the students should have the opportunity to use additional decorative materials to enhance their finished models.

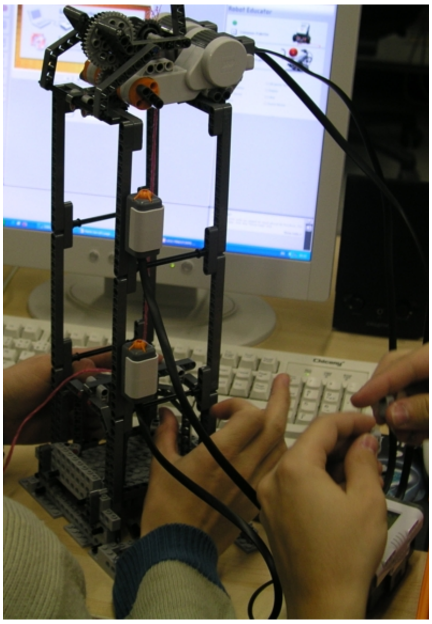


Figure 3. A lift controlled by two touch sensors. The cabin is lifted by two motors placed on the top of frame.

This activity was open-ended but it's nature didn't leave much space for students own inventions and creativity. They also voiced this opinion in the final interview. In next courses we introduced more theme based and even more open-ended activities, though we think there are students that need more guidance and instructions during deciding what they are going to design and build. We suggest that the creative robotics principle "focus on theme" [13] is a good way to give students the basic layout of what they are going to do, but the teacher should focus on the less skilled and less creative teams and help them to find more tangible model description - this can be done via guided discussion. We strongly agree that the process of finding the problem is equally or even more important than actually solving the problem.

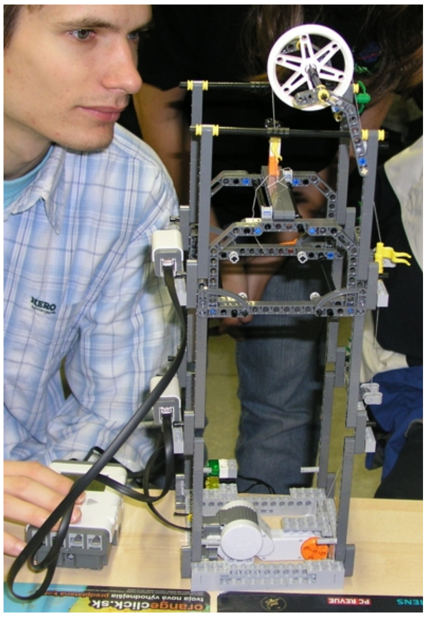


Figure 4. A lift using light signals

At the end of this activity we let the teams to present their models. Unfortunately we were not able to evaluate the challenge for the fastest lift. This should have been done and according to the final interviews students also expected some competition like activities. We think that it's reasonable to include both types of evaluation - competition and exhibition, as they appeal to different students. In case of our seminar the need for competitions is given by specific target group (most of the course participants are computer science, mathematics and management students and males).

VII. Discussion

We have described design of our robotics course for pre-service teachers and computer science students. After several iterations of whole course and introducing various types of assignments we propose that at the beginning students should solve smaller close-ended tasks with basic robotic model they do not build. This way they can learn about programming the robotic model and experience basic principles of event driven robot controlling. In later lessons it's reasonable to give students more space to create their own model and follow more constructionist lesson design with plenty of time for experiments.

Three lessons we detailed suggest that different amount of guidance and instruction is appropriate for various activities.

We have also applied some principles of creative robotics [13] and we argue their relevance for our specific target group.

In conclusion there are some relevant issues that should be considered when teaching robotics computer science students and pre-service teachers, in our course design we try to address them.

References

[1] Alimisis, D., Moro, M., Arlegui, J., Frangou, S. and Papanikolaou, K. , "Robotics & Constructivism in Education: the TERECoP project". In Kalas, I. (ed.) Proceedings of EuroLogo 2007. Bratislava, 2007.

[2] Barnes, D., "Teaching Introductory Java through LEGO MINDSTORMS Models." In Proceedings of the 33rd SIGCSE technical symposium on Computer science education, 2002, pp. 147 - 151

[3] Carnegie Mellon University, Professional Development Robotics Academy Classes, available 31.5.2010 at http://www.education.rec.ri.cmu.edu/content/educators/professional_dev/index.htm

[4] Carnegie Mellon Robotics Academy : Introduction to Robotics; Using the NXT. Electronic learning material, 2006

[5] Flowers, T.R., Gossett, K. A., "TEACHING PROBLEM SOLVING, COMPUTING, AND INFORMATION TECHNOLOGY WITH ROBOTS." Journal of Computing Sciences in Colleges archive Volume 17 , Issue 6 , pp. 45 – 55, May 2002

[6] Hoy, P., Webster, P., DiMarzio, C., Batterson, D.F. & Perkins (2006): "ROBOTICS CURRICULUM: USING LEGO® MINDSTORMS® ROBOTICS KITS." Pennsylvania State University: Youth Engaged in Technology (YET) Programme.

[7] Kabátová, M., Pekárová, J., "Learning how to teach robotics." Paper accepted for Constructionism 2010 conference. In press.

[8] Kumar, A. N. "Three Years of Using Robots in an Artificial Intelligence Course – Lessons Learned.". In ACM Journal on Educational Resources in Computing, Vol. 4, No. 3, Article 1, September 2004.

[9] Lau K. W., Tan H. K., Erwin B. T., Petrovic P. (1999) "Creative Learning in School with LEGO Programmable Robotics Products." In Proceedings to Frontiers in Education'99, IEEE CS Press, pp. 12D4/26 - 12D4/31 vol.2.

[10] Lawhead, P.B., Bland C.G., Barnes, D.J., Duncan, M.E., Goldweber, M., Hollingsworth, R.G., Schep, M., "A Road Map for Teaching Introductory Programming Using LEGO© Mindstorms Robots". In ACM SIGCSE Bulletin Volume 35, Issue 2, pp. 191 – 201, June 2003.

[11] Mauch, E. "Using Technological Innovation to Improve the Problem-Solving Skills of Middle School Students Educators' Experiences with the LEGO Mindstorms Robotic Invention System." In The Clearing House, Vol. 74, No. 4, pp. 211-213, Mar. - Apr., 2001

[12] Papert, S., "The Eight Big Ideas of the Constructionist Learning Laboratory". Unpublished internal document.

[13] Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. "New Pathways into Robotics: Strategies for Broadening Participation." In Journal of Science Education and Technology, vol. 17, no. 1, pp. 59-69, 2008.

[14] Sklar, E., Eguchi, A., "Learning while Teaching Robotics." In The AAAI Symposium Series: Accessible Hands-on Artificial Intelligence and Robotics Education, 2004.

The seminar webpage is available at http://edi.fmph.uniba.sk/lego and it contains lessons assignments (in Slovak), answers to some of them written by the students as well as the picture gallery of seminar work.

Centrobot Portal for Robotics Educational Course Material

Richard Balogh[1], Adrian Dabrowski[2], Walter Hammerl[2], Alexander Hofmann[3], Pavel Petrovič[4], Ján Rajníček[4]

[1]Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 81219 Bratislava, Slovakia, richard.balogh@stuba.sk,

[2]Höhere Technische Bundeslehr- und Versuchsanstalt Spengergasse, Spengergasse 20, 1050 Wien, adrian@atrox.at, hammerl@spengergasse.at,

[3]Department of Computer Science, University of Applied Sciences Technikum Wien, Höchstädtplatz 5, 1200 Wien, alexander.hofmann@technikum-wien.at

[4]Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia, ppetrovic@acm.org, jan.rajnicek@st.fmph.uniba.sk

Abstract—**This paper describes a prototype built as part of the Centrobot project. We present a web application intended for sharing robotics educational material. The system will allow browsing and editing a large set of projects, which we call robtivities (robotic activities). The aim of the portal is to supply information, robotivities and metadata, which can be used by teachers or pupils to educate and to learn. It will be the place, where users can discuss and improve their skills, find useful material and educational procedures: http://portal.centrobot.eu/**

Keywords - robotics education, portal, robotics projects

I. Introduction

The amount of experience collected in various applications of robotics technology in education in our and other groups has reached a critical mass, e.g. [1,3,4,5]. However, results from pilot studies, didactic materials, lesson plans, laboratory exercises, instruction sheets, manuals, simple and complex student projects, ideas for activities, links to suitable contests and other activities need to be easily accessible by the educators. Otherwise, the construction sets and educational robotic kits purchased, leased or borrowed by individual educational institutions are likely to finish locked up in a cabinet. Consequently, they would never be used properly and efficiently.

To support this argument, let us compare the number of schools that obtained the robotics sets LEGO Mindstorms Robotics Invention System in Slovakia through the governmental project Infovek from the year 2000 with the number of teams participating in robotics competitions RoboCup Junior and/or FIRST LEGO League. More than 120 elementary and secondary schools received the sets. However, only about 15% of them (18) participated in the competitions. All of the schools were invited, but more than half of them is not using the sets and does not respond to letters and e-mails from the sets distributor.

The reason is obvious. Even though the sets are a wonderful tool for interdisciplinary constructionist learning, there has been a lack of didactic materials and those existing have not been sufficiently accessible. The producer of the sets has made a pensum of projects and materials available, many of them are contributions of the community, especially in the most recent period. However, a large amount of material and activities is created by other third parties, and particularly by the teachers. Material is produced in local languages, and tailored for the needs, learning style, and common sense of the pupils and teachers from the specific region.

Most authors harness the potential of the modern media, especially the Internet, and they use the local websites, or general-purpose publishing sites to make their achievements available. The missing piece is the integration of these resources into one accessible location, where the resources would be classified, structurally and topically arranged, and provided in a standardized, understandable, and easy to use format. Centrobot portal aims to provide this missing piece. At the first place, it is focused on collecting the materials from our geographical area – Austria and Slovakia. However, the portal is open and all parties interested to share are invited and welcome.

In the following sections, we will describe the functionality provided by the Centrobot portal, explain its structure, and the whole concept. We have also implemented and describe a prototype that provides most of the intended functionality.

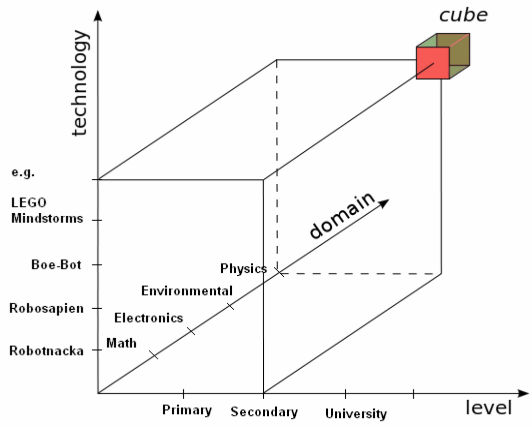


Figure 1. The Centrobot cube concept.

II. A General Idea

The Centrobot Robotics Educational Course Material Portal is a website for browsing, publishing, editing, and deployment of robotics educational material. It consists of individual units that we call *robtivities*. These are all kinds of activities, lesson plans, projects, quizzes, contests, etc. Robtivities are classified along the three main axes: *level*, *technology*, and *domain*. A particular location described by the three coordinates along these axes typically contains several robtivities. We call such location *a cube*, see Fig. 1. One robtivity can exist in multiple cubes.

By *level*, we mean the age of the robtivity audience. Level spans from the preschool education – kindergardens (up to age 7), through elementary schools (8-10), lower and upper secondary schools (10-14, 15-18) , and colleges (>18). Furthermore, the respective levels are structured further, shaping groups of beginners, intermediate, and advanced learners, depending a) on their level of knowledge and b) their level of expertese with particular technology.

By *technology*, we mean a classification of the various technological robotics hardware and software platforms used in the schools. For instance, LEGO MINDSTORMS NXT sets is one of the hw platforms, yet, it can be used in connection with many different software platforms, such as RoboLab, NXT-G, NXC, Java (Lejos), NXT logo, etc. Other hw platforms may include Parallax BoeBot, our prototyping platforms Sbot [9] and Robotnacka [4], robots Bee-bot, Probot, Roamer, MaVIN, Asuro, Aldebaran Nao, or RoboNova, for instance. Other sw platforms may include Basic-Stamp, AVR Studio, Imagine Logo, Spin, Microsoft Robotics Studio, and many other.

By *domain*, we mean the application area, school subject, or field of study. Examples are physics, mathematics, introduction to programming, mechanical engineering, or artificial intelligence at the coarse level. At several levels of higher detail, the robtivities can be classified into particular subfields, such as constructive geometry, frequency and period, mechanical wave motion, acceleration, search, heuristics, localization, etc.

For each robtivity, the specific coordinates along the three axes form its main classification. However, each robtivity can be assigned a set of general keywords in addition. The keywords can be arranged in hierarchies.

Moreover, a general-purpose material is typically relevant for all robtivities that share the same coordinate along some axis (battery charging procedure, Java language manual, etc.). Such shared materials may be placed to the container associated with the specific axis category, instead of replicating it in each and every one robtivity sharing that coordinate. The shared materials are then directly available when viewing any of the relevant robtivities.

The purpose of the classification of robtivities into the cubes is to allow for a possibility of systemmatic browsing. A teacher planning to setup a course may want to acquire an overview of the availability of the various robtivities relevant for his course, before dwelling more deeply into details of the planning and the implementation phases. In order to support this scenario, all robtivities must also contain the basic information, we describe below.

III. Robtivities

A robtivity footprint in our system consists of different information that can be valuable to the teachers when planning for their course, browsing the portal and selecting a robtivity. For lucidity and better comprehension, this meta-information is arranged in a standardized structure with five main categories: Organizational Parameters, Implementation, Technical Parameters, Support, and Resources. Registered users have the option to download all material of one robtivity for off-line use in one archive. The cube coordinates and didactical information are compulsory parts, the remaining elements are optional, but it is highly recommended that they are filled in.

A. Organizational Parameters

The parameters describe the first pieces a user needs to learn when assesing whether a particular robtivity is suitable for their aims.

Cube coordinates

Technology – e.g. LEGO Mindstorms NXT and NXC programming language;
Domain – e.g. Mathematics – Number Theory – Fractions;
Level – e.g. Lower Secondary – Introductory Level – No Previous Technical Skills;
In addition to the three main coordinates, the robotivity also has its *Language* – e.g. English, optional pointers to other language variations, *Version* – e.g. 1.0, and *Author(s)* – who developed it and who are responsible for the entry.

Didactical Information

Content – a short description of the course content and targets;
Required knowledge – a general or special knowledge required by students or instructor ;
Time consumption – for each of the activities: preparation by instructor, theory lesson, practical session, postprocessing.

Related Robtivities – lists other relevant robtivities:
Robtivities with preparational content;
Robtivities with similar content;
Robtivities with advanced content;

B. Implementation

If a robtivity appears suitable, a more detailed examination is needed to learn whether it can be implemented in the prospective lesson or course.

Course preparation

Environment – e.g. classroom, lab, mountain side, …
Equipment required – hardware, how to source (supplier links), measurement equipment, software development environment;
Presentations – material used for explaining the course content, aim, theory to students (format: PDF, PPT, optionally other formats, such as DOC, ODT);
Papers – any material used by students during the course to reach the targets, forms to be filled-in, etc.

Proceeding

Description of how the lesson is carried out – how many students, is it a team work or individual work, how to setup and start, how and what is to be observed, measured, reported, questions that are to be answered;

Sample solution – examples of hardware description, examples of code, pictures.

Multimedia artifacts (Audio, Video, Images) – in standardized formats.

C. Technical parameters

Robotics involves a complex technology and thus in addition to organizational and proceeding didactical information, robtivities provide a special place for non-didactical technical information that are or may be needed for successful deployment of the robtivity.

Construction manual
A general explanation of how to build the robot for this course, what is its main function, degrees of freedom, special requirements, modes of operation.

Description of components
All components that the users need to manipulate, e.g. sensors, microcontroller, power supply, software environment. For each relevant component, its user interface, interconnection parameters, and all technical details can be documented.

D. Support

Forum for discussions
Simple forum for exchange of experience, interesting observations during the lesson, problems and typical solutions (input to FAQs), etc. The authors are automatically notified by e-mail, when new questions appear.

Contact
Allows sending a direct e-mail message to the authors.

FAQ
Is a list of typical questions and answers that is maintained by the robtivity authors.

Rating and Feedback
A simple schema for grading the robtivities. The users who downloaded the robtivity as a single package, and those who selected it for their course are notified to provide rating and feedback after one week.

E. Resources

Robtivity resources
This section lists all referred publications, links, and all other resources that are important or useful for the robtivity.

General resources
Information that is useful for all robtivities that share the same technology, level or domain, i.e. user and programming guides, general didactical methodology, scientific resources, handbooks, textbooks, encyclopedia, and other. They appear in this section based on the robtivity coordinate classification automatically. Users may add new items and specify the respective scope of relevance for each item.

IV. Typical Use Scenarios

This section demonstrates selected most typical use sequences of actions when working with the portal. Let us first explain the user types the system supports:

guest, usually an unauthorized teacher, student, or another visitor, who can view almost all the published content, search, browse, and participate in forum discussions, the only exception is viewing the sample solutions, which are accessible only to authorized teachers;

authorized teacher, who can create new robtivities, edit the robtivities he authored, and administer the content contributed by others to these robtivities; authorization is also required for downloading a cube or a robtivity for off-line use, and for uploading media, solutions, and resources;

authorized student, who have the same priviledges as unauthorized student, with the exception of viewing the page that was created for his/her class or course by his/her teacher;

administrator, who can perform site maintenance, moderate all discussions, and edit or delete content.

A. Anonymous user searching for useful content

- select coordinates, alternatively select keywords
- view the resulting list of robtivities
- display the robtivity details one by one
- view individual documents, pictures, videos
- find one robtivity that is the most suitable, copy the robtivity URL for use in her course

B. Teacher preparing a course based on several robtivities

- create a teacher account, or login to an existing one
- be confirmed by another teacher or admin, if needed
- create a virtual classroom – an account for student authorization
- search for robtivities, and add them to the virtual classroom
- provide the student account name to the student in her class
- teachers can create copies of their virtual classrooms for repetitive use in multiple classes
- in each virtual classroom, the teacher can view the solutions submitted by the students

C. Teacher adding a new robotivity

- login to the system, or create a teacher account if new
- selects the most suitable cube, and eventually adjust the categories along one or more axes
- add a new robtivity, upload all relevant files, provide the obligatory and optional parameters
- can take a break and come back later to complete
- can add more authors who can edit all the robtivity content
- when the robtivity editing is finished, the author marks it as published

D. A student works with a specific robtivity

- logs into the virtual classroom as instructed by the teacher
- arrives at a list of preselected robtivities and a commentary prepared by the teacher
- can browse the whole site, if appropriate
- may upload his results for later inspection by the teacher

E. Admin performs the site maintenance

- logs into the system
- can run broken links detection, view their list, and respective pages, correct the links manually
- can configure the system backup and recovery

- can administer accounts, forums, edit or delete or robtivities
- can send an information e-mail to authors

V. Robtivity Life Cycle and System Maintenance

Robtivities are prepared by one or several authorized teachers during a period of time, when no other users may access them. Once the robtivity is completed, one of the authors may mark it as published, and it becomes part available at the portal for all the users, who can then rank the robtivity by assigning certain number of stars, add feedback in form of forum comments, or questions, upload their sample solutions, and resources, or send message to the authors. The authors can further edit their robtivity, add, change, or delete the content as appropriate, answer forum questions, appoint other authors who can continue maintaining the robtivity. Administrator always has the rights to remove inappropriate content, delete the whole robtivity, or deny access of some author, if needed. In normal circumstances, the administator will first contact the author to perform the changes. If all authors do not respond for very long period of time, and the information in some robtivity becomes obsolete, the administrator may choose to add a new author, who will continue maintaining the content. Eventually, outdated robtivities can be marked as outdated or possibly removed from the site.

The system performs a system log of all operations that are performed in the system – for both debugging, tracking and maintenance reasons. The log is saved into database, and can be viewed by the admin.

The system is designed to work in multiple languages, English, German, and Slovak versions are available, other can be added when needed.

VI. Example Robtivities

In this section, we provide two example robtivities as they are published in the Centrobot portal and verified in the classroom. The first one is using the popular NXT construction sets, while the second utilizes Sbot educational robotics platform [9] and used in the exercises of an introductory college course on robotics. For better readability, the information is presented in tables. In Centrobot portal, the information is shown in a structured format and viewed in a webbrowser.

A. Line-Following NXT Robot

Organizational Parameters	
Cube Coordinates	
Technology	LEGO MINDSTORMS NXT and NXT-G
Domain	Introduction to Robotics – Simple applications
Level	Lower secondary school – Beginners – No technical skills required
Version	1.1
Language	English
Author(s)	Walter Hammerl

Didactical Information	
Content	Building and programming a robot, which is able to follow a black line
Required knowledge	NXT-G basics
Time consumption	Preparation by instructor: 1 hour Theory lesson: 30 minutes Practical session: 2 units 50 minutes each Postprocessing: 30 minutes
Paths	◄ Introduction to NXT-G ► Passing a maze with NXT ► Spatial orientation
Implementation	
Course preparation	
Environment	Classroom with enough space to layout the line follower course
Equipment required	- 1pcs LEGO Mindstorms NXT per student - bright plane with a closed black line (e.g LEGO test pad 8547) - sheet of paper with a black line (2-3cm) - stopwatch to compare the individual performance - 1 PC with NXT-G Software 2.0 per student - 3 light sensors per NXT Alternatively, the lesson can be hold with groups of 2 students per robot
Presentations	(link to Line-follower presentation)
Papers	(link to Spreadsheet to record the lap times), real-world application of the line-following robots [6]
Proceeding	
Description of how the lesson is carried out	The instructor may start with a brief introduction to the theory of line following and eventually a review of NXT-G programming. The students shall build their line following robot themselves. In case they start from scratch the manual for building the basic NXT model is helpful. Students shall be encouraged to think about different ways to mount the sensors and to build their individual model. Discuss mechanical parameters for mounting the sensors: - distance between the sensors - clearance from the floor - distance from drive wheels, etc. Before starting to program the robot, the students shall provide a flow diagram, which may be reviewed by the instructor. Start programming the robots and test using the test pad. Good practice is testing the correct function of the robot by placing it on top of a support box, the wheels can move freely *(the table continues below...)*

	(continues from above...) without floor contact, and a piece of black and white paper. Record the time the robot needs to complete a given number of laps. Encourage the students to modify and improve the algorithm.
Implementation	
Sample solution	3D model of the robot, example of code: linefoloower.rbt, pictures: myrobot.jpg
Multimedia	Video: linefollower.avi
Technical parameters	
Construction material	
How to build the robot for this course	main function: Simple robot with 2 degrees of freedom, special requirements: 3 light sensors
Description of components	Easy start model, Line sensors LEGO Mindstorms user manual
Support	
FAQ	
Question	When I program the motors to move with *unlimited* duration they stop after a few seconds. What shall I do?
Answer	The program will set the motor operation to unlimited, the motors will speed up, and then the program will carry out the next command. If this is the end of the program the motors will stop. The simplest way to is to put the MOVE block within a LOOP block set to control=forever. See also the presentation material.

B. Bayesian Robot Programmiing with Sbot

Organizational Parameters	
Cube Coordinates	
Technology	Sbot and AVRStudio
Domain	Artificial Intelligence – Bayesian Robot Programming
Level	College – Intermediate students – Intermediate Sbot users
Version	1.0
Language	English
Author(s)	Pavel Petrovič
Didactical Information	
Content	The students will study learning in a robot with a probabilistic model of the world. The robot will learn simple behaviors of pushing and following objects.
Required knowledge	Theoretical framework of BRP, Bayes fomula, basic operation of Sbot robot
Time consumption	Preparation by instructor: 1 hour Theory lesson: 90 minutes Practical session: 90 minutes Postprocessing: 30 minutes
Paths	◄ Introduction to SBot ► Sbot localization
Implementation	
Course preparation	
Environment	Laboatory with a small arena and rectangular obstacles
Equipment required	- 1pcs Sbot per group - flat surface and obstacles of various shapes - a PC with AVRStudio development environment and BlueTooth connection
Presentations	(link to presentation Introduction to Bayesian Robot Programming)
Papers	(link to a journal paper on BRP) (link to detailed instruction, source, report)
Proceeding	
Description of how the lesson is carried out	Students first control the robot remotely for a short period of time. The program samples the sensors and builds a probabilistic representation of the behavior policy. When the robot runs in autonomous mode after the learning session, it should successfully push the obstacles. Students repeat the experiment with following the obstacle walls. Students modify the code based on instructions and observe, measure, document the outcome.
Implementation/*Proceeding* (cont.)	
Multimedia	Video: sbot_brp.avi
Technical parameters	
Construction material	
How to build the robot for this course	The robot is already built. Students only have to make sure the distance sensors are properly mounted and connected. The robot schematic and software framework is described in the user manual (link).
Support	
FAQ	
Question	If the robot is connected to a port > COM8, the terminal program cannot connect.
Answer	This is because the serial ports from COM9 in Windows do not exist. Instead, Microsoft invented the names \\.\COMXX, such as \\.\COM42. Some programs require this format for ports higher than COM8.

VII. Prototype Design and Implementation

Our prototype has been specified and designed by the partners participating in the Centrobot project and the first version was implemented by a bachelor student at FMFI UK in Bratislava, Ján Rajníček [2]. Detailed use-case, sequence, and class diagrams specified in UML can be found there, or in the Centrobot internal technical documentation.

A. Technology

Our prototype is platform-independent, but it runs on a Linux server machine with Apache, MySQL, and Tomcat server installed. It is based on JSP and Java servlets [7], XSL-FO, XML, and MySQL with Hibernate framwork, and PHP for forums on the server side, and HTML, CSS, and JavaScript (AJAX) [8] on the client side. It relies on third-party components Log4j for logging, and TinyMC for WYSIWYG editing. The backup is realized with crontab and bash shell scripts. This setup allows for interactive graphical user interface in web browser without being bound to the traditional request-response HTTP model, i.e. the scripts running on the client side are in a permanent communication with the server-side scripts and database without the need to reload the pages.

B. Design

The system architecture diagram is shown in figure 2.

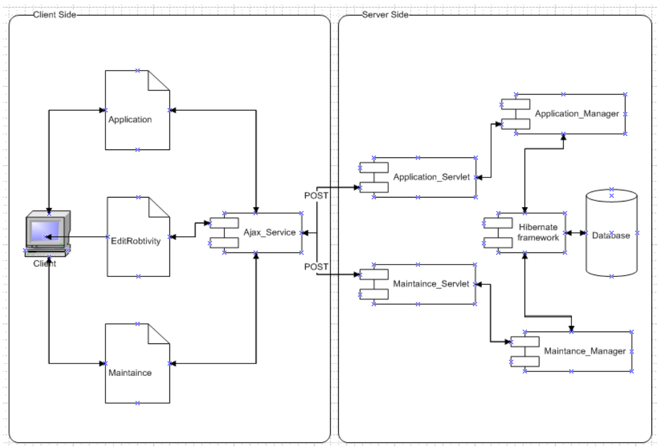


Figure 2. Architecture of the portal.

The system consists of 9 modules and a database. The *Application module* is a webpage with scripts running on the client allowing the users to login, or create their account and edit the profile, create, show, search or rank robtivities, access the robtivity forum, download cubes, upload content, setup, login, view, and delete virtual classrooms, and log all activity. *EditRobtivity module* is based on TinyMCE, and allows editing the various contents of the robtivities. Links to the files that were uploaded can be added to the text. *Maintenance module*, accessible only to admiinistrator provides all admin functionality as described above. The last module that runs on the client side is the *Ajax_Service module*, which mediates the communication with all server modules using the standard XMLHttpRequest object. The server-side resembles this structure. *Application_Servlet module* processes the requests from Application, and passes them further to Java package *Application_Manager module*, while the *Maintenance_Servlet module* processes the admin's requests and calls the Java package of the *Maintenance_Manager module* to perform the server-side admin's functionality. Both manager modules utilize the *Hibernate framework module,* a database interface.

VIII. Conclusions and Future Work

We have designed and specified a detailed concept for educational robotics web portal containing set of various robtivities, i.e. projects and activities for classrooms and/or after-school clubs and centers. A prototype of the portal has been implemented and deployed at http://portal.centrobot.eu/ also available through Centrobot project. Initially, it collects robtivities from our geographical locations in Austria and Slovakia. It is open for a new content provided by third parties.

Our aim is to run ensure a longterm operation of the portal, convert all the materials we collected and created in the past so that they will be available for general public. This involves fixing, debuging and improving the functionality, staying in contact with the content authors, collecting and processing their feedback. For instance, we would like to add a possibility to create on-line questionnaires or simple quizzes that could be filled-in by the students and automatically evaluated by the system based on the data provided by the teacher. The current search capabilities could be extended to provide an advanced and intelligent search options, and automatic filling of terms in the search form

Acknowledgement

Project Centrobot (ATMOS N_00023) is supported within the Slovak-Austrian cross-border cooperation programme 2007-2013 'Creating the Future' financed by the EU European Regional Development Fund.

References

[1] T. Kraft, A. Hofmann, "V*ienna Cubes – Robocup F180 Small Size Legue*, *Team*", ÖGAI Journal 03/2003, Österreichische Gesellschaft für Artificial Intelligence, pp 14 – 19, 2003.

[2] J. Rajníček, "*Portal of educational robotics for the Centrobot project*", bachelor thesis, FMFI UK Bratislava, 2010 (in Slovak).

[3] R. Balogh, "*Basic Activities with the Boe-Bot Mobile Robot",* DidInfo 2008, 14th International Conference. FPV UMB, 2008.

[4] P. Petrovič, R. Balogh, A. Lúčny, R. Weiss, "*Using Robotnačka in Research and Education",* Eurologo 2007 Conference, FMFI UK, 2007.

[5] P. Petrovič, A. Lúčny, R. Balogh, D. Ďurina, "*Remotely-Accessible Robotics Laoboratory",* Acta Mechanica Slovaca, Vol. 10(2-A), pp. 389-194, 2006.

[6] J. Svetlík, M. Sukop, "Robocar - mobilný robotický vozíkový systém," Acta Mechanica Slovaca. vol. 12, no. 2-a pp. 617-624, 2008.

[7] N. Z. Zakas, Professional JavaScript for Web Developers, Wiley, 2009.

[8] B. Burd, JSP: JavaServer Pages, M&T Books, 2001.

[9] Guštafík, David and Krasňanský, Pavol: Robot SBot 2.0. Student scientific conference, FEI STU, 2008, (in Slovak).

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

How to Make a Good System from Imperfect Components

Andrej Lúčny
Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics, Comenius University
Bratislava, Slovakia
lucny@fmph.uniba.sk

Abstract — **We present a control system of LEGO robot which is compounded from several different and competitive behaviors. It has been used for training of master course multi-agent systems to present to student a real example how a good overall behavior can emerge from imperfect individual behaviors.**

Keywords – robot, behavior, education

I. INTRODUCTION

Cheap and widely available robots are popular teaching aids. They are not only attractive for students, but they enable teacher to show many ideas from many domains in real operation – not only those ones tied with education of robotics. Of course, teacher has to invest his effort to such presentations and exercises and must have enough knowledge and/or effective support from field of robotics.

We present exactly such approach here: we use simple LEGO robot to demonstrate profit of agent-oriented programming. Agent-oriented programming is a method of software development derived from field of multi-agent systems. While multi-agent systems are necessarily distributed systems, agent-oriented programming applies the same modularity and organization for development of non-distributed systems. Good example of such system is a system for controlling a robot. This gives us opportunity to employ robotics in education of multi-agent systems.

The main profit of multi-agent systems (besides handling of distributed character of their applications) resides in ability to generate better overall behavior of system than the behaviors of its individual components can provide in total. (This ambition makes domain of multi-agent systems close to artificial intelligence). It is relatively difficult to demonstrate this idea hence range of usual applications typical for multi-agent systems overcomes conditions which teacher can emulate at classroom. Therefore we employed robot controlled by system with analogical modularity and organization. This enabled us to show the profit very clearly.

II. TECHNICAL ACCOMPLISHMENT

A. Hardware and software

We have used LEGO Robolab. We have build robot typical for line following equipped with two motors and tracks, one light sensor and one touch sensor; all controlled by RCX LEGO brick (Fig. 1). We have used Lejos firmware and development kit for RCX (http://lejos.sourceforge.net). We employed robot scene for 'Line Follower' contest, which is part of e.g. ISTROBOT contest.

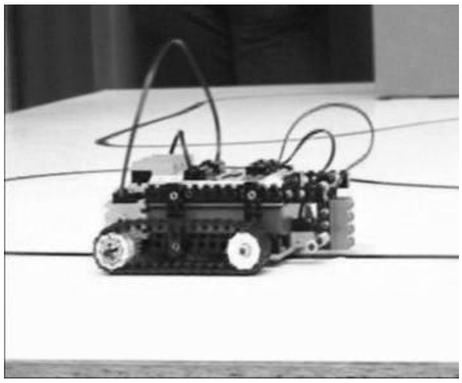


Fig. 1. Robot following line

B. Problem specification

Robot, architectural framework for building its control software and individual components for its development are given to students. Students are asked to fill the components gradually into the framework and put the robot into operation. Then the conditions in the scene are changed and students observe how robot is able to treat the modified situation. The scene difference is chosen to present that robot is operational also under changed situation but just due to multi-agent features of the employed framework.

Particularly, the framework is based on indirect communication among agents through a backboard ([3], www.agentspace.org). It enables any agent to interfere communication among other agents.

The task is similar to 'Path follower' category of popular contest ISTROBOT ([2], www.robotics.sk). Robot has to follow a line, but it has also to overcome a brick which is put on the line.

C. Problem solution

Usual solution of this problem is based on consecutive execution of two behaviors: on following the line and avoiding the brick.

This work was supported by VEGA grant VEGA 1/0280/08

Following is based on a single light sensor. We move only one track at same time. We move with a track while the light sensor detects line. Then we stop the track and start to move the other track. This simple behavior causes that the robot moves on line when the line is present and the robot stops otherwise.

Avoiding is based on blind sequence of proper movements started by touch of the touch sensor. The size and direction of the consecutive movements are selected due to dimensions and position of the brick. (It depends also on the current capacity of batteries little bit.)

The easiest way how to combine the two behaviors is a pure pipe-line (Fig. 2). This solution works only for one brick and it fails at all when we use a brick with other orientation (e.g. we turn the brick to perpendicular position).

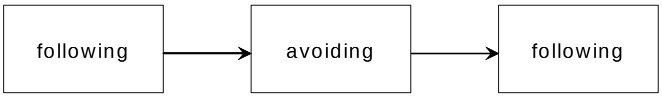


Fig. 2. Traditional combination of behaviors

A novel solution we like to present to students is based on parallel combination of the two behaviors (Fig. 3). Both Following and Avoiding are parallel modules with own control (with own threads).

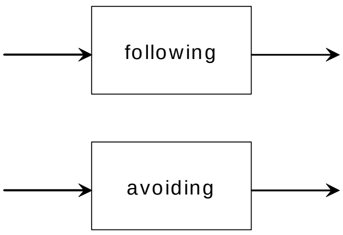


Fig. 3. Novel combination of behaviors

Following is active when line is present (and when we has just lost it), while Avoiding is active after the touch sensor is activated. Avoiding has a higher priority than Following at the moment of the touch. Its priority is lower otherwise (mainly we need to concern the moment when the line is found again). This cooperation is provided the framework.

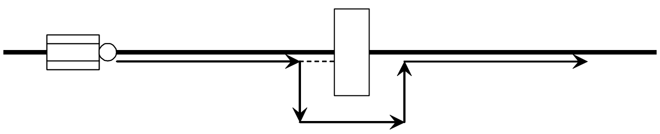


Fig. 4. Solution of the normal situation

This solution can handle the situation when the traditional approach works (Fig. 4). However, when we modify the brick orientation, it works too (Fig. 5). This fact is surprising and makes a strong impression to students showing them how our expectation can fail when we deal with systems which internal structure resemble to organization of living systems – even in such simplified case.

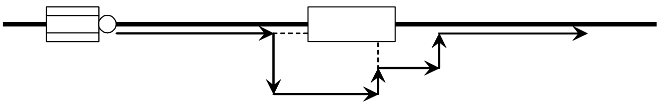


Fig. 5. Solution of the modified situation

Of course, we see no prodigy here. Just the behavior Avoiding designed for single use has been used two times due to another touch we did not expected. This is very clear when we see it, but it is hardly expectable until we see it.

In this way the non-traditional framework enabled us to create system which overall behavior overreaches the sum of capabilities of its individual components. We have built a good system from less perfect components.

III. EDUCATIONAL ASPECTS

This example is very simple, but just such examples are suitable to be shown during two hours of student exercise.

We have worked with group of 10 students (Fig. 6) divided to two groups. Lejos and windows shell extension for compiling java source files and running compiled classes (we avoided any IDE in this way) have been available for students to simplify use of equipment and to save enough time for dealing with application. At the first, students have to put to operation the traditional solution. Then they had to use the novel framework known to them also from previous exercises. Each group had a responsible person selected by teacher. One group succeeded to carry out the experiment, one failed – mainly due to lack of time. Even for this simple example time is critical and everything which has no direct relation to the main goal must be prepared in prior to exercise.

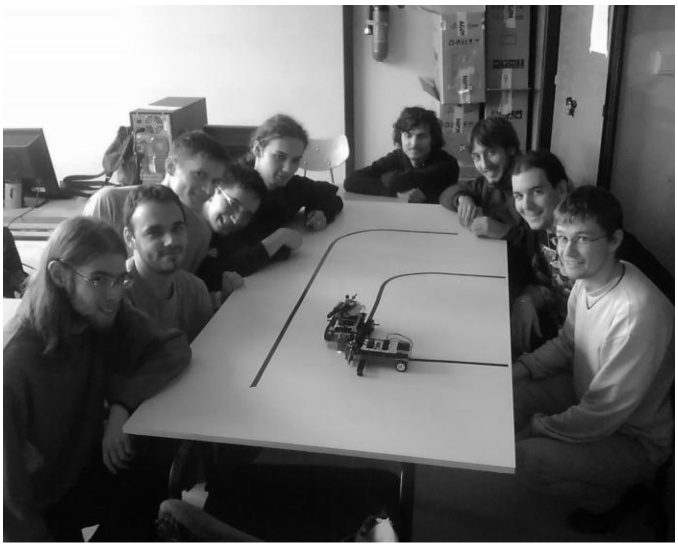


Fig. 6. Tested group of students

IV. Conclusion

We have presented one example how to employ robots in education. The aim of this employment was to show to students a profit from non-traditional organization of software. The example was tiny but suitable for student exercise. In this way we presented a principle from domain of multi-agent systems which could be hardly presented in a traditional way.

Acknowledgment

We thank to Jan Sefranek, associated professor at FMFI UK Bratislava for his long-term support of robotics in education at our Department of Applied Informatics.

This paper was created with support of VEGA 1/0280/08 “Application of multi-agent modularity for development of control systems and computer models”.

References

[1] Bagnall, B.: Maximum LEGO NXT. Variant Press, Winipeg, 2007

[2] Balogh, R.: Popularization of the robotics, In DidInfo 2008. 14th International Conference. FPV UMB, Banská Bystrica, 2008

[3] Lúčny, A.: Building Complex Systems with Agent-Space Architecture. Computing and Informatics, Vol. 23 (2004), pp. 1001-1036

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Some didactic aspects of teaching robotics

Anton Vitko, Ladislav Jurišica, Andrej Babinec, František Duchoň, Marian Kľúčik
Institute of Control and Industrial Informatics
FEI STU
Bratislava, Slovakia
anton.vitko@stuba.sk

Abstract—**The contemporary robotics is an excellent tool for teaching science and engineering and an attractive topic for students of all ages. Problems of robotics are fundamentally about the couple sensing – action, the two parallel activities bounded by the robot's dynamics. It is just the robot's dynamics that makes the relation "sense – act" difficult to control and calls for the advanced study. The fist part surveys the authors' experience as teachers and researchers in the field of robotics at the electrical and mechanical engineering faculties. The second part points out some issues of robot control and navigation as we teach them at the university level.**

Keywords-didactic aspects; mobile robotics; stationary robotics

I. Introduction

There is much more to robotics education than just teaching about robots. Robots are finding their way into the classroom so as to help teaching science, math, mechanics, teamwork and even management skills. Many enterprises rely on *off-the-job training* (formal learning) without considering its suitability for the learning tasks at hand. *On-the-job training* (informal learning) has a substantial advantage: it is more close to the problems to be solved. On other hand *on-the-job training* is often unplanned and therefore mostly ineffective. For this reason, bridging formal and informal learning, theory and practice, the abstract and concrete in robotics is the best way to convince the students at all grade levels that the robotic subjects are interesting and useful. The educators have found that teaching with and about robots provide a new and exciting way to interest and motivate their students.

At the Institute of Control and Industrial informatics in Bratislava was established the Office of Robotics Education as a way to help educators, students and parents with interests in robotic. We hope this webpage will serve as a helpful launching point.

From the perspective of teaching robotics may be useful to look at the relation between robotics and mechatronics. Some time ago I found on the Internet a scheme of mechatronic system. It revealed that the same is also true for a robotic system. The robotic system (robot) is also a purposeful connection of mechanical and electrical systems (electromechanical system) equipped with actuators through which the system acquires moving abilities. Its motion is controlled in real time by a digital controller which acts on the electromechanical system through a set of D/A and A/D converters Judging by the scheme its author probably supposed that the control program together with control data (e.g. desired motion trajectory) is loaded into the computer memory at the beginning of the working task. This may be the case of a grinding, milling or other numerically controlled manufacturing machines, which repeatedly do the same operations. Except for some force, torque or temperature sensors such simple "mechatronic" system does need any feedback from its (possibly changing) environment. Thus the scheme represented at most a classical "low level" controlled system without learning.

Complexity of current fixed or mobile robots goes much further. They are required to do tasks which go far beyond the capabilities of the classical industrial manipulators. Letting alone the sophisticated nonlinear robust and adaptive control, the primary requirement laid on modern "mechatronic" systems, which the contemporary robot undoubtedly belongs to, is ability to grasp a "mental image" of both its own state and the state its environment. Having this image (context) in mind the robot should be to improve its knowledge through learning from interactions with the environment.

From what have been said follows that the subjects of robotic cover a wide range of sophisticated problems, which require the university study. Therefore in what follows some problems of teaching the robotics at the university level will be briefly mentioned.

II. Some experience from teaching the robot modeling and control

To understand moving operation of a robot, the student must be familiar with robot kinematics, in particular the homogenous transformations, Denavit –Hartenberg parameters, problems related to the direct and inverse kinematic, manipulator's Jacobian matrix and the like. These problems are relatively easy grasped by all students regardless their previous education. Mastering the problems of robot kinematics is a basic prerequisite for understanding issues of robot dynamic.

The problems of robot dynamic are much more difficult to teach. Primary reason is that the students of electrical engineering are not sufficiently good in mechanics. So as to teach them the notions and mathematical means like the Euler-Lagrange equations, inertia matrix, expressions of kinetic and potential energy etc., the lecturer is forced to remind the basic principles of the mechanics. After doing this he/she can continue with explanation of the robot dynamics. The undergraduates of mechanical engineering are facing the opposite difficulties. They need some preliminary introduction to more sophisticated control issues.

After understanding the robot kinematics and dynamics the subject of robotics becomes much more interesting and attracts a great deal of the students' attention. The essential knowledge the students must comprehend consists in understanding the theoretic reasons why the robot manipulator (except for special configurations, e.g. SCARA robot) being controlled by linear PID controllers cannot reach an acceptable tracking performance. This knowledge is a stepping stone for presentation the philosophy of autonomous control, namely that of named as the computed torques methods.

Grasping the problems of multivariable control is again a rather demanding task. Here the most difficult is to explain principles of co-called inverse dynamics and subsequent synthesis of a robust and/or adaptive controller.

In relation to the design of a robust controller the special attention is given to the robot control based on the theory of variable structure systems (VSS). Though rather difficult, due to step by step explanations the students understand the basic principles of VSS control relatively easily and become fascinated with the possibilities the VSS control offers. In the end the strength of the VSS control is demonstrated by some results obtained with the VVS control of a flexible joint robot. They are acquainted with undesirable effects of the joint flexibility.

The syllabus ends by brief presentation of hybrid position-force control and control of mechanical impedance. It can be concluded that the subject provides students with a good overlook over the field of advanced control industrial robots.

III. Teaching mobile robotics – intelligent navigation and data fusion

A. Intelligent navigation

The robot navigation is another aspect of teaching robotics. An autonomously operating mobile robot must respond to instantaneous incentives coming from its own "body" and surrounding environment. To this end the robot needs to handle a wide range of unexpected events, detect and distinguish between normal and faulty states, classify them and finally, if the fault cannot be compensated by a nominal control it should switch to an appropriate fault-tolerating regime. To manage these tasks, the robot functionality must be organized into an appropriate architecture, i.e. a set of organizing principles and core components that create a system basis.

The control community is familiar with the term of "intelligent control", denoting the abilities the conventional control system cannot attain. Leaving alone various meanings of the "intelligent" system, some basic features characterizing an intelligent system will be mentioned here. To mention a few, it is making decisions, adapting to new and uncertain media, self-organizing, planning, and the more. [1, 2] Intelligent systems should not be restricted to those that are based on a particular constituents of so-called soft computing techniques (fuzzy logic, neural networks, genetic algorithms and probabilistic reasoning), as it is frequently done. Soft computing techniques should be considered as mere building blocks or even "bricks" used for building up a "large house" of an intelligent system. What makes a system intelligent is just a synergic use of the softcomputing techniques, which in time and space invoke, optimize and fuse elementary behaviors into an overall system behavior. For instance, fuzzy inference is a computing framework based on the fuzzy reasoning. But the fuzzy system is not able to learn and must be combined with neural networks which add the learning ability. To this end, the fuzzy rule-set is commonly arranged into a special neural architecture like ANFIS and NEFCON with Takagi-Sugeno-Kang and Mamdani inference respectively. [3] Intelligence of such system springs from successive generalization of information chunks (granules) - singular, crisp, and finally fuzzy granular information. [4, 5] Due to the information granularization a system becomes robust with respect to imprecision, uncertainties, and partial truths. Thus, the system's intelligence comes from its architecture i.e. from the inner organization of the system elements and functionalities. To demonstrate this, the *subsumption architecture* (developed in 1986 by Brooks [6, 7]) and used in the synthesis of navigation algorithms of a mobile robot developed at the authors' workplace will be briefly described.

The subsumption architecture was inspired by the behavior of living creatures and heralded a fundamentally new approach to achieving more intelligent robots. The robot behaviour is typically broken down into a set of simpler behaviours which are loosely co-ordinated towards a final goal. Behaviours having higher priority are subsumed under those with lower priorities (running at the background), thus a layered structure is developed. Contrary to the classical hierarchical architecture, in which a particular behaviour assumes control if a given set of logical conditions is fulfilled, the behaviours which are organized into subsumption architecture can appear concurrently and asynchronously and with different intensities. For example, if a robot is navigated in an unknown environment cluttered with obstacles, it is natural to assign the highest priority to the obstacle-avoidance behaviour, and lower priorities to the behaviours which are to be initialised e.g. if the robot finds itself trapped in a deadlock. Using such priority management, the robot being in a deadlock inhibits all obstacle-avoidance related behaviours. Instead, the behaviour being typical for escaping from the deadlock assumes control. In other words, the obstacle avoidance behaviour is normally "subsumed" by the deadlock-resolving behaviour. If the robot finds itself in a deadlock (e.g. in a partly closed space), the obstacle-avoidance behaviour is to some extent suppressed by the deadlock-resolving behaviour. Similarly, a striving-towards–a-goal behaviour subsumes both of them and therefore it possesses the lowest priority. An example of subsumption architecture that was used in the navigation of our experimental robot [8] is depicted in Fig 1. One reason why the highest priority is assigned to the obstacle-avoidance behaviour is that one can reasonably expect that the robot will encounter an obstacle when moving in a terrain. The deadlock-resolving behaviour (lower priority) subsumes the previous one because it is less probable that the robot will be trapped in a deadlock. These two behaviours are subsumed by the goal-striving

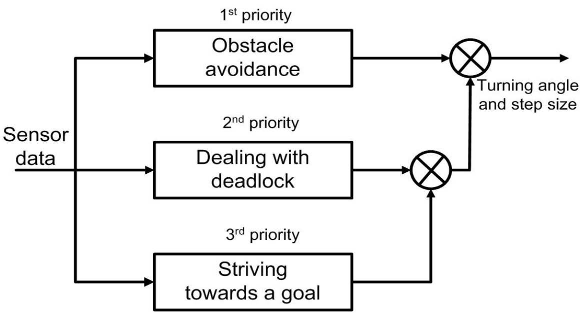


Figure 1. Subsumption architecture

behaviour (the lowest priority), because the probability that an obstacle-free landscape will appear in front of the robot is relatively low. If it happens, the goal-striving behaviour would inhibit or even suppressed both of them. The subsumption architecture is a kind of behaviour-based architectures [9].

Great advantage of the described architecture is that if implemented through fuzzy IF-THEN rules, the transition between behaviours is very smooth. In case that a transition is controlled exclusively by current sensor information, the system is called *reactive.* The reactive systems typify a majority of the autonomous robots operating in distant and unknown environments, like sea beds, battlefields, areas hit by disasters etc. The robot navigation based on the subsumption architecture is has find great popularity among students.

B. *Data fusion*

When teaching robot navigation the issues of data fusion cannot be avoided, because an autonomously navigated robot is a particular realization of an intelligent system. In this view the teaching the data fusion naturally precedes issues of robot navigation. The thing is that the robot functionality relies on numerous disparate sensors through which it grasps a consistent image of what is going on. An underlying idea of the sensor integration rests on a synergic use of the overlapping information delivered by the sensors of different types. An aim is to obtain aggregated information that would be more complex then that of received from a single sensor. The aggregated (or blended) information is beneficial at least from aspects of noise reduction and novelty extraction, which makes the data patterns hidden in raw signals more obvious.

It is stressed that a single sensor cannot provide a required amount of information. For instance, the ultrasonic range sensor used for identification of an obstacle is uncertain about the exact location of the obstacle to which the distance is measured. This is because of the wide angle of the ultrasound wave cone. Therefore there is a need to install an additional sensor, let us a laser one, which adds additional information about the obstacle direction. Another reason that necessitates the fusion, stems from the fact that mobile robot operates in changing environment; therefore the fusion must take place not only in space but also in time. Besides, the use of a set of (distributed) sensors of different modalities allows fusion of high-level information (e.g. statements) and even to grasp a context. This is to some extent, tantamount to mimicking human-like reasoning. For instance, the fact of finding a personal mine implis higher likelihood of finding other mines or even a whole battlefield (i.e. context).

The number of sensors needed for robot navigation and fault detection is relatively large. Examples include the GPS sensors, proximity sensors, odometers, accelerometers, gyroscopes, inclinometers, velocity, temperature, light and darkness sensors and many others. In order to know "what to fuse", multimodal information must be fused into a common format, and what is very important, the uncertainty of sensed and fused signals must be taken into account.

Special attention is devoted to the heretical structure of the data fusion. It is explained that at the lowest level is performed a pixel fusion of single signals or pixels. Features (mean value, variance, kurtosis, covariance, power spectrumetc.) are fused at the second level. As to the signals are of random nature, the fusion is usually based on the Bayesian statistics with Kalman filter [10, 11] as a typical representative. The aim of so called *complementary* fusion is to obtain not only accurate but also more complete information. For instance, images from two cameras looking in different directions are fused to obtain a more complex image. Another possibility is that more sensors sense the same quantity, e.g. sonar and laser range sensors. In this case the sensors "compete" in a sense, therefore one can speak about *competitive fusion.* The third kind is *cooperative fusion,* where one sensor relies on the others, (e.g. the battery state can be observed by simultaneous measuring the electric current and time).

The situation is illustrated in Fig. 2. As seen, at low levels run cooperative and competitive fusion while at high levels runs complementary fusion. Results of high level fusion are statements (declarations) about instantaneous state of the robot, saying for instance that "in the azimuthal angle "α" at the distance "d" is seen a small pond" or "the battery is discharged to 50% of its initial capacity". In general, at the lower level runs the signal fusion and at higher level runs the symbolic fusion. While a typical means for signal fusion is Kalman

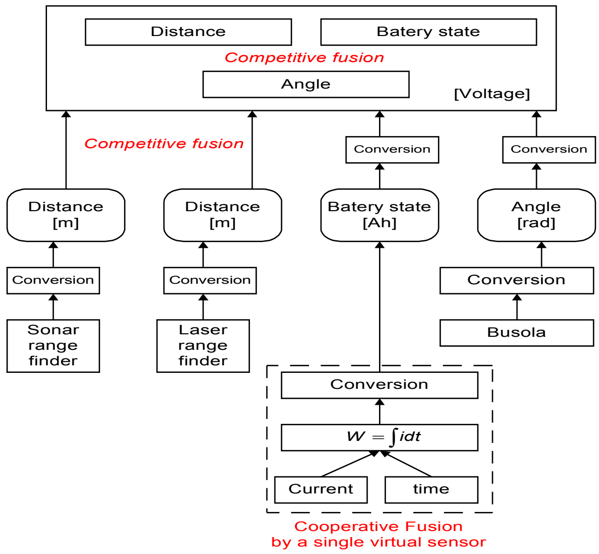


Figure 2. Types and hierarchy of data fusion

filtering, a typical means used at higher levels is either Dempster –Shafer theory of evidence [12-14] or fuzzy logic [15].

The students must become aware that results of the fusion process (at all levels) are not only estimated values (numeric or symbolic) but also corresponding *certainty values*. In case of Kalman filter the result is an estimate of the mean value and by way of the certainty value is used signal variance. Contrary to this, in case of Dempster-Shafer evidence theory the output is a symbolic value, supplemented by its *belief value*. Finally, in the case of fuzzy fusion, the output is the consequent of the fuzzy rule, supplemented by corresponding *degree of fulfillment* (firing strength). In the end of semester some means of data fusion are explained.

1) Example of low level fusion

Let us suppose that the random signal x with normal distribution is directly measured by two different sensors S_1 and S_2. The estimates are x_1, x_2, and their certainty values are given by the standard deviations σ_1 and σ_2. An optimal estimate X is then obtained by fusing the measurements in accordance with the rule

$$X = x_2 + \left[\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right](x_1 - x_2) \tag{1}$$

Variance σ^2 of the fused estimate X is given by the expression

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \tag{2}$$

The fusion process can continue repeatedly in such a way that the estimate X is considered as if it would be a new reading of the sensor S_1, that is x_1. The sensor S_2 performs the next measurement with the reading x_2, which is again fused with x_1. In this way the variance σ^2 gradually decreases while the preciseness of the estimate X is gradually improved.

2) Example of high level fusion

The high levels are occupied with more sophisticated procedures of notion identification, i.e. "what was observed" and "what it means to have observed that". The higher level is a domain for application of *possibilistic approaches*, which can directly handle symbolic quantities, e.g. propositions. Every proposition is accompanied by its certainty value (score), which expresses how certain the sensor is about its estimation of the measurand. Examples of fused propositions:

$z_{i,e}$ = there is a cube "i" in the robot's environment "e"

$z_{i,c}$ = object "i" belongs to cluster "c"

$z_{d,\alpha}$ = at angle "α" there as an obstacle at the distance "d"

Higher-level fusion is based either on *Bayesian statistics* (not mentioned here) or possibilistic means, like Dempster-Shafer evidence theory and fuzzy set theory but even a short recapitulation goes beyond this paper.

IV. CONCLUSIONS

Some didactic issues with a brief indication of syllabuses of stationary and mobile robotics taught at the university level were presented. Both the syllabuses and teaching experience as described here cannot be generalized. Every teacher can appropriately modify of them so as to reach the best educational results. The practical laboratory activities were not described. In general, the computer simulations of robot dynamics, control and navigation are supplemented with experimental measurements and control of both industrial manipulators and mobile robots.

The students' understanding of the robotic problems presented during the lectures and within laboratory activities can be considered as very acceptable. The graduates can easily join the robotic and related companies.

ACKNOWLEDGMENT

Support of the grant VEGA No. 1/0690/09 is duly appreciated.

REFERENCES

[1] Fu K.S., Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control. IEEE Trans. AC. Vol. 12, No 2, pp 17-21, 1971.

[2] Meystel A., Messina E., The Challenge of Intelligent Systems. Proc. of IEEE Inern. Symp. On Intelligent Systems (ISIC 2000), Rio Patras, Greece, 2000.

[3] Nauck D.; Klawonn F., Kruse R., Foundations of Neuro-Fuzzy Systems., 1th edition, Chichester, 1997.

[4] Jang J.S.R., Sun E.C.T., Mituzami E., Neuro-fuzzy and Soft Computing. 1st edition, Prentice Hall, Inc. Upper Saddle River, 1997

[5] Zadeh L.A., A New Direction in AI. Toward a Computational Theory of Perception. AI Magazine, pp 73-84, Spring 2001.

[6] Brooks R.A., Achieving Artificial Intelligence Through Building Robots. AI Memo 899, MIT, Art.Intelligence Lab., 1986.

[7] Brooks R.A., A Robot that Walks: Emergent Behaviours from a Carefully Evolved Network. Proc. IEEE Conf.on Robotics and Automation, pp 692-696, 1989.

[8] Vitko A., Markusek J., Jurišica L., Unified Simulation Environment for Learning Navigation of a Robot Operating in Unknown Terrain. Proc.of CLAWAR, Karlsruhe, Germany, pp 435-441, 2001.

[9] Mataric M. J., Behaviour-Based Robotics as a Tool for Synthesis of Artificial Behaviour and Analysis of Natural Behavior. Trends in Cognitice Sciences, Vol.1, No.3, pp 82-87, 1998.

[10] Saffiotti A., The Use of Fuzzy Logic in Autonomous Robot Navigation: a catalogue raisonné Technica Report TR/IRIDIA/, Univ.Libre de Bruxelles, 97-6,1997.

[11] Maybeck, P.S., Applied Optimal Estimation-Kalman Filter Design and Implementation. Notes of a continuing education course offered by the Air Force Institute of Technology, Wright- Patterson AFB,Ohio, since Dec.1974.

[12] www.wins.uva.nl/~arnoud/education/OOAS/fwi/Cap9.pdf

[13] Wu, H. et al., Sensor Fusion Using Dempster –Shafer Theory. IEEE Instrumentation and Measurement Technology Conf., Anchorage, AK, USA, pp 21-23, May 2002.

[14] Klein.L.A., Sensor Technologies and Data Requirement for ITS. 2nd edition, Artech House, 2001.

[15] Mendel J.M.,T.,Fuzzy Logic System for Engineering: A Tutorial. Proc. of the IEEE, Vol.83, No.3, pp 345-377, 1995.

Utilizing Lego Mindstorms as a Teaching Platform for Industrial Automation

Carolyn Oates, Alois Zoitl
Automation and Control Institute (ACIN)
Vienna University of Technology
Vienna, Austria
oatesc@acm.org , zoitl@acin.tuwien.ac.at

Abstract—**Industrial control systems are taught best using real systems. Such systems can be expensive, dangerous, and may break easily. In the other side simulations often do not react like the real system. IEC 61499 automation standard supports the current control system trend toward networks of event-driven distributed devices. Support for event driven control applications is new in IEC 61499 as are the tools supporting it. Three tutorials are presented to teach developing IEC 61499 event driven applications along with control theory basics using open source tools with the Lego™ Mindstorms hardware. This inexpensive training system can be used for teaching industrial control methods for students, as well as industrial professionals.**

Keywords: automation; control systems; robotics

I. Introduction

Real control systems, such as an industrial robot arm, are expensive; can be dangerous [3]; and may break easily. In a simulator timings and physical modeling often do not react like the real system, teaching the students only the software. Additionally industrial automation systems are undergoing a major transition towards distributed control systems adding new development paradigms. The problem of industrial automation education has been summarized by [12] as follows:

"During the last few years the education in engineering and mainly the control engineering, has suffered multiple changes due to the fast technological development and the current demands of the field."1

In order to support industrial automation engineers, the IEC developed standards to define how distributed control systems should be developed. The result of this standardization activity is the IEC 61499 [3], which provides a framework for networks of event-driven distributed industrial control systems. IEC 61499 applications are built using networks of new kinds of functions blocks (FBs), supporting event as well as data connections. Support for both event driven and distributed control applications are newly supported and required for the first time industrial automation by IEC 61499. The new kinds of FBs which support distributed control applications need to be learned. Although the standard is available now for nearly five years, little tutorial information is available. As new open source based tools like the 4DIAC–Framework for Distributed Industrial Control–are becoming available the gateway hurdle for adopting the new technology is greatly reduced.

[1] [12] II p. 3432

However a key open point for learning IEC 61499 based distributed control systems is the missing availability of cheap easily available training systems. Lego™ Mindstorms (LMS) offers with its building kit a flexible way of building small automation problems. Furthermore with the new system NXT it provides about the same computing performance as typical control devices used in the domain of industrial automation. With this work we like to show how LMS can be used together with 4DIAC to teach IEC 61499.

LMS has a great history for teaching robotics and control programming also with block like programming languages. However none of the available tools provides languages suitable for industrial automation engineers.

Lego™ Mindstorms software (a subset of Labview) allows sequential commands. So when using LMS software to blink the LED located on the light sensor, there is typically one light sensor block for *on* and another for *off* for the same light sensor. The same sensor may be tested in different phases of an application using different blocks. Telling the motor to move occurs via multiple TurnMotor blocks. Labview has data connections, but no event connections [7].

Lejos, Java on LMS, is object oriented so there is only one instance of a physical sensor, but the method to reads a sensor can be used multiple times. Behavior programming described in the Lejos tutorial can still reference the same instance in multiple behaviors [9]. In comparison FB instances are restricted by the standard to the one physical existence.

This article is structured as follows. In Section 2 we give a short introduction to IEC 61499. The environment is described in Section 3, followed by a description on how we developed the tutorials. The developed tutorials are described in Section 5. Finally we conclude the article and describe our next planed steps.

II. Short Introduction to IEC 61499

The standard IEC 61499 defines several models–the application model, the system model, the device model, the resource model, and the Function Block (FB) model–that allow the control engineer developing distributed control applications in a graphical manner. This short introduction to IEC 61499 should serve as basis for the rest of this thesis. A full description of IEC 61499's architecture may be found directly in the standard

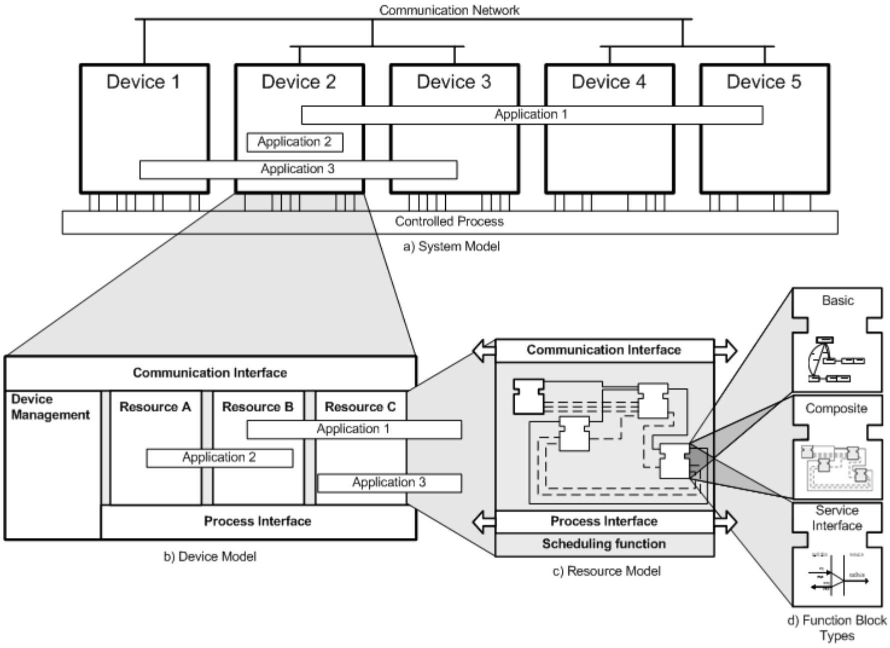


Figure 1. Overview on the main models of IEC 61499

IEC 61499-1 [6] or in a more comprehensible form in the books from Lewis [4] and Vyatkin [5].

The base model of IEC 61499 is the FB. A FB is a software component that is self contained and provides its functionality through a defined interface. This model has been adopted from the preceding standard IEC 61131-3 [11] and extended in its interface with an additional event interface. A trigger on one of the event inputs starts the execution of a FB. During the execution of the FB the input data will be processed, output data will be generated (depending on the functionality of the FB), and/or output events will be triggered. IEC 61499 defines three different FB types (schematically shown in Figure 1d):

Basic FBs (BFB) contain as main element a state machine that controls the internal execution on an input event arrival. This state machine is called Execution Control Chart (ECC) and is based on the *Sequential Function Charts* of IEC 61131-3. The ECC consists of three main parts: *ECC-states* with associated *ECC-actions* and *ECC-transitions* connecting the states. ECC-transitions are guarded by conditions. On an input event arrival the conditions of the current state's outgoing transitions are evaluated. The first true condition results in a state change. On state entry the associated actions of the state are executed. Actions consist of the execution of algorithms and/or triggering of output events. Algorithms may be programmed in any programming language. The main restriction is that algorithms can only access data inputs, data outputs, and internal variables.

Composite FBs (CFB) serve as container for FBs and may contain a whole set of FBs and their event connections and data connections. Incoming event connections and data connections are passed on to the internal FBs and vice versa for outgoing connections.

Service Interface FBs (SIFBs) provide a FB interface to functionality which is beyond the means of IEC 61499. Typical functionality encapsulated within SIFBS is the access to the control device's hardware, like the I/O interface or the communication interface. But also existing libraries that provide functions needed for the control system may be used through SIFBs. With SIFBs, this functionality can be encapsulated and the usage can be documented with so called service primitives. These service primitives allow to model event/data sequences explaining the usage of the SIFB. IEC 61499 distinguishes two general types of SIFBs. One is the requester SIFB, the other is the responder SIFB. The requester SIFB is an application triggered FB which remains passive until an event arrives at one of its event inputs. The responder type is a resource or hardware triggered FB. That means that it can send output events resulting on actions in the resource or the hardware (e.g.\ interrupts).

Through interconnecting the FBs with event connections and data connections to Function Block Networks (FBNs) the control functionality can be modelled in the application model. Applications are in general modelled without any device or control infrastructure in mind. The control equipment with their communication networks used for the data exchange between the distributed controllers is specified in the system model. A second part of the system model is the so called mapping. The mapping regulates which parts of the application are located on which control device. For example in Figure 1a Application 1 is mapped to the Devices 2, 3, 4, and 5; whereas Application 2 is mapped only to Device 2.

IEC 61499 models control equipment that is capable of executing IEC 61499 applications as devices. A device consists of a communication interface, a process interface, a device management, and may contain resources (see Figure 1b). The communication interface provides communication services for the device and the application parts residing in this device. The process interface provides the services for accessing the sensors and actuators needed to control the process (e.g. read the current motor position).

A resource is a functional unit that serves as containment for applications or application parts residing in the specific device and has independent control of its operation. Within a device resources can be created, deleted, configured, etc. without interfering with other resources and their contained applications. For applications a resource has to provide an execution environment (Figure 1c). That means it has to deliver event notifications to FBs and has to allow FBs to process the incoming events corresponding to their internal structure. A resource gets access to the communication interface and process interface from the device. SIFBs are the means to provide these services to the applications.

The management functionality within a device has the main task to administrate all applications and all resources located in this device. The management also provides an external interface for engineering tools allowing engineering tools downloading and uploading applications to (from) the device. This external interface is provided through the communication interface. Therefore the management needs an access to the communication interface (Figure 1b). At device level it provides the services to create, initialise, start, stop, kill, and delete the instances of resources and to query the attributes of resources. At resource level the same services allow the handling of FB instances and their interconnections.

III. ENVIRONMENT

The environment uses only open source applications. The 4DIAC-IDE is used to develop IEC 61499 standard compliant systems, applications and FB types. The standard provides portability and plug&play for controller applications. Applications are uploaded on to the Lego™ "controller" hardware [8] running the 4DIAC RunTime Environment (FORTE) under eCos operating system.

Figure 2. shows two IEC 61499 development tools, 4DIAC and FBDK. The tools generate XML files which comply with the IEC 61499 standard and can be exchanged.

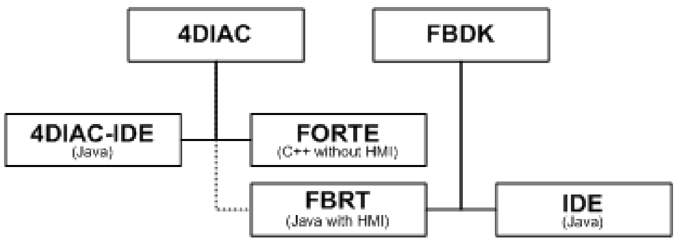


Figure 2. Development Environment

As shown in Figure 2. , the function blocks and applications are developed and mapped to device resources via 4DIAC-IDE. FBs are then exported to FORTE. The 4DIAC FB type export translates the FB's IEC 61499 XML representation into C++ code suitable for FORTE.

Figure 3. Upload to Lego™ Mindstorms NXT

FBDK FBs are reusable, so a simulation using FBDK HMI FBs to display the output is possible. This is useful for unit testing FBs inputs and outputs by event. The "device" is a Java window.

After the FBs are developed the LMS firmware must be flashed with the eCos+FORTE using SAM-BA [1] (see Figure 3). SAM-BA is provided by Atmel, the maker of the at91sam7s (ARM7) chip in the LMS [8]. At this point FORTE is running a simple Ethernet over USB program to upload and run an application.

eCos is an reconfigurable embedded operating system, so only the resources that exist in a device must be included. Control systems are typically embedded systems. Students who learn to work with LMS with eCos have a head start using eCos on other control devices.

IV. COURSE DEVELOPMENT

The tutorials assume no automation background. The tutorials build up concepts stepwise. Beginning FBs and IEC 61499 applications developed are reused and refined in following steps and tutorials. A simple example is presented and then the student must create or refine the presented example.

Research by Lego™ and MIT encourages the use of the freer explorative constructivist philosophy of education [2] by letting students explore rather than directed learning. However the problem solving cognitive philosophy is also popular for teaching control theory [12]. Teaching of basic concepts to model the problem need to be more guided. Once the student has framework to model the problem, they can be given more freedom and still communicate their work using IEC 61499 standard.

These tutorials are a mixture of cognitive and constructivism teaching philosophies. The first tutorial is guided problem solving learning, because specific control theory concepts using IEC 61499 standard are to be taught. After a basic example a related task, but slightly harder task is assigned. The second tutorial is meant to allow the student more freedom to use what they have learned. The only new concept is composite FBs. The third tutorial is a mixture and has the goal to teach the concepts of buffering and use of a bus.

Figure 4. shows a typical control loop. A line follower uses a controller to stay on the line. Calibration and software connection to hardware are also typical tasks in automation.

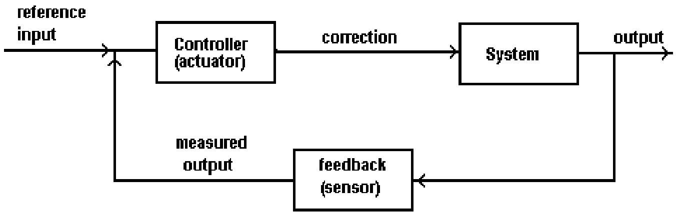


Figure 4. Feedback control loop

Tasks were chosen to teach control theory concepts as well as development of control application using the standard and tool.

In [5] three tutorials are presented for IEC 61499. The first tutorial modifies an LED application with 4 LEDs. The LEDs blink, or “chase” up or down. Turning an LED on and then blinking the LED was used as the first real test case for 3 different devices including LMS this semester. The NXT LabView Configuration VI also uses setting the light sensor’s LED as part of an example NXT software block [7].

In [5] the second tutorial used simple equation as the first full application. We tried a similar internal tutorial with 4DIAC/Forte for $x^2 + y^2$ with a network-like interface between FORTE (C++) and FBDK (Java). However there were many questions from students afterwards, especially about FORTE. There were fewer questions after developing an application to blink an LED. The blink application was first simulated with FBDK in a Java window and then applied to the actual hardware. The toggling the LED FB was reused in later applications.

We identified a set of key concepts of IEC 61499 and automation engineering for which it is important the trainee grasps in the first tutorials:

- How to represent feedback and feed forward control in IEC 61499
- IEC 61499 devices represent control hardware
- Sensors and actuators are represented by SIFB. Typically you have one instance of an SIFB per sensor or actuator.
- Error handling is performed through Boolean FB interface variables and appropriate events.
- Boolean input qualifier named QI is used for turning event processing on and off.

V. TUTORIAL APPLICATIONS

The three tutorials developed teach the use of IEC 61499 function blocks to build three working applications. First the environment (4DIAC, [10]), hardware (LMS, [8]), and standard with a simple application are taught. A LMS FB library is provided with FBs to directly interface with the LMS hardware. This includes sensors, motor, and shutdown, plus hardware status (battery power status). A LMSUtil library will be developed during the tutorials.

Sensor FBs are associated with physical sensors or motors. Students must be careful to send and receive events to the FBs associated with exactly one physical resource. The second tutorial application uses a different kind of sensor hardware. The third tutorial application teaches timing and using buffers to send information between applications.

A. Tutorial 1: Line Follower

The first tutorial is a line follower application with on- and off-the-line calibration. If multiple light sensors are available the application can be expanded to use 2 or 3 light sensors.

We want to test if it more understandable to start with Basic FB (BFB) or a Service Interface FB (SIFB). A greater than BFB will be explained first. Then a two point controller (hysteresis) basic FB is assigned. So they go from a “one point” controller” to a two point controller.

For teaching how to provide an interface via a FB to the hardware the student is asked to develop a SIFB for the Lego™ light sensor. This should also help the student understand what the purpose FORTE C++ Eclipse compilation is for. The sample FB will be the touch sensor, which reads data the same as the light sensor. The light sensor ports must be initialized, which shows the direct connection to the ARM7 processor. The test application is a light blinking application utilizing the light sensor’s LED. The Boolean data input QI is initialized to true. Errors indicated by the output variable QO=false are ignored for the moment.

The light blinking application also introduces the important and often needed Event FB library, which provides FBs for manipulating the event flow as well as timed events (i.e., cyclic triggers or delayed events).

Next, the boundary between on-the-line and off-the-line for the environment and a light sensor must be found. First the light sensor must be read and connected into the calibration calculation. There is no single way to do two-color calibration, but it’s important how the process knows and handles reading different colors. The top part of Figure 5 shows calibration where all samples of one color are read and all samples of the second color. All samples are averaged together.

Next the boundary between the two colors is used to turn on an LED if the light sensor is over “black” and off when it is over “white” as shown by DarkTst FB and Led4 FB in bottom part of Figure 5. Here it is emphasized that a port should only be used once.

Without error checking it is possible that a second FB for the same port/light sensor will be erroneously used. So error handling and Service Sequence diagrams explained must be explained. IEC 61499 uses + events to indicate no error and – events indicate error condition.

The application should now react when a port is allocated twice, since only one resource can be connected to a port. The INITO event combined with QO, event qualifier is split via an event switch into INIT- (QO=false) and INIT+ (QO=true). If the student previously used the same port/light sensor their design error will cause their application to no longer work.

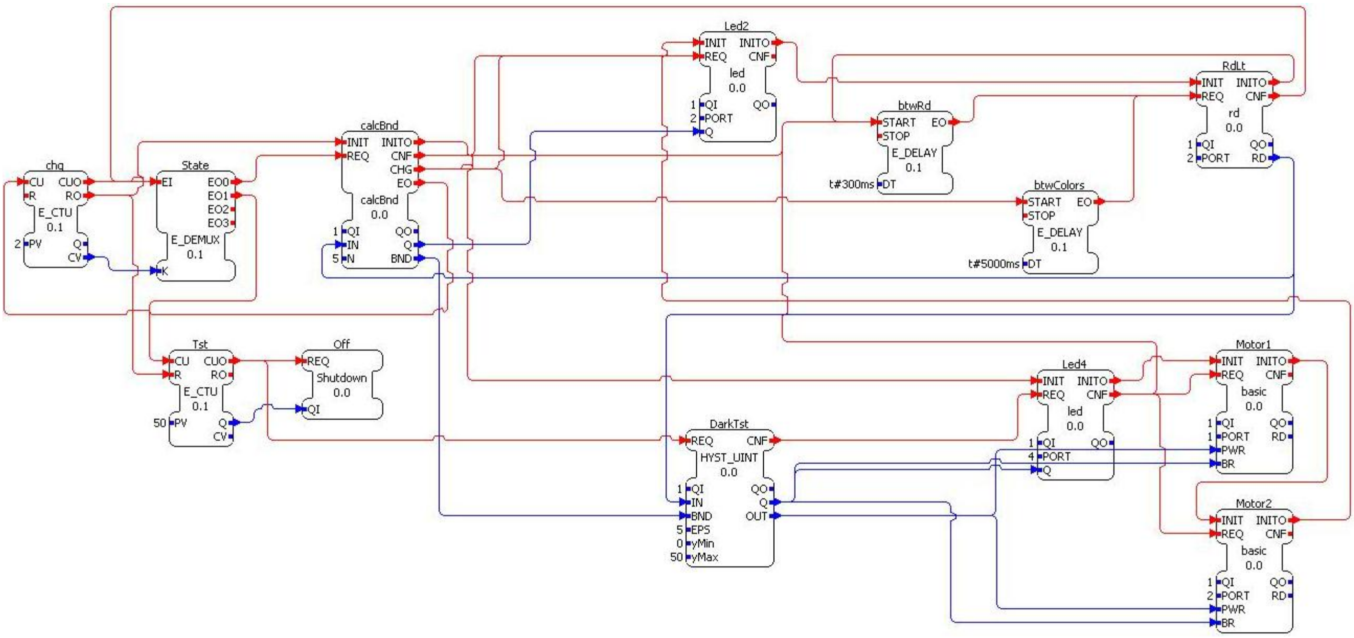


Figure 5. Simple Line Follower with light calibration

From personal experience making this failure helps the student remember each FB represents one real physical resource.

Finally the state of the light sensor can be used to tell the motors how to move can now replace toggling an LED. This final application should also be developed stepwise. First instead of just turning the LED on and off, the 2 motors can be set. The LED toggle is left for debugging. Toggling one motor off when not over the dark line allows the application to follow the line in one direction only. This simple line follower is shown in Figure 5. When a basic version is working, then the application can be expanded to a general line follower with feedback control and finding the line. Instead of just one light sensor to detect if the robot is over the line, multiple light sensors could be used. For example if three light sensors are used, the middle light sensor is over the line and other two light sensors straddle the line.

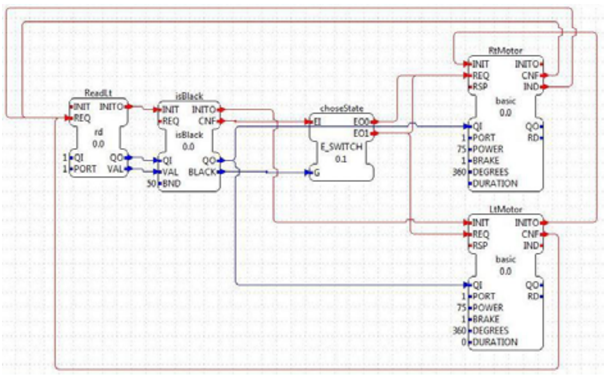


Figure 6. Simple Line Follower using composite FBs

B. *Further Tutorials*

The second tutorial uses the ultrasonic sensor as part of a simple Cartesian robot to keep a certain distance from the car. A Lego™ robot must be built to move forward and backwards based the "car's" length and up and down based on the feedback from the ultrasonic sensor. The student must develop their own control loop and Lego™ robot. Developing composite FBs is introduced to combine FBs together. Figure 6 shows how composite FBs would simplify the simple line follower by encapsulating the light calibration. The student must be careful to include error checking when needed in the composite FBs. Since only the input/ output events and variables are seen care must be given to not accidently reuse the same port.

The third tutorial uses stations to detect an object, its color, accept or reject it, and optionally deliver it. A pick-and-place robot is suggested. The application stations detect information and pass it on ahead of time via buffers, so the next station is prepared when the object arrives. This application teaches buffering data with time deadlines

The IEC 61499 tutorial examples can build on each other if multiple LMS NXT kits are available. The car is used in car wash and the Cartesian robot as one station in the assembly line.

VI. Conclusions and Future Work

Industrial automation is phased with major paradigm changes. First distributed control systems require a complete rethinking of how control applications are developed. By providing cheap and available tutorial systems control engineers can move up to the new paradigm much faster. With this work we showed how Lego™ Mindstorms NXT can be such a training platform for the new standard IEC 61499. We are developing tutorials which on the one hand utilize the LMS hardware and on the other side are representatives for typical industrial automation tasks. The final tutorial versions will appear on the 4DIAC website [10] under the development wiki.

Our next steps are to test the tutorials on different user groups in order to validate the contents and the structure of the tutorials. The first tutorial will be tested with students doing a practice work for the institute this summer. The last two tutorials will be tested with students in the fall. We also plan to use wireless communication between Lego™ Mindstorms NXT to teach using devices and applications across device boundaries.

REFERENCES

[1] ATMEL, AT91 ISP/SAM-BA® User Guide, http://www.atmel.com/dyn/resources/prod_documents/6421B.pdf.

[2] Mindell, D., Beland, C., Wesley, C., Clarke, D., Park, R., Trupiano, M. 2000. LEGO mindstorms, the structure of an engineering (r)evolution, 6.933J Structure of Engineering Revolutions, http://web.mit.edu/6.933/www/Fall2000/LegoMindstorms.pdf.

[3] S. Greengard, *Making automation Work*, Comm.ACM, Dec.2009, Vol.52,No.12, pp.18-19, http://doi.acm.org/10.1145/1610252.1610261.

[4] R. Lewis, "Modelling Control Systems Using IEC 61499 – Applying Function Blocks to Distributed Systems", The Institution of Electrical Engineers, London, 2001.

[5] V. Vyatkin, IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, O3neida Publ.

[6] IEC TC65/WG6, IEC 61499: Function Blocks, Parts 1 – 4, International Electrotechnical Commission IEC Std., Rev. 1.0, 2004/2005.

[7] Labview, Creating Lego Mindstorms NXT Software Blocks, ftp://ftp.ni.com/evaluation/mindstorms/NXT_Creating_MINDSTORMS_Blocks.pdf

[8] LEGO™ MINDSTORMS NXT Hardware Developer Kit, http://mindstorms.lego.com/en-us/support/files/default.aspx

[9] LeJos Tutorial, http://lejos.sourceforge.net/nxt/nxj/tutorial/index.htm

[10] 4DIAC – Framework for Distributed Industrial Automation and Control, www.fordiac.org

[11] IEC TC65/WG6, IEC 61131-3: Programmable controllers – Part 3: Programming languages. International Electrotechnical Commission, Geneva, 1993

[12] Paja, C., Scarpetta, J.. and Mejia, E. Platform for Virtual Problem-Based Learning in Control Engineering Education, p-3432-3437, IEEE EDUCON 2010, April 2010.

SyRoTek - A Robotic System for Education

Jan Faigl*, Jan Chudoba†, Karel Košnar*, Miroslav Kulich*, Martin Saska* and Libor Přeučil*
* Czech Technical University in Prague, FEE, Department of Cybernetics,
{xfaigl,kosnar,kulich,saska}@labe.felk.cvut.cz
† Czech Technical University in Prague, FEE, Center of Applied Cybernetics
{chudoba}@labe.felk.cvut.cz

***Abstract*—This paper presents insight to ideas and the current state of the project *SyRoTek - System for a robotic e-learning* that aims to create a platform for students' practical verification of gained knowledge in the fields of Robotics and Artificial Intelligence. A set of real mobile robots is being developed in order to provide remote access to real hardware for enrolled students. The advantage of the real system over a pure virtual simulated environment is in realistic confrontation with noise and uncertainty that is an indivisible part of the real world. In such a system, students can acquire in deep understanding of main studied principles in an attractive form, as students (especially future engineers) like to control real things. On the other side, this can be a potential issue if an accessibility to the system have to be guaranteed in 24/7 mode. In SyRoTek, robots are designed with special attention to long-term and heavy duty usage. Moreover, safety mechanisms are realized in several layers of the proposed software architecture that provide access to robot control and sensors. In addition, support for semi-autonomous evaluation of students' solution of their assignments is a part of the system.**

***Index Terms*—artificial intelligence, robotics, e-learning**

I. Introduction

Computers have been domesticated in the education process during last decades. Simulations of real processes can be easily realized and students can gain better (and faster) understanding of main studied principles. However, the real world tends to be more complicated than a pure virtual environment mainly due to noise and uncertainty. That is why it is important to engage real robots in the education. Even through it is not hard to control a simple robot, the final robot behaviour mostly depends on the real environment. It is known fact that early ideas of Artificial Intelligence clash with complexity and uncertainty of the real world. Therefore, it is very useful to confront algorithms with reality during students labs. Maintenance of real robots that are easily used by students can be very costly, thus so-called virtual laboratories have been investigated and developed by robotic groups. The advantage of these laboratories is that the Internet access allows to control a real robot even from students' homes or dormitories.

Several robotic systems with remote users' access have been realized since nineties, once the Internet becomes available. Early systems allow control of hardware devices in the tele-operating manner [1], [2], [3], [4], [5]. One of the first integrated robotic system for e-learning is the project ARL Netrolab [6], started at University of Reading in 1993 [7]. The used mobile robotic platform consists of robotic manipulator, sonars, infrared range finders and a set of cameras. Netrolab provides access to the robot control and sensors. The measured sensor data have been stored for further analysis. The follow-up project allows control a small rover in an environment simulating a surface of Mars [8]. Probably the most complex system have been developed in the project RobOnWeb [9] at Swiss Federal Institute of Technology in Lausanne (EPFL) [10]. Five fundamental services of web interface have been defined: chat, video, robot control, virtual robot representation, and logging. In the project REAL [11], four frames are used to provide a remote access to an autonomous mobile robot. The first frame provides the basic access to the laboratory and reservation system. The second frame realizes a tele-operated access to the robot. The additional frame enables possibility to use user's navigation module (written in C programming language) to control the robot. During the autonomous robot navigation, sensor data are collected by user's module and stored in the dedicated user space for further processing. The last frame represents module of a distance learning. A combination of a simulated environment with reality has been applied in the project LearnNet [12], [13]. The VRML technology has been used to model the real environment at the user side, while only coordinates of objects are transmitted over the Internet. This technique avoid necessity to transmit large video files of a real environment, thus it is suitable for low-bandwidth networks. A set of robots has been accessible for users in the project Virtuallab [14]. Several cameras monitored a play-field and a user can use a combination of several views to get better overview of the robots movements. The robots can be controlled remotely via the ActiveX technology or by a program in C++, Delphi or Java programming language. An open source solution based on the Player/Stage framework [15], [16] has been planned in another project of a virtual robotic laboratory [17], which unfortunately seems to be no longer active.

The aforementioned projects are only a small selected set of representative projects that deal with the remote access to real hardware devices. Lot of other projects can be found, however, the main concepts are pretty much similar and have been proposed in the aforementioned approaches. The main differences can be found in the used technologies that are improved over time and in a combination of several concepts in order to find the most suitable solution for particular requirements. Also new systems provide additional features that came from new technologies and progress in forms of e-learning education process.

The *SyRoTek - System for a robotic e-learning* is one of the current systems, which is similar to other projects. It shares many ideas of the previous systems, but it has been designed with different aspects that provide additional features over the previous systems. In this paper, we describe the main ideas and concepts of the system, which enable contingency to use the system in regular students labs related to Robotics and Artificial Intelligence as a field to verify learned theories and to gain practical experience with real robots.

The paper is organized as follows. The basic overview of the system and its architecture is described in Section II. Description of the designed robotic platforms and developed hardware parts is presented in Section III. Access to the system from user's point of view is described in Section IV. The essence of the SyRoTek e-learning part can be considered in a concept of assignments, which is described in Section V. Finally, remarks and the current progress status are presented in the conclusion.

II. System Overview

SyRoTek consists of an arena with real autonomous mobile platforms, communication infrastructure and the main control computer accessible from the Internet. The overview of the system is shown in Fig. 1. Robots are placed inside an arena

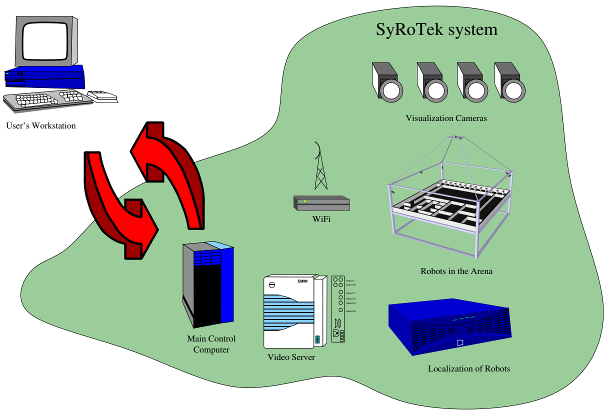


Fig. 1: SyRoTek system overview

with dimensions of 3.5×3.8 m including docking stations with a robot battery charging system. Several cameras support visualization of the real scene and creation of video records that are provided by a video server. Estimation of robots positions is crucial in various robotic navigation tasks, also it is useful for evaluation of user's assignments, thus a localization module based on processing of an image from the camera placed above the arena has been developed. The main control computer provide access for users from their workstations to SyRoTek through the Internet.

The architecture of SyRoTek consists of three main layers: the low-level hardware layer, core layer, and user interfaces, see Fig. 2. The hardware layer is a set of firmwares for micro-controllers and drivers for specialized devices (e.g. laser rangefinder, camera) that are used to collect data from sensors, to control the robot, and to watch the power system of the robot.

The core layer provides basic functionalities of the system and consists of several modules. The system module ensures safety and accessibility of robots from other parts of the system. The task module represents a set of supporting objects for tasks, e.g. realization of dynamic changes in the environment, tasks evaluation. The user module serves as the main access point to the system for regular users. It realizes an interface between SyRoTek-core and selected end user communication protocol through which a user controls a robot and reads sensors data.

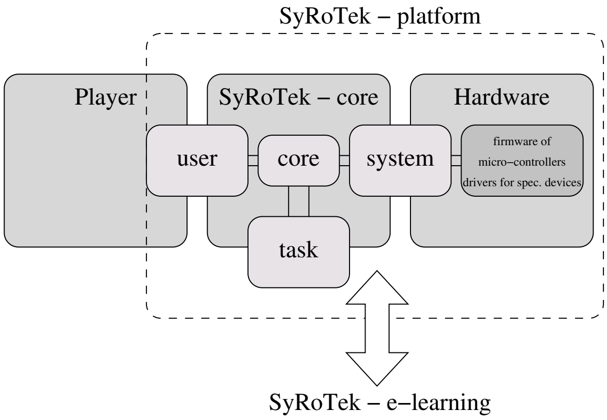


Fig. 2: SyRoTek architecture overview

The layers represent the so-called SyRoTek-platform that is hardware components and necessary software, which provides independent access to the components. The end user of SyRoTek will not be in direct connection with the SyRoTek-platform internal interfaces. Instead, another interfaces are provided. This abscission is realized due to the following reasons. At first, it allows selection of already known and used (by robotic community) abstractions and interfaces to hardware devices, in our particular case the Player [16] framework has been selected. Moreover the hardware part of SyRoTek is considered to be used in longer horizon that currently selected technologies for the current web based remote access to the e-learning part of the system. Thus, the separation of the SyRoTek-core from the presentation layer allows possible further replacement of the web pages by modern technologies, e.g. using visual impressive presentation based on new HTML5, CSS3 features, new toolkits like silverlight [18] or another Adobe Flash technology replacements.

The whole system is implemented as a set of services that provide access to particular functionalities of the system: robot and hardware parts, web pages, visualization and development tools. Besides, a set of maintenance tools and services are part of the system. The set comprises monitoring and notifications of status changes, power management, shutdown policies and emergency actions, like self-docking in the case of a low power. All these are designed to improved reliability of the whole system and possibly avoid system damage by an improper usage.

III. Hardware Description

The hardware components of SyRoTek consists mainly of a closed play-field called *arena* and a set of mobile robotic platforms. All obstacles are removable and a part of them can be controlled remotely. The robots have been designed for a long-term and heavy duty usage. The arena is placed in a university computer lab, see Fig. 3. Although the system is designed for a remote access, students can directly see the robots.

Fig. 3: SyRoTek arena

A schema of the robot is depicted in Fig. 4. The robot is called S1R and its body consists of the main chassis and an optional front module. The robot has differential drive realized by two Faulhaber 2224 motors with a gearbox (20/86:1) and the magnetic encoders IE2-512. The on-board power is provided by six Li-Pol Kokam 2400 mA-EHD-30C cells with nominal voltage 3.6 V connected in serial[1], thus the real voltage is in the range from 18.0 V to 24.6 V. The *power board* provides the main on-board voltage 5 V using the power regulator LM2596 - 5V/2A and the Atmel ATmega 2560 Micro-Controller Unit (MCU). A battery charger based on LTC4008 is integrated to the *power board*. The motors are controlled by the *control MCU* (cMCU) that is Hitachi H8S/2639 operating at 20 MHz placed on the *control board*, the maximal velocity of the robot is designed to be around 0.35 m/s. The on-board computer (OBC) is the Gumstix Overo Fire module with ARM Cortex-A8 OMAP3530 processor unit operating at 600 MHz and running the Linux kernel in version 2.6.x. The so-called *sensor bus* based on the I^2C bus is used to connect the *power board* and additional sensors to the OBC while cMCU is directly connected to OBC via dedicated asynchronous serial interface. A dedicated MCU called *bridge* is used for interfacing sensor bus to SPI of OBC. In order to guarantee data packet delivery time from the control computer to OBC a dedicated RF module is planned to be used, probably based on Nordic nRF24L01. Besides, WiFi can be used to transmit a large amount of data.

The chassis serves as carrier of basic sensors of the surrounding environments: five infrared range finders (Sharp GP2D120), three sonars (Devantech SRF10), floor sensors (twelve infrared sensors) and the intelligent camera module CmuCam3 [19]. The range sensors are directly connected to cMCU, while other sensors are connected to the sensor bus. Sensors of the robot internal states including the compass (Philips KMZ51) and the encoders are connected to cMCU. Besides, temperatures are measured in various places of the robot body, and currents to the motors are measured as well in order to provide the so-called software bumpers.

A dedicated MCU is used to wrap particular interface to be sensor bus compatible. Even though this unification requires additional MCU, it is advantageous from the software point of view. A unified communication mechanism can be used with various devices, and to transmit data from sensors to OBC and the main control computer, see schema of the communication between sensors and users in Fig. 5.

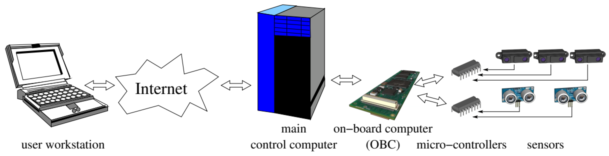
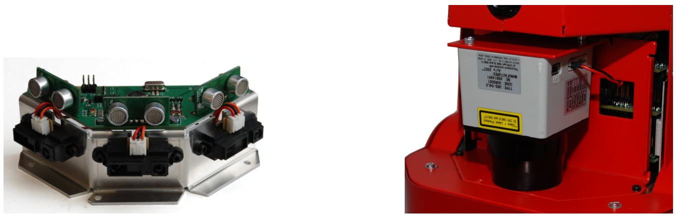


Fig. 5: A schema of a communication between sensors and users

Additional sensors, e.g. the front sensor module can be connected to the sensor bus, or directly to the OBC. Nowadays, two types of the front sensor module are available, see Fig 6. The first one is equipped with three sonars (Devantech SRF10) and three infrared range sensors (Sharp GP2Y0A21Y), the module is connected to the sensor bus. The second one uses the laser range finder Hokuyo URG-04LX and it is connected to OBC via the USB interface.



(a) (b)

Fig. 6: Two types of front sensor module

Three robots S1R during the exploration task are shown in Fig. 7. Notice the patterns on top of the robots that are used by the localization system to estimate the current positions of the robots.

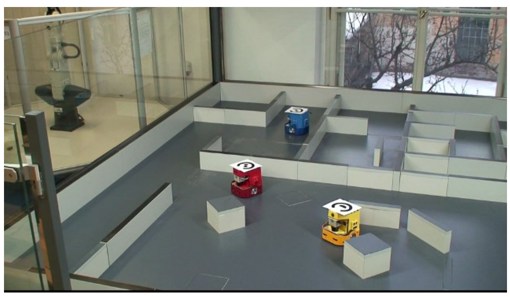
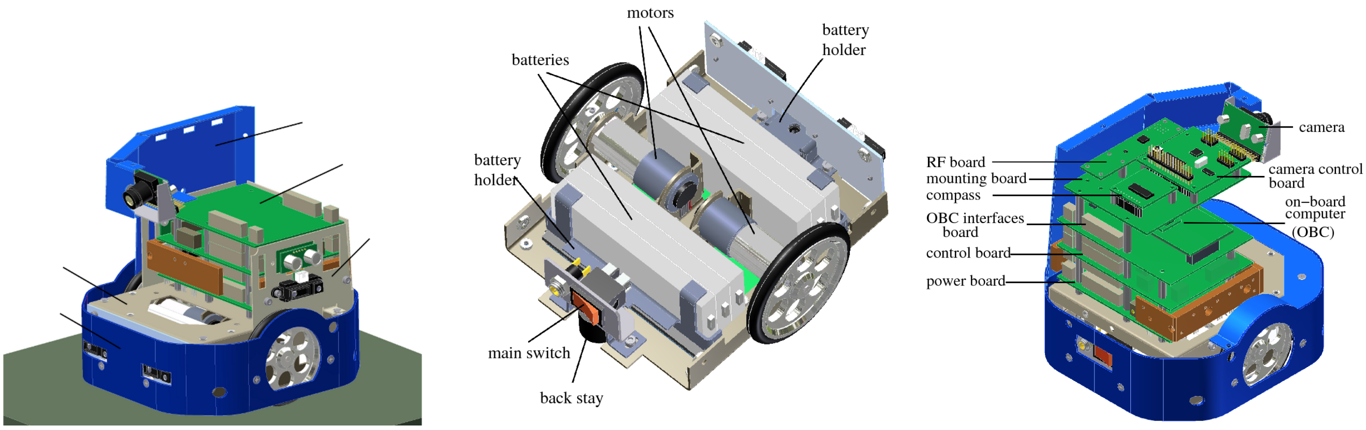


Fig. 7: Three S1R robots during exploration

[1]Based on real experiments, the battery pack provides energy for around eight hours of a continuous robot moving without additional power saving techniques.



(a) construction parts (b) components in the chassis (c) electronic components

Fig. 4: Schema of the SyRoTek robotic platform - S1R

IV. User Access

Three types of user access can be found in SyRoTek: web, remote shell, and data (video streams and sensors data). The web access can be considered as a primary gate to the system. It provides basic description of the whole system, account creation request, reservation system, maintenance of a user profile, courses and particular assignments. A more detail description of this part of SyRoTek is dedicated to Section V. In this section, the next types of accesses are described.

SyRoTek is focused on an e-learning in robotics, particularly it aims to provide support of knowledge transfer of foundations of several robotic problems and also practical verification in various robotic task. It means that a student can use real robots to verify the learned principles in a real practical application, so the student is requested to create a program that is able to navigate a real robot in an environment.

The practical orientation of the robotics steers SyRoTek to provide support of software development process oriented to robotics. The best practice in robotic development is an initial creation of an algorithm or a control program that is verified against simulation, which is typically much faster process than with a real hardware. Moreover, a program that is able to navigate a mobile robot is often consisted from various components, and the complete program can be quite complex. Thus, it is advantageous if a student can use already available components. Also a good hardware layer abstraction is a plus in order to create a simple program that can be easily transfered from a simulation to real robots. These considerations are the main reasons why the Player/Stage framework [16] has been selected as the main SyRoTek user interface. The Player has a hardware abstraction based on a set of interfaces and devices that are proven by more than ten years of history by several robotic researchers around the world. The Player can be accompanied by simulators Stage or Gazebo. The Player follows a client/server concept in which the user application is a client that is connected to the server (player) via TCP connection. The server provides interfaces representing particular devices, which can be real devices or simulated ones. So, the system can be used in various configurations, e.g. a server running at user's workstation or at a robot, which is remotely accessible.

A. Robot Access Module (robacem)

Even though the Player is flexible enough to be used in a robotic application, it does not provide required functionalities of SyRoTek. The main issue arises when an authorization to particular sensors have to be granted, e.g. if an evaluation or monitoring of user's application performance have to be realized. The authorization is not a part of the Player at all. When user's application is connected to the Player server, only one program is able to actively control the robot (its motors) by a dedicated serial interface, e.g. RS232. In such a case, the robot will be inaccessible for system services, which is not desirable. In addition, a user can accidentally send a command that can navigate a robot into forbidden areas. Such a situation cannot be handled in low levels firmwares, because robot surrounding environment have to be taken into account, so a high level action monitor is required. From the other point of view, an evaluation can be based on different sensors, e.g. a robot position from the global localization systems, that can be abused by a user to quickly solve the given assignments. Therefore, to authorize access and to guarantee accessibility to the robot for authorities (like monitoring and maintenance services) an additional component called ROBot AcCEess Module *(*robacem) is used in SyRoTek-platform.

Robacem represents a robot at a particular computer. The S1R robot uses OBC that is connected with the main control computer via WiFi or dedicated low-bandwidth radio channel with guaranteed transport delays. Therefore two robacem modules are running for each robot in SyRoTek: at OBC and at the main control computer. Robacem allows simultaneous and independent access of system monitoring services and Player servers, which are accessible from user applications. A basic schema with possible places where users' applications can be executed is shown in Fig 8. The connection between user's application running at the main computer and the player server at OBC (represented by the red arrow) is possible. However, it can be used only with special attention. During preliminary experiments, a client application connected to the player is able to generate very intensive traffic, which significantly reduce the response of WiFi connections to other robots. Therefore, such a connection can be used only if additional bandwidth limits are involved, e.g. restriction of a connection bandwidth. Otherwise a user can cause degradation of system functionality.

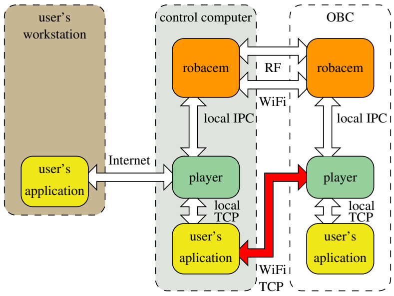


Fig. 8: Connections of process with robacem modules

B. User's Remote Access

Student's program to control mobile robots requires necessary software development tools that have to be installed at a users's workstation, which can be tedious. Therefore a remote access to the main control computer, which is fully configured, is allowed. A user can use secure shell (ssh) or secured graphical access by ssh tunneling of XDMCP. These protocols are easy to use within standard installation of Linux based distributions or other unix based systems and they do not require additional proprietary software. Moreover, a remote process execution can be configured in such a way that a user does not recognize a difference between local and remote execution at a glance.

The remote shell access is advantageous in a situation when user's program requires low transport delays, which cannot be guaranteed in a case of a low bandwidth Internet connection. The shell does not have high requirements, and the user is able to execute or even develop her program remotely with slow connections.

C. Data Access and Visualization

The best way how to access to the robot is a connection of user's application to the player server running at the main control computer. Our pilot experiments indicate that a connection with 512 kbit/s bandwidth provides sufficient comfort, if video streams are not required.

Video transmission requires an additional bandwidth that is why it is considered as an independent communication channel. In a robotic application, data from real sensors are processed in order to generate the most suitable action. It is very useful if data are visualized and combined with a real view of the scene. We consider the Stage simulator (in version 3.x) as a base of our visualization systems. The simulator provides models of sensors with particular visualization, therefore we enhance it by consideration of several views that can be combined with videos of the real scene. An example of such a visualization is shown in Fig. 9.

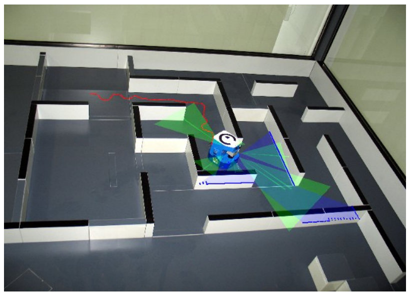


Fig. 9: Visualization of the arena and real sensor data

A user can use our modified stage simulator as a visualization of the real situation in the arena. According to her Internet connection, she can select particular video streams from several cameras mounted in the SyRoTek arena and various quality (resolution and bandwidth) of videos. Moreover, videos can be recorded during user's application execution and together with recorded data they can be used for debugging or as a proof of program functionality in the assignment evaluation process.

V. Assignments

SyRoTek as an e-learning system is considered to be practically (task) oriented, due to its relation to real robots. The studied principles of the related domains can be demonstrated in reality by moving a robot in the arena. From this perspective, the essence of SyRoTek lies in robotic tasks. Besides, the supplementary materials can be presented to students in standard ways, e.g. in a form of web pages.

In our first ideas and concepts (based on the previous and current virtual laboratories) we have planned to use one of the already available web based e-learning systems, particularly Moodle [20] has been considered as the most suitable candidate. Later, we recognized that a practical part of assignments (robotic tasks) is tightly related to the software development process of an application to control real mobile robots, which is not a part of general systems for Content Management System (CMS), or Learning Management System (LMS). Such systems can be customized, but most of the specific functionalities of SyRoTek have to be implemented from scratch, which can be more costly (due to general system API) than a creation of a simple specific (single-use) system. Based on this premise, we have reconsidered necessity of a general CMS and instead of primary usage of such a system we use direct description of tasks according to the SCORM 2004 definition [21]. Specific information related to the robotics, resp. SyRoTek, are stored in the Learning Object Metadata (LOM), therefore it can be eventually used in any system that supports SCORM 2004. A relation database has been selected to store the tasks definitions. Its main advantage is relatively cheap creation of copies of assignments and fast access to the definitions that are crucial properties of the desired feature of SyRoTek that is an individualization of assignments.

E-learning systems are sometimes denoted as impersonal. In SyRoTek, we use current technologies to create a support for more personal relation between a teacher and his course students. An individualization of a particular task for each student enables capability to reflect current knowledge of the student and his focus to the most relevant parts of the problem. Such an individualization needs a set of supporting modules that substitutes particular sub-tasks of the assignment and are helpful to quick and targeted knowledge transfer to the student. Initial versions of these modules are part of the system, but further student's implementation of particular assignments can be used in future.

A. Courses and Tasks Concepts

Courses can be divided into three categories in SyRoTek: introductory, intermediate, and advanced. The first category are courses to afford fundamental algorithms in key robotic domains like simple robot control, reactive behaviours, dead-reckoning, sensor processing and path&motion planning. In these courses, students are also introduced to the provided SyRoTek functionalities. The intermediate courses are based on Top Assignments (TA) that comprise from several fundamental problems. These courses are organized to guide students to acquire knowledge of necessary fundamental algorithms in order to solve TA of the course. The advanced courses are similar to the intermediate courses. The difference is that the advanced courses aim to solve the selected TA itself.

Two groups of TAs can be defined: basic and advanced. The basic TAs are typical problems in robotics and artificial intelligence, which are well studied or well described, e.g. simultaneous localization and mapping, inspection, exploration, coverage, pick&delivery. The advanced TAs are hard problems, for which it is expected that students will either study literature to find some approximate solution or they will creatively develop its own approach. These problems are typically designed as multi-robot tasks where cooperation and coordination of robots play an important role, e.g. games like pursuit-evasion, capture the flag or treasure hunt.

B. Task Evaluation

From the e-learning point of view, teacher's access to SyRoTek is also important. The system allows specification of constraints under which a task can be solved by the particular students. The system supports verification of the task in semi-autonomous manner. A teacher can write a module that is simultaneously executed with student's program within a dedicated period for submission. Such a module monitors behaviour of student's program to control the robot or it can dynamically change environment according to the robot behaviour, e.g. an evader controlled by student's program can be pursued by a different program in pursuit-evasion scenarios. An output of the student program can be automatically processed to verify student's results. A performance of the robot behaviour is captured and video is created for the teacher to support evaluation of student's solution.

VI. Conclusion

The SyRoTek project is in the second half period of solution, therefore this paper presents only the main ideas, concepts and preliminary results. First robots have been created and concepts of the user access to robot functionalities have been verified in selected robotic tasks. These experiments support the main ideas of the proposed concepts, however it also show possible communication issue related to limited bandwidth of the used WiFi infrastructure. The issue can be solved by additional restrictions of the direct users access to a robot in order to guarantee desired quality of accessibility for other users. Thus, it is not a drawback, as it will improve the overall reliability of the system.

The further development will concern to finalization of robots hardware, creation of an initial public access to system, and preparation of supplementary materials. It is expected that SyRoTek will be open in trial application for users at the end of the year 2010.

Acknowledgment

The work presented in this paper has been supported by the Ministry of Education of the Czech Republic under program ”National research program II” by the project 2C06005.

References

[1] “http://www.telegarden.org,” 1997, (accessed 27 July, 2010).
[2] “http://www.telescope.org,” (accessed 27 July, 2010).
[3] “http://www.iai.uni-bonn.de/~rhino/tourguide,” 1997, (accessed 27 July, 2010).
[4] “http://www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/Xavier,” 2001, (accessed 27 July, 2010).
[5] “http://robotoy.elec.uow.edu.au,” (accessed 17 July, 2006).
[6] “http://www.arl.rdg.ac.uk/research/netrolab,” 2003, (accessed 27 July, 2010).
[7] G. McKee and R. Barson, “Netrolab: a networked laboratory for robotics education,” *IEE Colloquium on Robotics and Education*, April 1995.
[8] “http://www.redrover.reading.ac.uk/RedRover/index.html,” (accessed 26 September 2006).
[9] “http://asl.epfl.ch/research/projects/RobOnWeb/robOnWeb.php,” 2002, (accessed 27 July 2010).
[10] R. Siegwart and P. Sauc, “Interacting mobile robots on the web,” in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, May 1999.
[11] E. Guimarães, A. Maffeis, J. Pereira, and et. al, “Real: A virtual laboratory for mobile robot experiments,” *IEEE Transaction on Education*, vol. 46, no. 1, February 2003.
[12] “http://www.learnet.de,” (accessed 27 July 2010).
[13] I. Mas'r, A. Bischoff, and M. Gerke, “Remote experimentation in distance education for control engineers,” in *Proceedings of Virtual University 2004, Bratislava, Slovakia*, December 2004.
[14] “http://www.robotika.sk/projects/virtuallab,” 2008, (accessed 27 July 2010).
[15] “http://playerstage.sf.net,” (accessed 27 July 2010).
[16] B. P. Gerkey, R. T. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor systems,” in *In Proceedings of the 11th International Conference on Advanced Robotics*, 2003, pp. 317–323.
[17] “http://vlab.pjwstk.edu.pl,” (accessed 26 September 2006).
[18] “http://www.silverlight.net,” (accessed 27 July 2010).
[19] “http://www.cmucam.org,” (accessed 27 July 2010).
[20] “http://www.moodle.org,” (accessed 27 July 2010).
[21] “http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Overview.aspx,” (accessed 27 July 2010).

Mobile Robotics at FEE CTU

Jan Faigl*, Tomáš Krajník*, Karel Košnar*, Hana Szücsová*, Jan Chudoba†, Vladimír Grimmer*, Libor Přeučil*
Czech Technical University in Prague, Faculty of Electrical Engineering
*Department of Cybernetics, †Center of Applied Cybernetics
{xfaigl,tkrajnik,kosnar,chudoba,preucil}@labe.felk.cvut.cz, {grimmvla,szucshan}@fel.cvut.cz

***Abstract*—In this paper, we describe concepts and main ideas of the labs of the Mobile Robotics course at Faculty of Electrical Engineering, Czech Technical University in Prague. Besides, we present our gained experience from three years of teaching of the course. We consider the students' contact with real hardware and real sensor data as the most important part of mobile robotics as the mobile robot can quickly lose information about its position in contrast to stationary robotic manipulators. Thus, the autonomous navigation is a crucial problem. Moreover, a computer simulation cannot substitute complexity of reality, such as noise, imperfect measurements and random events. To achieve our desired pedagogical goals we have decided to develop a new small platform that will be based mostly on off-the-shelf components and it will have sufficient computation power to use the Player robotic framework. The labs are organized into four consecutive assignments with increasing complexity and a final assignment that combines particular students' results from the previous tasks. The final assignment is to create an algorithm that navigates the mobile robot in order to create a topological map of the environment and reuse this map for later navigation.**

***Index Terms*—robotics, e-learning**

I. Introduction

The course "Mobile Robotics" is an optional subject of the Technical Cybernetics study program at Faculty of Electrical Engineering (FEE), Czech Technical University in Prague (CTU). The course assumes a small student group because of the necessity of individual and personal contact between students and teachers. When the course has been opened at summer semester in the year 2008, it had less than twenty enrolled students. In this paper, we describe the main concept of the course labs, selected solutions and gained experience from three years of the course.

The course is taught within fourteen weeks of the semester and is organized into the same number of lectures and labs. The lectures are dedicated to theoretical description of basic principles of navigation of autonomous mobile robots. These lectures cover relatively wide range of topics from motion control, sensor data processing and path planning, to environment modeling, localization, mapping and simultaneous localization and mapping (SLAM) techniques [1]. It is clear that only the most important ideas and concepts of mobile robotics can be presented within the limited time of the course. Therefore, the core of the transferred knowledge is in understanding of fundamental principles and issues of the mobile robot navigation in a real environment.

The principles discussed at the lectures are practiced during the labs assignments with real robots. As a part of our lab concept, we had to select the most suitable robotic platform for our desired pedagogical goals. The desired platform had to be affordable, allow a reconfiguration of sensors, provide enough processing power and be robust enough to be used by inexperienced students.

Based on our previous experience in the field of experimental research of navigational algorithms for autonomous mobile robots, we have chosen to create a small platform, even that similar platforms (in the sense of robot dimensions) have been available in the market. To allow reproduction and reuse of our platform by others, we have decided to use off-the-shelf components, integrated the platform in the Player [2] system and published documentation, construction plans and software on our web pages [3].

The main reason for our choice was based on the fact that marketed solutions are too simple and insubstantial, e.g. LEGO Mindstorm or Fischertechnik ROBO Mobile Set, do not provide enough processing power or cannot be extended by advanced sensors, e.g. Rogue Blue ERS, Arrick Arobot Mobile Robot, Carper Rover OOPic-R Combo, Rogue ATR Base, Kit, Lynxmotion 4WD1, Inex Interactive C Robot Kit V2.0, AIRAT 2, Surveyor SRV-1, Hemisson, Khepera, or are too expensive, e.g. Pioneer 3-DX, Koala.

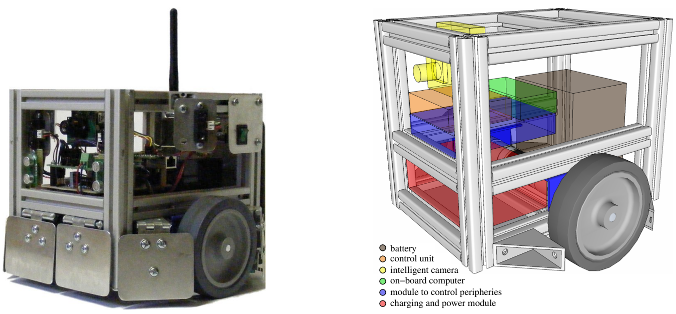
The main idea of the labs is to properly setup the robot sensor system and to develop an application that is able to control a mobile robot in order to create a map of a real environment. At first, students are introduced to the field of mobile robotics in four consecutive assignments with increasing complexity. In these assignments, students adopt methodology to create an application controlling a real mobile robot. After that, the students are given a general description of the final task, which is exploration of unknown environment. In this task, the robot should plan its actions in order to create a complete map of the surrounding environment. Such a map should provide enough information for effective path planning to destinations given by a human operator. Specification of the final task is general, in fact, it is impossible to fulfill such a general task within the course labs. Therefore, the students are requested to specify restrictions and conditions in which their robot will be able to fulfill the exploration task.

The rest of this paper is organized as follows. The developed mobile platform called *MORBot* is described in Section II. The developing environment and concepts of students' work on their applications is presented in Section III. The assignments, their main purposes and challenges for students are described in Section IV. Remarks and ideas for further course improvements based on gained experience from the three runs of the course are presented in the conclusion.

II. MORBot Platform Description

The aforementioned requirements lead to design a new robot, which was inspired by the research platform G^2Bot [4] and robots for the Eurobot competition developed in the Gerstner Laboratory [5], [6]. The robot we needed for the course had to be smaller than these platforms, because of the space restrictions (the course takes place in an ordinary computer lab). Moreover, it had to be simpler and easier to use. Due to the target application of the platform, a motion on a flat surface had been considered as sufficient.

The design has been constrained by our preliminary budget. The required cost of particular components for a single robot has been targeted to be less than one thousand euros. The main idea, concepts and preliminary components have been suggested by authorities, however the final robot design and construction has been realized by students in two Bachelor's [7], [8] and one Master's thesis [9].



(a) chassis (b) components

Fig. 1: The MORBot platform and its main components

The robot hardware consists of an aluminum chassis, power, motion, sensor and control subsystems, on-board computer and an intelligent camera, see Fig. 1. The hardware subsystems are interconnected by several buses, see Fig. 2. Its software composes of micro-controller units (MCU) firmwares and the Player server from the Player/Stage framework [2] providing hardware abstraction for users.

A. Robot hardware

The skeleton of the robot is composed of interlocked X-shaped aluminum beams (Item profiles). These are firm enough to support robot devices and provide reliable shock protection. The skeleton dimensions are 16.0×22.0×18.5 cm and the robot circumference is about 85 cm. The total weight of the robot (including battery and sensors) is about 5 kg.

The power subsystem is composed of a 12 V, 5 Ah sealed lead-acid battery, a charging control board and a voltage stabilizer, which provides 5 V for additional electronic boards.

The motion subsystem is based on the differential drive Devantech RD01 and two supporting rollers mounted at the

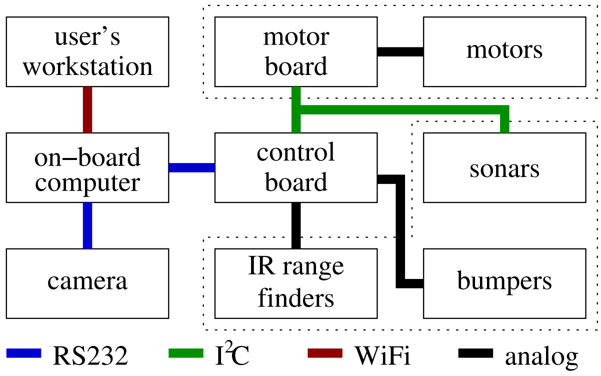


Fig. 2: Robot subsystems scheme

back side of the robot, the maximal forward velocity is about 0.8 m/s. The motor driver MD23 controls speeds of both wheels and counts pulses from motor IRC sensors. The counter values are sent to the control board via the I^2C bus.

The sensor subsystem consists from these sensors: four Sharp GP2D120 (IR) rangefinders, two Devantech SRF10 sonars and seven mechanical bumpers. The IR rangefinders have detection range between 0.04 and 0.30 m and the sonars detection range is 0.1-4.0 m. The IR rangefinders provide an analog voltage signal needing to be further processed and the sonars SRF10 supply the measured distance in centimeters via the I^2C bus. Positions of the sonars and the IR sensors are not fixed and students can reposition them according to their intentions and the particular assignment. The mechanical bumpers are based on microswitches covered by metal plates, see Fig. 3.

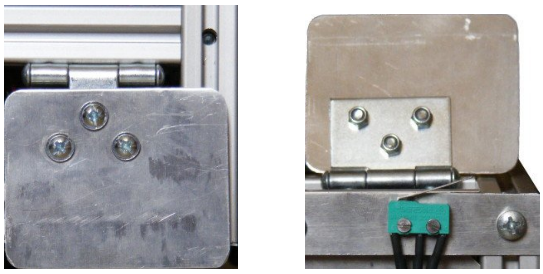


Fig. 3: The mechanical bumpers

The control subsystem works as an interface of the motion and sensor subsystems to the on-board computer. The control board uses the Atmel ATmega 168 MCU to continuously gather data from the sensors and the motor control board. The MCU estimates the robot position from values of the motor board IRC counters. The current status of the sensors and the estimated position are provided to the on-board computer via the RS232 interface on request. Moreover, the on-board computer issues commands, which set speeds of the motors. Control subsystem ensures safety of the robot independently on the on-board computer. When a frontal bumper is pressed, the control unit prohibits forward movement and vice versa, thus preventing damage to the environment and to the robot.

The primary purpose of the on-board computer is to run the Player server, which interfaces the robot devices and sensors. Moreover, the computer allows using more advanced sensors like cameras or laser rangefinders. The computer is based on the Gumstix Verdex XL6P [10] motherboard with two expansion boards: the netwifimicroSD and interface board called PortBoard [9]. The motherboard utilizes the Marvell PXA270 processor running at 600 MHz, 128 MB RAM and 32 MB of internal FLASH RAM with installed Linux operating system in version 2.6.x. The expansion boards provide a micro SD card slot, Ethernet and WiFi interfaces, three serial ports and I^2C and USB buses.

An intelligent camera, the CMUcam3 [11], is installed on top of the robot and is connected to the on-board computer via the RS232 interface. The camera itself is capable of recognizing and tracking objects with distinctive colors [12]. The on-board computer specifies a color of the searched object and the camera starts to send object position in its image coordinates.

The fact that the robot is not completely covered is appreciated by students, because they have a good overview of the robot inner structure. Moreover, they can change placement and configuration of particular sensors and realize that sensor configuration must be reflected in software controlling the robot.

B. *Robot software*

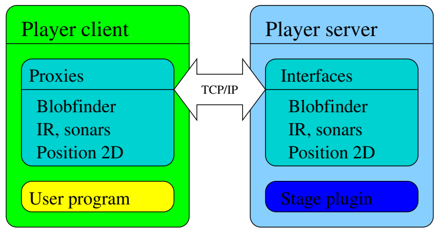
The software of the robot is divided into three layers. The control software (firmware), which runs on the control board MCU, gathers sensory data, estimates the robot position and provides an interface to the robot motors. The image analysis software running on the CMUcam3 can also be considered as a part of the firmware. The second layer provides abstraction of hardware devices and it is realized by the Player server running on the on-board computer. The last layer is students' application itself, which is a client application to the Player server.

One of the advantage of the Player system is the support of various devices like the CMUcam3, IR rangefinders, sonars, motors and odometric systems. However, due to our own design of the control board, we had to develop our own driver for it [8]. The Player system offers an easy-to-understand tutorial on driver development, thus implementation of a new driver did not take a long time. A student, who wants to use the robot, can access its on-board computer remotely via a WiFi connection. The student program connects to the Player server running on the robot can retrieve sensor values and set robot angular and forward velocities.

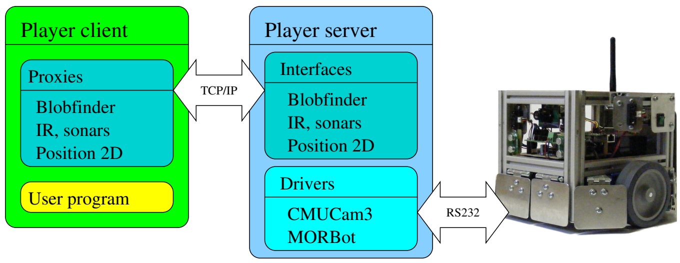
Documentation needed to build the robot including component list, electronics board schemes and software is accessible through the web pages [3].

III. Developing Environment of Students' Assignments

The practical part of the labs is based on the Player/Stage framework [2], which uses the client/server architecture. The Player/Stage framework is widely used, well documented, free and open-source [13]. The basic principle of the Player/Stage framework is shown in Fig. 4. The Player server provides a



(a) a connection to the Stage simulator



(b) a connection to the MORbot

Fig. 4: A basic scheme of the Player/Stage framework

unified networked interfaces to robotic sensors and actuators. Programs, which control the robot, connect to the Player server as network clients. Therefore, the robot control programs can be written in any programming language and can run on any computer with a TCP/IP connection to the robot. The Stage is a simulator plugin to the Player server, thus a developer without a real robot can use it to substitute the real hardware and environment by simulation. It is another great advantage of the Player system, because it provides straightforward deployment of the program verified in the simulator to the real robot.

These properties allow students to work at home, thus prior to the school labs they can verify their programs. Consequently students can spend more of their time at the lab by consultations of found issues with a teacher and practical verification with a real mobile robot. The students can either download and install the Player/Stage on their computers or use the standard unix-based graphical remote access protocol to run the Player/Stage on a university server.

One of the encountered issues is the fact, that not all students are familiar with alternative development tools and concept of the client/server architecture. They tend to prefer the particular development environment adopted by their previous experiences (typically not well mentored). To allow fast introduction to the development tools, we have prepared a skeleton of user's client application. The skeleton composes of several files written in C++, which define classes representing the robot and its devices. Some methods of these classes are empty and are supposed to be implemented by students in the lab assignments. For example, the *robot* class defines (empty) methods, which correspond to the robot behaviours, e.g.

`moveTo(x,y)` or `explore()`. The files with C++ codes are complemented by recipes ("Makefiles"), which prescribe how to build sources and create the executable binary file. Thus, only a minimal set of tools (`gcc` compiler and `gmake` build system) is required. After several practical applications, students reported that this framework is comfortable, straightforward and easy to use, especially within a remote session.

This positive feedback is very important for us, due to the fact that students are introduced to the programming language Java in the first years of their study at FEE, CTU. The Player framework is used with the C/C++ interfaces and for some students it is difficult to learn a new programming language. Even that students do not become experts in C/C++, they recognize that for a certain task a new (not already known) tool can be more appropriate. This helps to students to realize that the essence of programming does not lie in mastering one particular language, but rather in knowledge of algorithms and systematic principles.

Another possible issue of the used Player/Stage framework is a requirement of the unix-like operating system to install and use the framework. Even that the recent Player version 3.x supports Windows, unix-based operating systems are advantageous for students, because the development tools are part of standard installation, e.g. the `gcc` compiler or the `gmake` build system. However, if a student would refuse to install a unix-like system on her computer, she can install the freely accessible Xming program [14], which provides access to the fully configured university server with all the necessary tools.

Students are organized in teams with three or four members, and the students are encouraged to actively use a Version Control System (VCS) for the program sources, in particular the Subversion system [15]. The VCS allows not only comfortable and convenient organization of source codes but also provides information on how students work during the semester. It is not surprising that most of the students increase intensity of work at the end of the semester. We use reports of students' activity to emphasize the importance of continuous work, see Fig. 5 for an example. The reports also provide us a valuable feedback on how students deal with the assignments. Moreover, teachers can access and review students' source codes and suggest corrections by adding notes to the source. This absolves us from tedious management of e-mail attachments.

The VCS and the provided framework allow easy download and compilation of students projects. Just two commands are needed to bring the up-to-date program version and create an executable file, i.e. `svn update` and `gmake`. This significantly reduces teachers' load when examining students' programs and allows to focus on really important issues.

IV. Labs Assignments

The main objective of the labs is to make the students understand the nature of uncertainty in mobile robotics systems. The objective is materialized in assignments to create programs that will provide an intelligent behaviour to the MORBOT platform. The students should learn that uncertainty in measurements and action results can be dealt with by means of feedback loops realized by controllers, which compute angular and forward velocities of the robot from sensor values and gather information about the robot surrounding environment. In addition, the students should learn that a proper decomposition of a mobile robotic task should be carefully chosen to avoid the pitfalls of uncertainty.

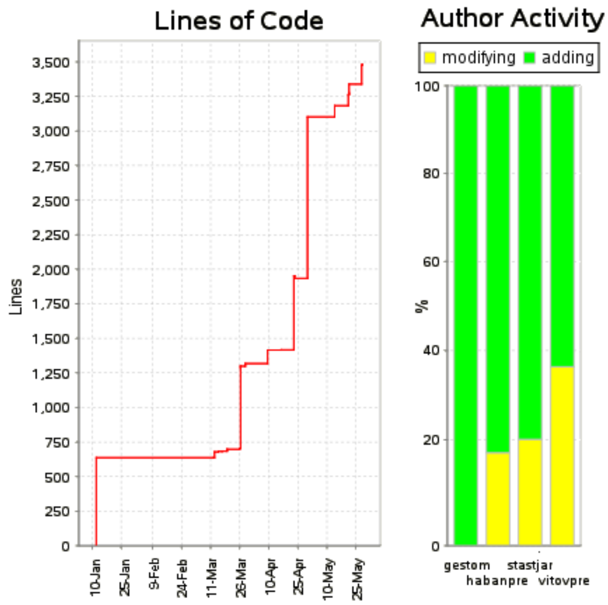


Fig. 5: Example of a students' VCS activity report

The labs consist of fourteen sessions (one per each semester week) that take place in the university computer lab. One session lasts 90 minutes and it consists mainly of contact time with the teachers. Students can also contact the teachers individually during consultation hours. Besides, it is expected that students spend additional time working at home, or at lectures. Students are introduced to the labs and course organization during the first session. The last session is dedicated to evaluation of the final assignment. The final objective of the assignment is to create a program that will control the platform in order to explore and map an environment and it should be solved within four weeks. Prior to this task, four simpler assignments have to be solved by students, two weeks are devoted for each task. At first, the students have to implement a simple position controller of a mobile robot and use ranging sensors to detect obstacles. In the third assignment, they have to extend the previous solution to a "Bug" type algorithm. The fourth task serves as a basic introduction to image processing, resp. visual navigation. These assignments introduce students to the usage of the Player/Stage framework, robot control, sensor data processing and fundamental robot skills and behaviours. Description of assignments is presented in the following subsections.

A. Robot Control

In the first assignment, students create a simple control algorithm that will navigate the mobile robot to a certain position in an environment without obstacles. Their control application has to determine forward and angular velocity of the robot based on the desired position and the current position estimated by the odometry. In this assignment, the students familiarize with the development tools and software framework. Moreover, they learn how to implement a simple controller of a mobile robot and encounter the first problems caused by the uncertainty in robot position. They also realize that a position from the odometry is defined in a local frame of reference and depends on the starting position of the robot.

B. Obstacle Detection

The second assignment is an extension of the first one into an environment with an obstacle. The robot has to use its rangefinders to prevent collision with objects in its path. The students have to decide, where to place range-finding sensors and design an algorithm, which processes real sensory data. The students are requested to deal with real sensors and to consider how the sensor output is related to detectable obstacles. Therefore, they have to design filters that deal with sensor noise, non-linear characteristics of the IR sensors [16] and false sonar echos. A measurement of the output characteristics is necessary to find a more precise transformation of sensor output to the obstacle distance. The assignment is fulfilled if a robot stops before an obstacle and continues its motion once the obstacle is removed.

C. Collision Avoidance

In this assignment, the students get familiar with the subsumption architecture. The students are requested to create an implementation of a reactive navigation algorithm of the ”Bug” class [17] that will control the robot in order to reach a given location in an environment with several obstacles. The robot combines algorithms from the previous tasks with new behaviour in one program for more complex collision avoidance. In the case that an obstacle is detected, the new behaviour has to actively circumnavigate the detected obstacle and recognize if the desired destination is not reachable.

D. Visual Navigation

The students have to implement a reactive navigation algorithm that will control the robot in order to reach an object of the selected color. The students do not have to implement image analysis procedure, but they use the CMUCam3 [11] camera, which provides image coordinates of objects with the specified color. As before, the assignment can be solved by two feedback loop controllers. One transforms object coordinates in the captured image to the robot angular speed and the second computes the robot forward speed from IR sensors and sonars. Similarly to the previous tasks, the students can use the Stage simulator before deployment of the algorithm to the real robot. An example of the camera, IR and sonar simulation is shown in Fig. 6.

E. The Final Task - Topological Exploration

Finally, the students have to combine and extend algorithms implemented in the previous assignments to make the mobile

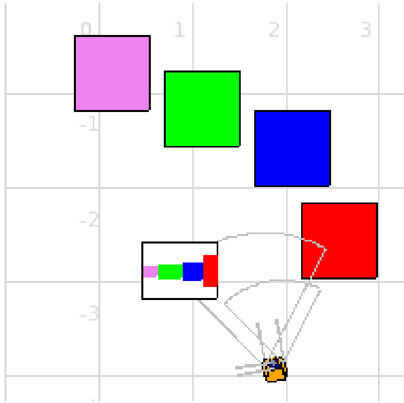
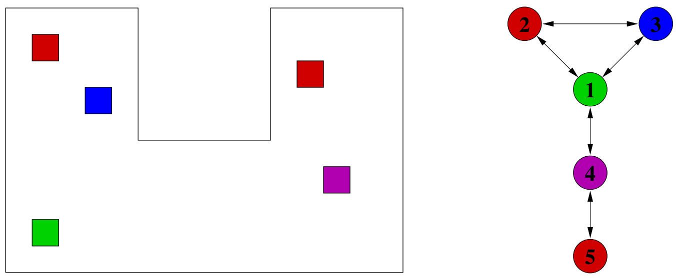


Fig. 6: An example of sensor simulation in the Stage simulator

robot capable of exploration of unknown environment. The robot has to create a topological map of the environment and use this map for planning and reasoning. An operational environment of the robot contains two types of objects - visual landmarks and obstacles. The visual landmarks are boxes with distinguishable colors and the obstacles are white walls or gray boxes. An example of an environment and a created topological map is shown in Fig. 7. A particular related real environment is shown in Fig. 8.



(a) environment (b) map

Fig. 7: An example of an environment and a created topological map

The time needed to create an exploration algorithm, which will work at any situation, is much longer than the time dedicated to the assignment. Therefore, the students have to specify constraints under which their algorithm will be able to successfully finish the exploration. Examples of such constraints are “a box with some color is not visible from a box with same color” or “at least one box is visible from robots starting position”. This helps students to understand the complexity of real world and to select between more complicated, less reliable solutions, and more robust solutions with clearly specified constraints. An important part of the assignment is a discussion why students assume particular constraint, and how the problem could be solved if such a constraint cannot be assumed.

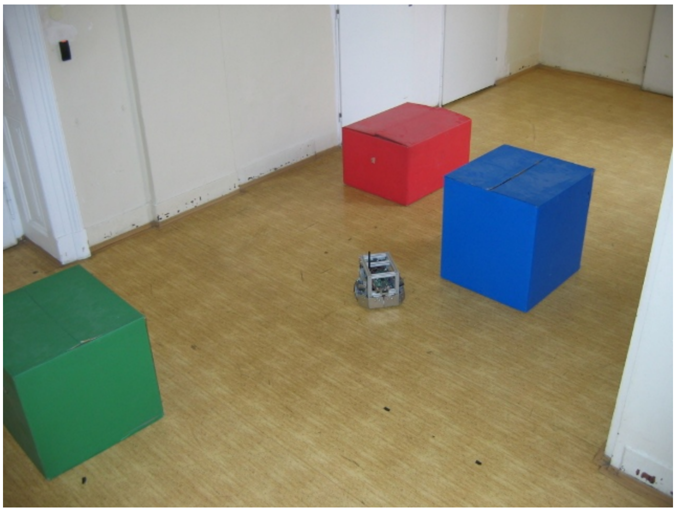


Fig. 8: An example of the real environment with MORBot

V. Conclusion

The presented ideas and concepts of the labs have been realized within three runs of the Mobile Robotics course. During the years of the course we gain practical experience and collect valuable observation and remarks from students. A part of them have been presented in the above sections of this paper. However, the part of them that are inspiring for further course improvements are presented in the following paragraphs.

One of the interesting observation is that even though a fully configured computer for development is accessible for students, they prefer to solve the assignment at their own laptops rather than using university computers. The students also tend to use multiple operating systems instead of Windows only installation and they are pretty much familiar with modern Linux based operating systems. However, we realized that for inexperienced users an installation of the Player/Stage can be quite difficult. It is mainly due to inappropriate dependency libraries provided by the used operating system (particular Linux distribution) in its standard installation. Thus, having a prepared "Robotic Linux Distribution", which will allow live usage from a CD or a USB FLASH drive, will be a great advantage.

We also observed that students hesitate to ask even a complicated question because they are afraid the question can be considered stupid. Sometimes, they stuck on simply solvable problem because they do not ask. Therefore the proactive approach of the teacher is advised. The teacher can request the students to show their progress and go through the students' codes with them.

Besides, we recognized two additional possible improvements. At first, students appreciate a mechanism that will inform them if they are behind the labs schedule, i.e. some kind of automated assignment evaluation. However, an important aspect of such evaluation has to be taken into account. It might leads to reduction of the assignments to solutions which are aimed to satisfy the submission automaton only. In such a case, the creativity and encouragement to do extra work beyond the basic assignments would be suppressed.

The second improvement is related to the used CMUcam3. The camera image is not directly accessible, thus students cannot see how particular color is changed under various illumination in real-time, which increase difficulty to understand the problem of color detection. This is in contrast to the main advantage of an intelligent camera that provides abstraction from the image. Even though the used on-board computer provides sufficient computation power for an eventual image processing, the main issues is in the USB interface (version 1.1), which does not support sufficient bandwidth to use a regular WEB camera with a raw image and sufficient frame rate and resolution.

We plan to address these improvements in further years of robotic course at FEE, CTU.

Acknowledgment

The work presented in this paper has been supported by the Ministry of Education of the Czech Republic under program "National research program II" by the project 2C06005 and by the Grant Agency of the Czech Technical University in Prague, grants No. SGS10/185.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *IEEE Robotics and Automation Mmagazine*, vol. 2, 2006.
[2] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," in *In Proceedings of the 11th International Conference on Advanced Robotics*, 2003, pp. 317–323.
[3] "The MORBot Project," http://imr.felk.cvut.cz/morbot.
[4] J. Chudoba, R. Mázl, and L. Přeučil, "A Control System for Multi-Robotic Communities," in *ETFA 2006 Proceedings*. Piscataway: IEEE, 2006, pp. 827–832.
[5] O. Fišer, H. Szücsová, V. Grimmer, J. Popelka, V. Vonásek, T. Krajník, and J. Chudoba, "A Mobile Robot for Small Object Handling," in *EUROBOT 2009 - International Conference on Research and Education in Robotics*. Paris: LAMPA, Arts et Métiers ParisTech, 2009.
[6] T. Krajník, J. Chudoba, and O. Fišer, "A Mobile Robot for EUROBOT Mars Challenge," in *Research and Education in Robotics: EUROBOT 2008 - Revised Selected Papers*. Heidelberg: Springer, 2009, pp. 107–118.
[7] V. Grimmer, "Platform for mobile robotics education (Bachelor's thesis)," 2008, (in Czech).
[8] H. Szücsová, "Software for mobile robotics education (Bachelor's thesis)," 2008, (in Czech).
[9] K. Havlovic, "Periphery board for mobile robots," Master's thesis, Czech Technical University in Prague, 2008, (in Czech).
[10] "http://www.gumstix.com," (accessed 28 July 2010).
[11] "http://www.cmucam.org," (accessed 27 July 2010).
[12] M. Janouch, "Embedded image processing," Master's thesis, Czech Technical University in Prague, 2008, (in Czech).
[13] "http://playerstage.sf.net," (accessed 27 July 2010).
[14] "Xming X Server for Windows - http://sourceforge.net/projects/xming/," (accessed 28 July 2010).
[15] "http://subversion.apache.org," (accessed 28 July 2010).
[16] "Sharp IR rangers - http://www.acroname.com/robotics/info/articles/sharp/sharp.html," (accessed 28 July 2010).
[17] V. Lumelsky, *Sensing, Intelligence, Motion: How Robots and Humans Move in an Unstructured World*. Wiley-Interscience, 2005.

Building robots as a tool to motivate students into an engineering education

Francis Wyffels, Michiel Hermans and Benjamin Schrauwen

Ghent University - Electronics and Information Systems Department
Sint-Pietersnieuwstraat 41, 9000 Gent - Belgium

***Abstract*—Today, robots have become an integral part of our society: children have robot pets, mobile robots are mowing our lawn and robot arms are assembling cars. Since people are clearly fascinated by these mechanical slaves, we were wondering: why not use robots as a tool to teach more abstract concepts in a practical way. Recently, a new course was added to the first year of the Bachelor's in the engineering program at Ghent University. In this course, first year students have the opportunity to get more hands-on experience through several projects. In this article we focus on one of these, titled 'How to build your own intelligent robot'. This work covers our approach for the practical sessions. Additionally, we elaborate on the low-cost robot platform that was built specially for this course, and which can be used easily by other schools or universities. Two years after the introduction of the robot project, we find that students not only like the sessions, but are very motivated to solve problems which would be otherwise considered too abstract and tedious.**

I. Introduction

Nowadays, robots are slowly finding their way from industrial settings to households, clinics and schools. Robotic pets, such as the Sony Aibo [1], are commercially available, robots such as Roomba are cleaning our houses and the first prototypes of social pet robots, such as the huggable robot Probo [2], for robot-assisted therapy are built. Similar to this evolution, robots are finding their way to the classroom [3], [4], [5] although often still hindered by economic constraints and some less successful stories. In [6], authors conclude that robots did not have any positive influence on student learning. However, other studies [7] show that robots can motivate students to actively do things that are not required for the course.

Recently, a new course was added to the first Bachelor's year of the engineering program at Ghent University. After an introduction of nine lectures which cover mainly written and oral presentation techniques, students have the opportunity to get more hands-on experience through several projects. Approximately 400 students have to pick their favorite subject from a list of 19 different projects such as constructing a small but precise catapult, design of a fish ladder and design of an intelligent robot. In what follows we focus on the project entitled 'How to build your own intelligent robot'[1]. This assignment is organized in several sessions during which students try to solve different, relatively small problems, each focusing on a particular problem in mobile robotics.

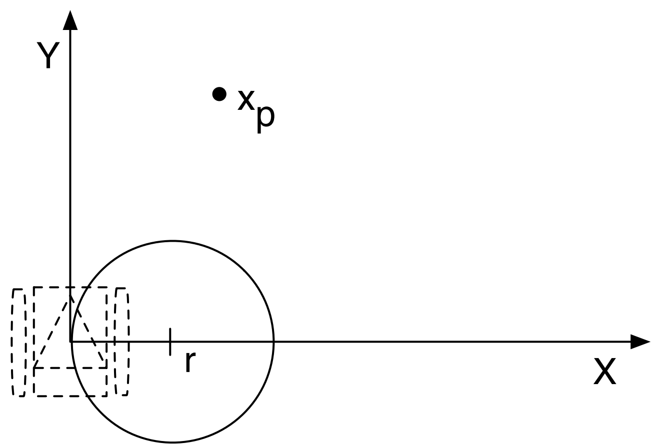


Fig. 1. Graphical depiction of the main geometry problem which the students have to solve during the first milestone. Throughout the course, students look for a solution such that their robot can drive as close as possible to a certain goal x_p. Students start with a two-wheeled mobile robot which is able to drive straight forward or turn with a certain radius r. Next, sensors are added and thus feedback is used to reach the goal. At the end, the morphology of the robot is changed such that the robot can cross obstacles and difficult terrain.

This work covers our approach for the practical sessions. In the following section we give an description of the course. Next, we elaborate on the low-cost robot platform that was built especially for this course and can be used easily by other schools or universities. After that, we describe the content of the hands-on sessions and the feedback we got from anonymous polls taken by the students.

II. How to build your own robot

The main goal of the course is threefold. First, we show that secondary school math can be applied to a real engineering application. Next we try to give the basics of several practical skills that are useful in robotics. Finally, the students have to improve their communication skills, specifically working in a team and presenting their results in oral and written form. In order to meet our main goal, we organize eight sessions which center around one practical problem: *programming a mobile robot such that it can reach a predetermined end goal in space* illustrated by Figure 1. The students try to solve this problem

[1] Additional material such as pictures and videos can be found on our website: http://reslab.elis.ugent.be/studentcourses

step by step, to break up to problem three milestones were set: *pétanque*, *golf* and *hiking*.

- In the first milestone, *pétanque*, basic concepts of mobile robot kinematics and open loop control are introduced. The students need to solve the geometry of the robot trajectory and perform measurements of the robot speed and movement.
- Next, the second milestone, *golf*, introduces light sensors and closed loop control of the mobile robot. Here, the destination of the robot is indicated by a bright light and a white mark on the floor. Students need to program an algorithm that lets their robot drive towards the light.
- Finally, for the third milestone, *hiking*, students have to rebuild the robot in order to allow its basic morphology to cross obstacles and rough terrain.

To complete a milestone, students have to do calculations and measurements such that they can implement a solution. In order to improve their communication skills they have to defend their solution with an oral presentation and to write down a report.

The milestones are divided over eight hands-on sessions which are organized weekly. At the start of the course, students form five teams of four students which are graded both as a whole and individually. The course counts for six credit units which indicates that an average student spends approximately 180 hours on the course, including classroom lectures. Evaluation is done throughout the semester by means of graded reports, graded oral presentations and evaluation of the given solution and collaboration. At the end of the semester, each group of students has to produce a final written report and a final presentation.

III. Low-cost robot platform

For the course, we searched a robot platform that is cheap, robust, easy to repair and flexible:

- The robot must be cheap in order to make it possible to provide enough robots such that students can work in small groups.
- The robot should be built robustly and should be easy to repair. When a large number of people are working with a device things can break or wear out easily. The robot platform should be robust enough to work under demanding conditions, and if something breaks it should be repairable without too much work.
- The robot hardware should be flexible such that the platform can be adapted to different circumstances and different tasks. It should be possible to add or remove different types of sensors. without taking the robot apart.

We found this combination of properties in a platform we build of Lego™NXT bricks and the Dwengo-board. A photograph of the robot platform can be seen in Figure 2. We designed the robot ourselves with as few pieces as possible. The Dwengo-board is a microcontroller platform with a PIC18F4550 and a wide range of onboard devices which can be used directly to build a robot without the need for designing additional electronics. It comes with a display, motor driver, a USB- and serial port and an expansion connector where sensors can be plugged in easily[2]. The microcontroller can be programmed in C using Microchip MPLAB IDE, the C-compiler is freely available for educational purposes.

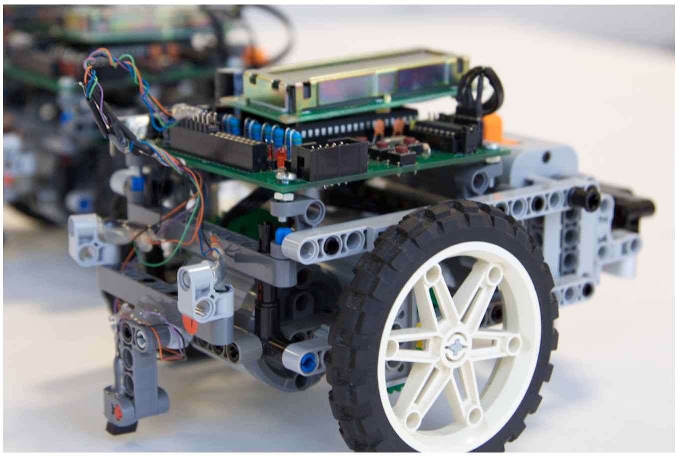


Fig. 2. Robot platform used in the course seen from the front. The construction is build by Lego™NXT bricks. The core of the robot is formed by the Dwengo-board which contains a PIC microcontroller. Through the expansion connector the robot can be extended with multiple sensors. In the visualized setup, two light sensor and one ground sensor was added to the platform.

The power supply of our robot platform is provided by six (rechargeable) AA batteries which can power the robot for the duration of at least one lesson. The total cost of the robot platform is estimated at 120 euro and is determined mainly by the microcontroller platform and the two Lego™NXT motors.

IV. Hands-on sessions

The core of the course are the eight weekly held hands-on sessions. In order to meet the course goals we choose to apply a combination (not necessarily all) of following teaching methods in one session:

- Homework: searching a solution for a problem through homework by investigation of existing literature and using creativity. Often, a session ends with an open question for which they have to seek an answer at home.
- Presentations: usually, the homework included preparation of a presentation in which they formulate their ideas, solutions for the posed problems.
- Brainstorm moments: the student presentations were followed by classical brainstorm sessions during which students try to extract the best elements from each presentation in order to come to a solution.
- Theoretical introduction: during each sessions the main concepts and workflows are introduced by means of a theoretical introduction. We choose to keep these introductions as brief as possible and they never last longer than one hour in order to get maximal attention.

[2]The full specifications of the microcontroller platform can be found on http://www.dwengo.org

Additionally, we interact (questioning, polls, ...) as much as possible to keep them attentive.

- Hands-on work: by applying several methods and doing measurements themselves, students get the most experience in how to bring theory into practice. Therefore, the main bulk of the time slot of the lessons was dedicated to this.
- Competition: at the completion of each milestone, competitions are held such that students are able to compare their results with other groups. They are assured that the result of the competition doesn't directly influence their grades.

As been said before, three milestones are divided over eight hands-on sessions. In order to reach the first milestone, during three hands-on sessions students have to find a solution to program a robot so that their robot can reach a certain goal (x, y) on a flat surface. At this stage, the robot has two wheels and no sensors. Additionally, students have to assume that the robot is limited to drive straight forward over a distance D or taking a turn with certain fixed radius r over an angle θ. Therefore they have to find the angle θ and distance D in function of the goal on (x, y). Some additional problems, such as finding the shortest possible path, have to be solved. Next, students have to measure the properties (speed, possible deviation when driving forward,...) of their robot and estimate the angle and distance so that the robot reaches as close as possible a given point. Since most of the students have no programming experience yet, programming is done through a graphical interface we provided in which they can specify how long the robot has to follow a certain path. The workflow of this milestone is comparable with the game *pétanque* for which a ball has to be thrown so that it lands as close as possible to the object ball. Such as the open loop control of the robot, during the flight, one can not intervene with the ball.

In the second milestone we introduce the concept of feedback. Two light sensors and one ground sensor[3] were added to the robots, while the destination was marked by a light source and a white sign. During four sessions students have to measure the properties of the sensors, program the robot using a state chart, and finally program their robot using the programming language C (using some helpful libraries and starting from a template such that not much knowledge of C is necessary). Again, they have to find the optimal (quickest) way to get their robot to the destination. One can compare this with the game *golf* for which it is possible to correct (by multiple strokes), give feedback, in order to get the ball into the hole.

Finally, the third milestone is devoted to finding a solution to drive a robot, cf. *hiking*, over difficult terrain. Until now, they refined the intelligence and the senses of their robot. However, this doesn't enable it to drive over obstacles or irregular surfaces. Therefore each team has to design a new robot that is inherently able to do so by how it is constructed. A good example of this, using a limited amount of pieces, is depicted in Figure 3. The idea is to introduce the concept of morphological computation [8], i.e. using morphology rather than a 'brain' (microprocessor), to solve locomotion problems.

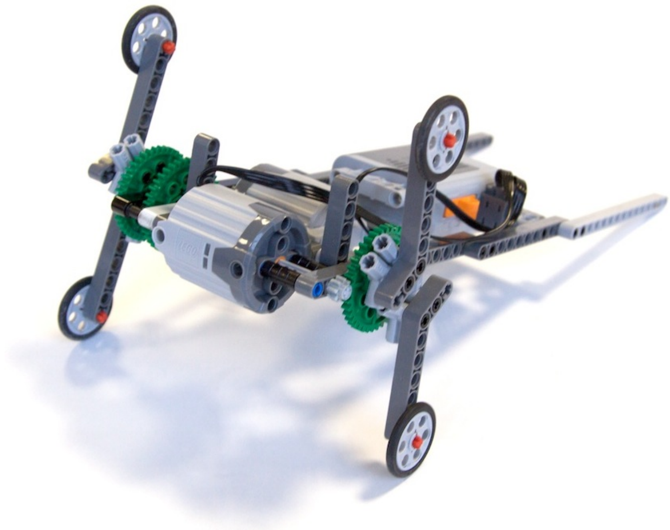


Fig. 3. A minimalist robot design which illustrates the concept of morphological intelligence: without being programmed to do so, the robot is able to go over obstacles.

For every milestone at least one presentation and one report has to be completed. After the first presentation and report a lecture is held during which pronounced good and bad things are pointed out. Additionally, for every report we gave some remarks to each group individually.

V. Feedback from the students

Since the first introduction of the course two years ago, we have had one official evaluation (organized by the faculty) and two informal evaluations (organized by the specific lecturers of 'How to build your own intelligent robot'). The overall conclusion is that students highly appreciate our project and our enthusiastic approach. In the official evaluation, students gave the project a score of 87%. Additionally, 90% of the students agreed with the proposition that the course increased their interest for engineering while the other 10% had no strong feelings about this question and thus didn't agree nor disagree.

On top of the official evaluation, we wrote an informal questionnaire, which probes for a more detailed opinion of the course. Again, we learned that students were charmed by the content and our approach. In order to find out whether students found the course useful we posed following three propositions with which they could strongly agree, agree, stand neutral, disagree or strongly disagree:

1) I have the feeling I had to use my creativity during the course
2) I have the feeling I learned from the hands-on sessions

[3]The ground sensor is distance sensor, but is used to measure the reflectance of the underlying surface. This allows the robot to detect it has reached its end goal, which is marked with a white spot

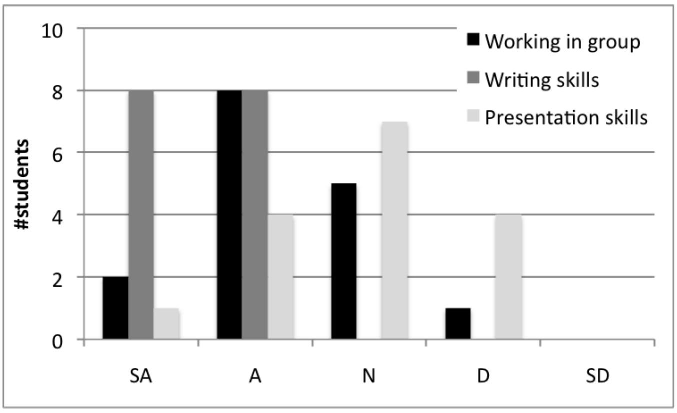


Fig. 5. Results from an evaluation which asks for the success of the course in increasing their working in group skills, writing and presentation skills. Students can strongly agree (SA), agree (A), being neutral (N), disagree (D) or strongly disagree (SD).

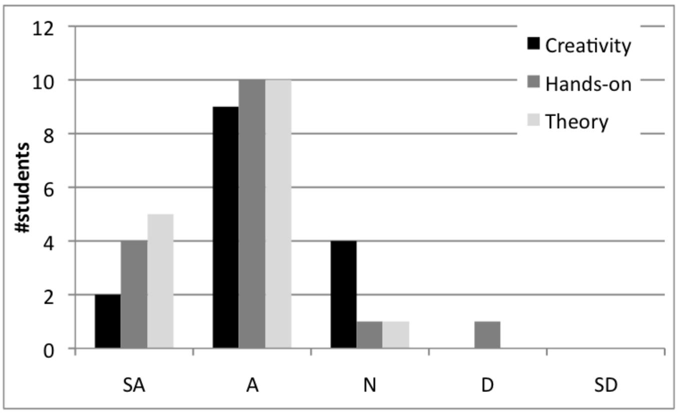


Fig. 4. Results from an evaluation which asks for the level of creativity they had to use, wether they learned from theory or not and wether they learned by doing things themselves. Students can strongly agree (SA), agree (A), being neutral (N), disagree (D) or strongly disagree (SD).

3) I have the feeling I learned from the given theory

On all three questions, students replied positive, as can be concluded from the graph in Figure 4. We belief that a combination of factors form the basis of our success. By implementing a robot platform in our course, students can immediately see the consequences of their thoughts and actions. For the same reason, hands-on sessions where practical engineering skills can be fully expressed form the core of the project. Apart from this, we believe that motivation and enthusiasm of the lecturers also play a role in the positive reception of the course. This enthusiasm is transferred to the students and motivates them to solve the posed problems.

We also wanted to know wether the students found the course helpful in order to improve their communication skills. The following three propositions were posed:

1) I learned how to work efficiently in a team
2) I learned how to write a good report
3) I learned how to give a good presentation

From the results presented in Figure 5 we learn that students are able to work in group, even if the members are not acquainted with each other from start, and that we succeeded in teaching them how to write a good report. However, some students believe that their presentation skills did not increase by our course. We believe we can overcome this problem in the future by giving more detailed and individual feedback of their presentation.

VI. Conclusions

In this work, we gave an overview of the course *how to build your own robot* which is held in the first year of the Bachelor's in the engineering program at Ghent University. In this course, students use high school mathematics to solve problems in the domain of mobile robotics. Additionally, student communication skills are increased by working in group, writing reports and giving presentations for a group. From student polls, we learned that students not only gain useful new skills but are also motivated to solve problems in the domain of engineering which could be otherwise found abstract and tedious.

References

[1] M. Fujita, "Aibo: Toward the era of digital creatures," *The International Journal of Robotics Research*, vol. 20, pp. 781–794, 2001.
[2] J. Saldien, K. Goris, S. Yilmazyildiz, W. Verhelst, and D. Lefeber, "On the design of the huggable robot probo," *Journal of Physical Agents*, vol. 2, pp. 3–12, 2008.
[3] M. Cooper, D. Keating, W. Harwin, and K. a. Dautenhahn, "Robots in the classroom - tools for accessible education," in *Proceedings of the 5th European Conference for the Advancement of Assistive Technology*, 1999.
[4] M. Goldweber, C. Congdon, B. Fagin, D. Hwang, and F. Klassner, "The use of robots in the undergraduate curriculum: experience reports," in *Proceedings of the 32th Technical Symposium on Computer Science Education*, 2001.
[5] J. Challinger, "Efficient use of robots in the undergraduate curriculum," in *Proceedings of the 36th Technical Symposium on Computer Science Education*, 2005.
[6] B. S. Fagin and L. Merkle, "Quantitative analysis of the effects of robots on introductory computer science education," *Journal on Educational Resources in Computing*, vol. 2, 2002.
[7] J. S. Kay, "Robots in the classroom ... and the dorm room," *Journal of Computing Sciences in Colleges*, vol. 25, pp. 128–133, 2010.
[8] R. Pfeifer and J. Bongard, *How the body shapes the way we think, a new view of intelligence*. MIT press, 2006.

An experience for teaching humanoid robotics in computer engineering studies

Martin Mellado
Instituto de Automática e Informática Industrial
Universidad Politécnica de Valencia
Camino de Vera s/n 46022 Valencia, Spain
martin@ai2.upv.es http://www.ai2.upv.es

***Abstract*—In this paper, the author presents his personal experience on teaching robotics, and more specifically on teaching humanoid robotics, within the studies for the Computer Engineering degree in the Escuela Técnica Superior de Ingeniería Informática of the Universidad Politécnica de Valencia, Spain. In the paper, first of all there is an introduction to the topic and how is the situation of robotics courses within computer engineering degrees in the more significant centres in Europe and Spain. Then, the paper progresses to explain how is the situation in Valencia, where there is a Robotics Course and a Robot Laboratory Project, explaining their main characteristics, contents and progress plan. The theoretical content for the unit on humanoid robotics is explained. The paper goes in details on the available equipment and the content of the laboratory sessions, the knowledge acquired by students and the exercises they have to do in order to pass this part of the course. Main issues covered on the Robot Laboratory Project are development of walking procedures for humanoid robots, the sensor control and the robot navigation in mazes. A student contest is organized so that the different student groups can show their abilities to program specific robot tasks, such as races and going up and down stairs and ramps. Last but not least, the paper will show the conclusions on this teaching experience on robot humanoids.**

Keywords- robotics; humanoid robots; teaching robotics; robotics in education

I. INTRODUCTION

In the last years and currently, humanoid robotics is one of the most difficult but popular topics in robotic research. Last advances on this field have produced promising results such as the Honda Assimo, the Sony Qrio and the AIST's HRP-2 and HRP-4. Every year there are more international conferences which include this topic, and specific conference for this topic, such as the IEEE-RAS International Conference on Humanoid Robotss [1], which shows the late advances on this field.

In opposition to this recent interest on research and diffusion on humanoid robotics, it seems that teaching on this field is not progressing according to the repercussion of efforts done in researching. The teaching in humanoid robotics is mainly focused for post-grade programs such as master and doctor courses and/or degrees. An example can be found in the 6th International UJI Robotics School IURS-2006 "Humanoid Robots" [2]. There are some other research courses on humanoid robotics, such as [3] in Carnegie Mellon University and [4] in University of Southern California.

Robot contests, such as RoboCup [5], are growing all around the world. During this year, the most significant robot competition at the European level has been the RoboCup Mediterranean Open, RomeCup [6], with different contests. One of the most significant one is the Football (Soccer) competition with standard platform league (then Nao robot of Aldebaran [7]) which has been won in 2010 [8] by the Spanish team Los Hidalgos of Instituto de Automática e Informática Industrial of Universidad Poltécnica de Valencia in cooperation with the Universidad de Murcia.

II. THE STUDIES ON ROBOTICS IN COMPUTER ENGINEERING STUDIES

A. Robotics in computer engineering studies

A review of the most representative university centres (schools and faculties) around Europe imparting the degree of Computer Engineering has been done in order to know the importance of robotics teaching in the most significant centres. The centres have been selected according to the Academic Ranking of World Universities in Computer Science – 2009 [9] to select the three most significant centres in Europe and in Spain.

For Europe, none of the best three centres according to this ranking (in Oxford [10], Zurich [11] and Cambridge [12]) has any course related to robotics in the studies of Computer Science BA degree.

Related to the robotics courses in Spain, the analysis has been done considering the three most representative Spanish schools and faculties imparting the degree of Computer Engineering, which are, together with the ETSInf that will be commented next section:

- Facultat D'Informàtica de Barcelona, Universitat Poltiècnica de Catalunya [13]. In the new studies, there is a robotics course of 75hours including industrial robots and mobile robots [14].
- Facultad de Informática, Universidad Politécnica de Madrid [15]. There is no course on robotics in the studies for the degree in Computer Engineering. It will be included a course on autonomous robots in the Master Program.

In centres as in Escuela Politécnica Superior, Universidad Carlos III of Madrid [16] or Escuela Politécnica Superior,

This work has been partially supported by the Escuela Técnica Superior de Ingeniería Informática (ETSInf) de la Universidad Politécnica de Valencia.

Universidad de Almería [17] there are robotics courses now, but in both cases, these courses will disappear in future years for the new degrees adjusted to European Education Space. In Universitat Pompeu Fabra [18] there will be a course on robotics in the future when there is no one now.

From this study, it can be stated that, even considering that it is clear the important rule of computer engineers in the development of humanoid robotics in the future, there is a lack of formation in this field.

B. *Sudies on Robotics in the ETSInf, the school of computer engineering in the Universidad Politécnica de Valencia*

The School of Computer Engineering (Escuela Técnica Superior de Ingeniería Informática, ETSInf) is the result of the integration into one single institution of the Higher Technical School of Applied Computing and the Faculty of Computer Science due to the Bologna process. The ETSInf was officially created on February 27th 2009. This School inherits a long tradition of teaching and research in Computing that goes back to 1982.

The degree programmes taught at the School of Computer Science are:

- Computer Engineering (5-years programme).
- Computer Technical Engineering (3-year programmes).
- Library and Information Management (2-year programme as second stage).
- Bachelor degree in Computer Engineering (4-year programme).

The Computer Engineer undergoes extensive training in all computer-related areas. This solid training allows graduates to fit easily into different professional careers and to efficiently manage the ever-changing technological advances in the field.

In the last year of the degree course students can choose between some specialist pathways. One of them, Industrial Computing offers a complementary teaching in the application of computer engineering to the industrial systems and processes, including aspects such as real-time systems in industries, robotics, computer aided design and manufacture, automation and control of industrial processes, industrial instrumentation and computer networks, computer vision and digital image processing.

This general objective is implemented through the courses offered in the context of the pathway, which are distributed so that the student must be enrolled according to the scheme shown in Fig. 1.

From this structure it can be noted that there are two courses related to robotics, Robotics Course and Robot Laboratory Project, which will be introduced in the next section.

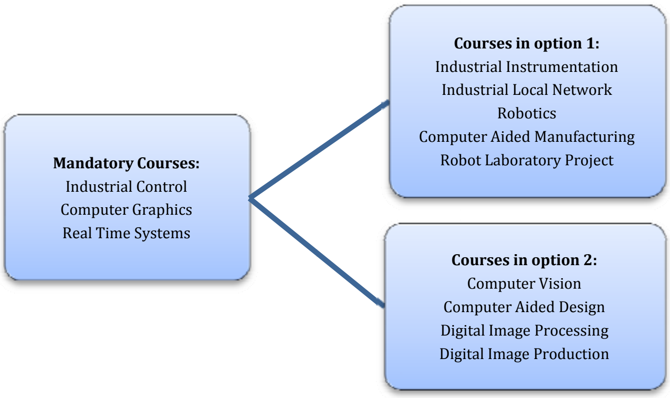


Figure 1. Structure of the Industrial Computing pathway

III. ROBOTICS COURSE AND ROBOT LABORATORY PROJECT

A. *Robotics Course*

The objectives of the robotics course are:

- To introduce the basic concepts of robotic systems in the platforms: industrial robot manipulators, mobile robots and humanoids.
- To learn the basic programming of each of the different types of robots.

The content of the course is according to the following program defined in parts and units:

- Part 1. Introduction to robotics
  - Unit 1. Basic principles of robotics
  - Unit 2. Spatial relations in robotics
- Part 2. Articulated robots
  - Unit 3. Industrial robot manipulators
  - Unit 4. Robot programming
- Part 3. Mobile robots and humanoids
  - Unit 5. Mobile robots
  - Unit 6. Humanoid robots

To fulfil the objectives, laboratory work is organized according to:

- Introduction to robotics:
  - Use of a robot simulation software
  - Spatial relations in robotics
- Industrial robot manipulators:
  - Programming simulation of robots
  - Industrial robot programming
- Mobile robots and humanoids:
  - Applications on mobile robots
  - Applications on humanoid robots

The robotics course has a total of 4.5 Spanish credits [1] meaning a total of 45 attending hours. The distribution of these hours is shown in Table 1.

TABLE I. HOUR DISTRIBUTION OF ROBOTICS COURSE

Didactic Part	Attending Hours	External Hours*
Introduction to robotics	13	5
Industrial robot manipulators	20	20
Mobile robots and humanoids	12	5
Total hours	**45**	**30**

* Estimated

The course evaluation is made using the following methods:

- Course project. It is a teaching strategy in which students develop a new and unique product by progressing with a series of tasks looking for effective use of resources.
- Online written test with open answer. Time trial, via web, in which the student builds his/her response. Students can use any material support.
- Laboratory tests. A short practical exercise that students must fill at the end of a laboratory session.

The final mark is obtained with a weight addition: 70% with the project coursework and 30% in continuous evaluation (on-line tests and laboratory tests).

All the Robotics Course teaching material is available in an OpenCourseWare website [19]. OpenCourseWare (OCW) is a web-based publication of course contents. OCW is open and available to the world and is a permanent activity. Its origin came from MIT [20] but nowadays is spreading through the world. In Spain, OCW is managed by Universia [21].

B. *Robot Laboratory Project*

The objective of the Robot Laboratory Project is to develop computer projects in the field of the material studied in the Robotics Course with the integration of concepts acquired in other courses studied, mainly within the Industrial Computing pathway.

With this objective, there is only one learning unit dedicated to the development of practical computer projects in the field of robotics.

Different projects are offered to the students, as for example:

- An automation project with the use of the industrial robot and auxiliary devices (conveyor tracking, rotation table, …).
- A project for mobile robots, in order to generate a program to solve a maze or to generate sweeping trajectories on a small room
- A project for humanoid robots, as will be explained in the next section.

1 Conversion: 1 ECTS credit = 1.25 Spanish credits; 1 Spanish credit = 0.8 ECTS

The students choose one project and work on it during the semester. All hours in Robot Laboratory Project are dedicated to the practical development of the projects, with 60 attending hours and 20 external hours (estimated).

The evaluation of Robot Laboratory Project is made using a team project, developed during the semester.

C. *Robotics teaching resources*

Resources available for the laboratory sessions of the Robotics Course and for the project work on the Robot Laboratory Project are:

- Robotics Laboratory, with an industrial robot (ABB IRB 140), two mobile minirobots (Khepera-II) and 11 humanoid robots (Robonova-1).
- Software for the laboratory sessions: MS Visual Studio 2008 for C++ programming, VirtualRobot software for robot simulation and programming, EditRapid for ABB robot programming using Rapid language and RoboBasic v2.5 for Robonova-1 programming.

IV. TEACHING HUMANOIDS

A. *Teaching Theoretical Concepts on Humanoids*

As it was seen in the contents of the Robotics Course, there is a unit for humanoid robots. The objectives of this unit are:

- To understand the basic characteristics of humanoid robots and their possible applications
- To learn the basic methods for humanoid motion control and its problems
- To understand the possibilities of humanoid minirobots
- To understand the development of a specific case of humanoid minirobot

The contents of this unit are:

- Introduction. This part covers the definition humanoid robots, their evolution compared to human evolution, differences between humanoid & other robots, social aspects to be considered, current problems and working fields, …
- Applications, detailing possible service applications but also including industrial ones
- Motion control. In this part, the problem of stability in humanoid robots is introduced, and possible kinematics models are basically introduced with some examples. Walking strategies for stride execution and methods to capture human motion and their possible applications to humanoid robots are also explained in this point
- Humanoid minirobots. Within this part, it is explained the following issues: the origin of minirobots, several commercial humanoid minirobots, the robocup competition and an example product developed at our university, microbiro, with its hardware and software architecture and main feasibilities.

B. Humanoid equipment for laboratory session

For laboratory sessions we have available a total of 11 Robonova-1 humanoid minirobots (Fig. 2). Hitec's Robonova-1 is a fully articulated, 12" high humanoid robot, which includes a HSR-8498HB digital servomotor in every joint.

Robonova-1 kinematics has 5 joints for each leg and 3 joints for each arm, giving a total of 16 joints moved via servos. These servos can be programmed developing users' programs in RoboBasic language with the development tool RoboBasic. Programs are downloaded into the robot controller Micom board MR-C3024 through a RS232 cable.

Servos can be modified in a range of degrees (from 10º to 190º), although some joints have a smaller range (for example, the ankle or the knee) because of physical constrains.

The position of joints can be defined with program sentences, and changing angles of motors in a certain way, the robot can make several movements like walking, running, dancing, etc.

Figure 2. Some of the 11 Robonova-1 robots used in the laboratory work.

In addition, the robot controller can manage some sensors (proximity, inclination…) and the data obtained with these sensors can be evaluated within RoboBasic programs running on the controller. For example, we can write a program with an 'if then else' sentence that depends on a variable whose value is the inclination of the robot. Depending on the position of the robot, we could execute the correct series of sentences to stand up the robot (if it is face up or face down).

Every Robonova-1 used in the course includes the following sensors:

- An infrared proximity sensor on its chest
- An infrared proximity sensor on each of its arms
- A tilt (inclination sensor) in its back
- An IR LED on its head to receive remote control orders.

Robonova-1 kits come with a remote control. Robot programs can get information from the remote control (for example, which button has been pressed) and use this data in the code as if it was another sensor. In this way, different programs can be run or robot motions and actions can be modified according to user's actions in remote control.

Robonova-1 is powered with a 5-cell NiMH rechargeable battery. In this course, for every Robonova-1 there are two batteries in order to be able to use the robot with a battery while the other one is recharging. Notice that for a full charge of the battery, it has to be plugged almost two hours, and then it gives about one hour of working time with the robot. Obviously, these times are approximated and depend on robot motions. Hence, battery save is very important, mainly considering that when battery is not fully charged, the robot can work in a wrong way.

RoboBasic is an exclusive BASIC extended programming language designed for controlling humanoid robots. With RoboBasic, commands that are needed to control a robot have been added to the general BASIC programming language. Because the grammar of RoboBasic is based on the general BASIC programming language, most of RoboBasic is similar to or the same as BASIC. In order to develop programs and download them on the robot, a development program, also called RoboBasic (v2.5) is provided with Robonova-1.

C. Laboratory sessions on Humanoids

The first two laboratory sessions within the Robotics Course intend to be a starting point for humanoids practical work with the use of a humanoid robot Robonova-1. There is a specific tutorial [22] prepared to introduce the robots to the students (as users and programmers) who never had a previous contact with this robot or its programming language RoboBasic. The hardware of Robonova-1 and some basic programming guides to start moving the robot are described at the tutorial.

Students are grouped in couples, so the maximum numbers of student in a session are 20 (only 10 Robonova-1 robots are used so that there is one extra for demonstrations). A very simple first exercise allows students to start working with the system, software and hardware and to try the connection of the robot to the computer and to verify their communication. The students have to program a task to control the light that is on the robot head. The students learn then how to use the RoboBasic system, including the steps to introduce a program, to compile it, to edit program errors, to download the program in the robot and to execute it.

Next step is to program movements. The students begin with the simplest way to make a sequence of movements: moving the robot manually to different positions, memorize them and play them. RoboBasic has facilities to do this and the students soon are moving the robot to different configurations.

After verifying the previous example, in the sessions some exercises related to robot motions and postures are given for the evaluation of the laboratory sessions. Possible exercises are:

- Keep in balance on one leg.
- Step forward.
- Step lateral (right or left).

- Step backward.
- Roll 45º around robot position (right or left).

During these laboratory sessions, the students have a close contact with the robots and learn mainly the stability problems that exist to get a proper motion control on humanoid robots (Fig. 3).

D. Projects on Humanoids

After the previously explained laboratory sessions, the students are ready to develop their own project on humanoids in the Robot Laboratory Project. In the academic course 2009/10, six students of 15 have chosen to develop their project on humanoid robotics.

Every pair of students is assigned a robot through the semester so that their developments are specific for this robot. They start their project with the common goal of developing walking procedures for the humanoid robot.

Then they start controlling the sensors on the robot, first with the tilt sensor. A program must be done so that the robot control its inclination angle and move its arm with opposite angle, so that the arm keeps always in vertical status. Note that in this way, the program is using a servo motion as an output indicator of a value.

The next step is to control the three infrared reflectance sensors for distance computation. From the sensor specifications, the students must compute an approximate value of the distance from the value read for the sensor. Calibration is a critical issue in this problem.

Figure 3. Two students working with the Robonova-1 miniobot.

The project is organized in a contest with the following four trials:

- A race for going up and down a stairs
- A race for going up and down ramps
- A race avoiding obstacles in a simple maze
- An open trial to demonstrate some robot programming abilities

The definition of the first three trials is shown in Fig 4. An event for the contest has been organized this year in May [23], with an attendance of more than 75 students of the degree to watch the trials. Some pictures are shown on Fig. 5 and Fig. 6.

V. Conclusions

Humanoid robotics is one of the most promising research topics in a close future for science and technology. Computer engineers have a crucial work to do in this field as robots must be programmed and controlled. Nevertheless, not many universities are including robotics in their Computer Engineering degrees. This paper shows an experience of teaching humanoid robotics in this studies, explaining in detail how is organized a course and a laboratory project.

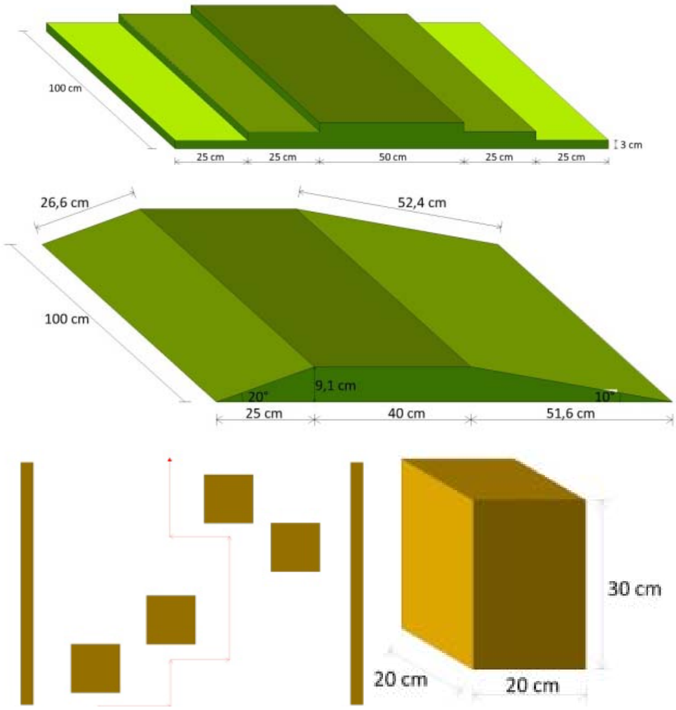


Figure 4. The definition of the stairs, ramps and maze trials.

Figure 5. A robot on the stair contest trial.

Figure 6. A robot on the ramp contest trial.

REFERENCES

[1] 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids09), UPMC, Paris, December 7-10, 2009. www.humanoids2009.org/

[2] Summer School on Humanoid Robots, Universitat Jaume I. Available online: www.robot.uji.es/lab/plone/events/iurs06

[3] Course 16-264 on Humanoids, Carnegie Mellon Univessity. Available online: <www.cs.cmu.edu/~cga/humanoids-ugrad/description.html

[4] Course on Perceiving and Controlling Humanoid Behavior, Univessity of South California. Available online: <robotics.usc.edu/~maja/teaching/cs599-hum.html>

[5] Robocup competition. www.robocup.org

[6] Robocup Mediterranean Open. www.robocup-mediterranean-open.org

[7] Aldebaran Robotics. www.aldebaran-robotics.com

[8] RomeCup 2010: premiati i vincitori in Campidoglio. Available on line: https://intranet.ai2.upv.es/alfresco/webdownload.jsp?docUrl=https://intranet.ai2.upv.es/alfresco/d/a/workspace/SpacesStore/fdc9f7cf-e929-45b2-a402-16fcd4af98e7/CS_RomeCup2010_vincitori.pdf

[9] Academic Ranking of World Universities in Computer Science. Shanghai Ranking Consultancy, 2009. Available online: <www.arwu.org/SubjectCS2009.jsp>

[10] Computing Laboratory, University of Oxford, www.ox.ac.uk/

[11] Department of Computer Science, Eidgenössische Technische Hochschule, Zurich, www.ethz.ch

[12] Computer Laboratory, University of Cambridge, www.cam.ac.uk

[13] Facultat d'Informàtica Barcelona, Universitat Politècnica de Catalunya. www.fib.upc.edu

[14] Grado de ingeniería en informática. Available on line: <www.fib.upc.edu/es/grau.html>

[15] Facultad de Informática, Universidad Politécnica de Madrid. www.fi.upm.es

[16] Escuela Politécnica Superior, Universdidad Carlos III de Madrid. www.uc3m.es/portal/page/portal/escu_polit_sup

[17] Escuela Técnica Superior de Ingeniería, Universidad de Almería. cms.ual.es/UAL/universidad/centros/politecnica/index.htm

[18] Escola Superior Politècnica, Universitat Pompeu Fabra. www.upf.edu/esup/

[19] Asignatura OCW Robótica. Available on line: www.upv.es/ocwasi/2009/7166

[20] MITopenCourseware. ocw.mit.edu/index.htm

[21] UCW universia. ocw.universia.net/es/

[22] M. Mellado, Programming a humanoid robot. Available in [19]. December 2009

[23] Los robots toman la ETSINF. Available online: <tv.inf.upv.es>

Autonomous Guided Vehicles Applied to Industrial Engineering and Management Studies

André Dias, Nuno Dias, Daniela Campos, Hugo Ferreira

***Abstract*—This article presents a framework to an Industrial Engineering and Management Science course from School of Management and Industrial Studies using Autonomous Ground Vehicles (AGV) to supply materials to a production line as an experimental setup for the students to acquire knowledge in the production robotics area.**

The students must be capable to understand and put into good use several concepts that will be of utmost importance in their professional life such as critical decisions regarding the study, development and implementation of a production line.

The main focus is a production line using AGVs, where the students are required to address several topics such as: sensors actuators, controllers and an high level management and optimization software.

The presented framework brings to the robotics teaching community methodologies that allow students from different backgrounds, that normally don't experiment with the robotics concepts in practice due to the big gap between theory and practice, to go straight to "making" robotics. Our aim was to suppress the minimum start point level thus allowing any student to fully experience robotics with little background knowledge.

I. Introduction

Simulation environments achieve great success in robotics learning especially when the students are at the beginning of their studies. Simulation is very useful for several reasons: students are just starting to learn robotics, the hardware is not ready, the operations place is very far away or inaccessible (e.g. other planets), the setup is often not operational (and to make it operational it can be difficult and time consuming), or even when the developers only want to test some small things and don't want to wait all the time needed for starting up the robots.

When students are just beginning to learn robotics, dealing with the hardware is complicated and simulation can teach the basics of robotics without having to deal with hardware problems.

The input focus of this framework was an Industrial Engineering and Management Science course in which students are required to learn how a production line works as well as everything around it. The students need to study production lines using robotics transporters. The aim is to use Autonomous Ground Vehicles to supply materials to a production line as an experimental setup for the students to acquire knowledge in the production robotics area. Students are expected not just to know the necessary concepts of industrial management but to understand how to apply them to a real-life problem/challenge.

André Dias, Nuno Dias, Daniela Campos and Hugo Ferreira are from School of Management and Industrial Studies, Rua D. Sancho I, 981 4480-876 Vila de Conde, Portugal {adias, ndias, danielacampos, hugo.ferreira}@eu.ipp.pt

The main objective is stock management in a production line using AGVs, where the students are required to address several topics such as: sensors, actuators, controllers and an high level management and optimization software.

Introducing the students to robotics requiring almost no prior knowledge on the subject was a challenge, therefore a two step approach was used: a simulation one and a practical one. There are several works presenting the simulation of a production line using AGVs [1], [2], and several presenting studies about the dynamics of AGVs [3], [4] but the solution presented allows the student to learn about the flow of materials in the production line, electronics and AGVs dynamics.

The paper is organized as follows: Section II describes the proposed architecture. Section III describes the experimental setup, developed to support the previous assumptions by having a two step approach: a simulation one and a practical one. Section IV provides some conclusions and discusses future work.

II. Proposed Architecture

The proposed architecture main objective was to develop a framework capable of managing the automated transport of materials in any given production line. In this framework it was necessary to keep in mind that first year industrial management engineering students have no prior knowledge of robotics and little knowledge in electronics, therefore, the proposed architecture had to be easy to interact with.

The main architecture of our framework can be seen in Fig. 1.

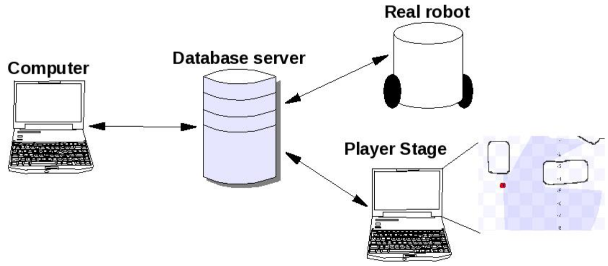


Fig. 1. Global architecture

The typical interaction used in the industry to manage the production is an Enterprise Resource Planning (ERP) which uses a backend database. Our approach was to use a database to interact with both the simulation environment and the application developed by the students. In this approach the students are expected to develop an application that interacts with the database according to their industrial management scenario.

The industrial management scenario is an industrial plant with physical representation of the stock area (both of raw materials and finished products) and the number of existing production lines. All the supply and collect points are marked and tagged.

The idea is that the students, using this scenario, would develop an application that determines the stock quantity of raw materials, the production rhythm (the maximum production rate of the production line is given), the stock quantity of finished products and controls the supply of materials to and from the production lines. The supply of materials is done using AGVs. Therefore, the students need to take into account the AGV dynamics since it adds time restrains to the supply of materials which can lead to lower production rates. The control of the AGV dynamics and route planning and optimization is not an objective for these first year students. Moreover, the simulation platform was chose so that these features could be added in future work to be used by more advanced students.

All the location points where the materials are collected or supplied are marked in the scenario. This is done by adding tags in a database. This database is given to the students and has tags for every supply or collect point and the initial quantity of materials in each point.

The students application interacts with this database to implement their supply management strategies.

This application uses a known open source tool from robotics community: Player project [6] (formerly know as Player/Stage).

This framework is composed by 3 layers:

- Player Project
- Database
- Application Program

A. Player project

The Player project [6] [5] is a multirobotic simulator (see Fig. 2) that uses a client/server model to manage the robot interface. It is possible to interact with a simple sensor or with an entire robot. There are several robots, devices and algorithms predefined in the original source but, if needed, new ones can be created.

The Player is capable to perform tasks by dealing with sensors and actuators interfaces (drivers) being simulated or from a real robot.

The Stage is a multirobot simulator in a two dimensional world. This software package integrates already a vast list of sensors models that can be easily modified and new sensors can be added. The Stage can talk with the Player allowing robot simulation if there isn't any real robot available.

The use of Player project allows the students and developers to avoid the time consuming task of developing the physics simulator and is a project with constant development, well established and flexible enough to add new robots such as AGV or robotic arms. Using a powerful tool such as Player project enables the further development of this framework, allowing the students to edit the industrial plant, adjust the number of production lines and AGVs, control the AGV dynamics and perform route planning and optimization tasks.

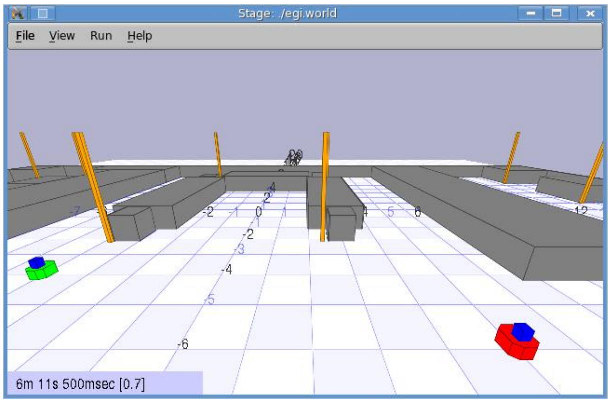


Fig. 2. The simulation scenario with two production lines in Player Project

B. Database

In a typical simulation environment such as the proposed Player project, the interaction with the simulator is made by using a UDP socket communication. This communication layer requires knowledge that is beyond the scope of the course in which this framework was implemented. Therefore, a database was developed in an open-source platform, MySQL, that will be used as the information layer, allowing the simulation environment to interact with the students applications.

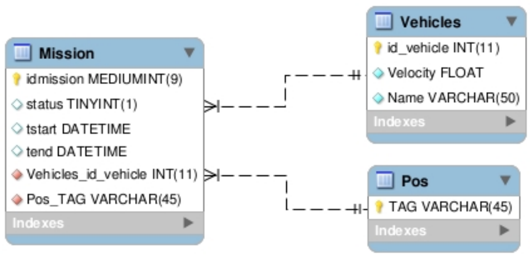


Fig. 3. Normalized implemented database

The database is composed by three tables (see Fig.3), two of them are associated with the simulation environment (*Vehicles* and *Pos*) and the third to the dynamic of the vehicles in the product line (*Mission*). The tables associated to the simulation environment are loaded with information regarding the AGVs available in each scenario and the supply/lifting end product positions in the warehouse. The database also allows the interaction with standard platforms or hardware solutions designed by the students without having to extend their knowledge on wireless networks or other type of communications.

C. Application program

The application program developed in GTK+ [8] is an interface that allows the student to initiate their simulation work (see Fig. 4). The GTK+ toolkit was chose because it is open-source and cross-platform which is important considering the different operating systems that students use.

Fig. 4. Management Software

This application was developed to be user friendly, having only two main controls responsible for starting and stopping the simulation, encapsulating all the hard work that is needed by the simulation environment. It is responsible for managing the requests from the database and interact with the simulator producing all the data logs required. Some data logs are showed in real time, so students can understand what is happening in the simulator window, and other logs are output to a file as being final results from the simulation itself.

In order to manage more than one robot a multi-threaded application is used by allocating a thread to each AGV present in the simulation scenario. Each thread is responsible to access the database requesting the mission for the assigned robot and then providing the path to execute the mission. The next mission will only be executed after the last one is completed and all timestamps from the mission are recorded in the database.

III. Experimental Setup

The proposed architecture was made available to the students allowing them to optimize the production line by controlling the AGV's dynamics without any prior knowledge on robotics. This approach allows the students to develop management applications that are able not only to study the AGVs dynamics but also other important themes such as stock and supply chain management which are areas of interest in industrial engineering and management sciences.

Using knowledge acquired in the programming, algorithm and data structuring course, the students develop graphic applications that can parametrize both the missions and the AGVs by interacting with the proposed database available in the framework.

The students' application interacts with the database by sending tags with the information regarding the location where the AGV needs to pick up material or where it needs to deliver the material. This information is read from the database and fed to the simulation environment where the students can visualize the AGV's movement but also obtain other useful information. The architecture allows the user to configure its' scenario by stipulating the maximum load and speed to the AGV's but also by introducing unique physical characteristics of the AGV (like wheel diameter, etc) and adjust the AGV's reaction based on a physical model.

A. Simulation Environment

The first step is simulation, allowing the students to validate the AGV dynamics relatively to the material supplying. In this phase it is possible to extract important information about the whole system which includes: supply times, route optimization and inventory management.

After starting the simulation environment, that consist of the definition of the layout that is already available (given by the teacher), and the application program installed in a class room, the students are now able to interact with the database.

The layout that is developed by the teacher has a challenge for each student that consists in a representation of all stages of the supply chain process(raw materials, production line and finished product). The black dots presented in the Fig. 5 are the tags where each AGV has to move. The example illustrates four spots (tags), two of them representing the supply materials (MP1_0002, MP1_0003) and the other two the production line supply points (LP1_0001, LP1_0002).

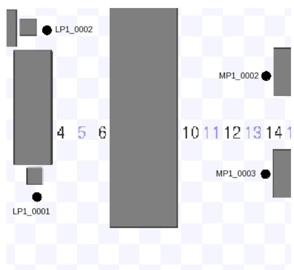


Fig. 5. A small layout example with tags. These tags are locations where the AGV has to move. The grey zones are areas where the AGV cannot pass through.

These events (or tags) have to be generated according to the students previous stock management planning and it can also serve as testing ground for different strategies. With this information, the students start to define the mission in the database. If the objective is to supply the production line LP1_0001, (Fig.5) from the supply warehouse tagged MP1_0001 the students perform the following SQL command:

>>MYSQL_CONNECT(IP_MACHINE, USERNAME, PASSWORD)

>>INSERT INTO mission (Pos_Tag_id, Vehicles_id_vehicle) VALUES(MP1_0001, 1)

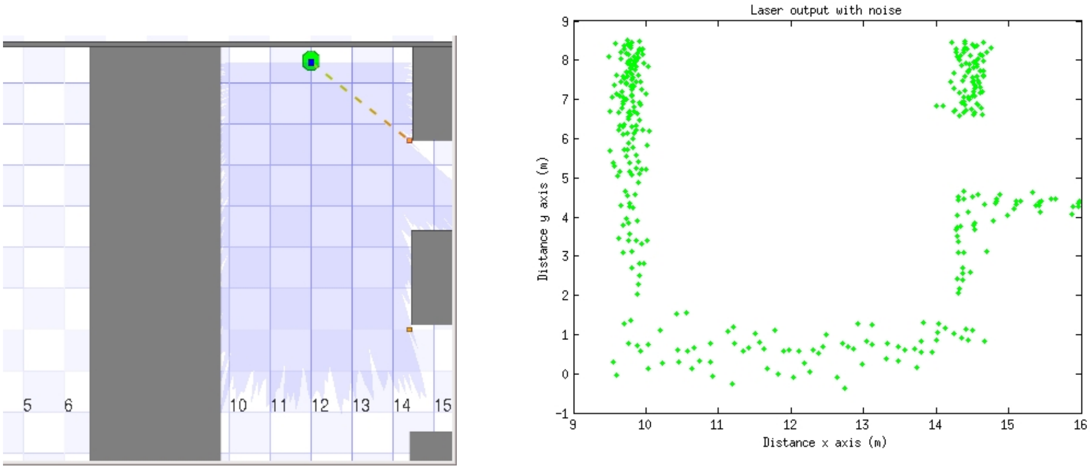


Fig. 6. AGV navigation with laser sensor (left). Environment mapping from AGV laser sensor (right).

>>INSERT INTO mission(Pos_Tag_id, Vehicles_id_vehicle) VALUES(LP1_0001, 1)

>>MYSQL_CLOSE(sock)

The AGV loading, travel and unloading times are managed by the application program defined in the layout setup. Besides this results the students can also learn several robotic topics such as path planning and sensor navigation (see Fig. 6 and Fig. 7).

A major advantage from this simulator is the capacity to evaluate different type of sensors considering the proposed scenarios. Figure 6 represents a sensor behavior with its own characteristic measurement noise. Students without in depth knowledge in electronics can choose different sensors for different scenarios by testing them, analysing their advantages and disadvantages from a practical point of view.

In figure 7 the AGV path and localization process that was simulated by the students can be observed. This is important because they can analyze the results and be able to evaluate different locomotion and navigation methods [7].

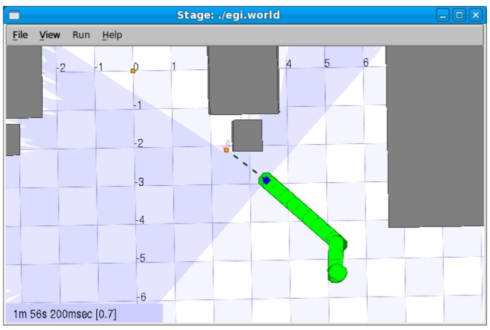


Fig. 7. This is an exemple of an AGV moving to LP1_0001 tag after receiving a command. The AGV path is also represented.

B. *Robotic platform*

The second step is the implementation of a small scale robotic platform to be used by the students and that have identical physical characteristics to the simulation step.

This allows the evaluation of the software developed by the students in a practical case as well learning the behavior of sensors, actuators and controllers in a real scenario. Each robot has a wireless communication system allowing the execution of the same instructions used in the simulation step.

The students can program the robots events the same way they did in the first step, therefore, this step adds no extra difficulty.

Since first year industrial management students don't have the necessary knowledge to implement the robots this step is intended only as a physical simulator of their production line stock management strategies. However it is an important stage to stimulate the students in achieving the best strategies and also to get them to take an interest in robotics.

The implementation of the robotic platforms is intended to be done by the students in the following years.

IV. Conclusions and Future Work

This article presents a framework for the students to acquire knowledge in the production robotics area using Automated Guided Vehicles.

This work allows the students to acquire skills in robotics area and supply chain management. The students are able to experiment with several practical scenarios and the presented simulation environment gives them an opportunity to see the results of their implementation.

This approach proved to be a good strategy to motivate the students to learn robotics and lead them to want to know more about this area.

Considering the impact of the proposed solution, we would like, in the future, to integrate the simulation environment with an e-learning platform so that it can be accessible by all students at school and at home. Also, the teacher/supervisor could easily add/change scenarios.

Further development of this framework to allow the students to have more control on the variables of industrial production planning was an initial idea and will be done in future work. The concepts necessary to take advantage of these new features are beyond the scope of first year students but this has great potential for more advanced students.

In another line of work, the students can develop their own small scale robot platform to interact with the present framework.

Currently we are working in the integration of Gazebo (a 3D multirobot simulator provided by Player project) in order to achieve more realistic feeling in the simulation output.

V. Acknowledgments

The authors would like to thank the School of Management and Industrial Studies and Porto Polytechnic. This work is sponsored by ESEIG and IPP. We would also like to acknowledge the support from Autonomous System Laboratory, LSA/ISEP/IPP.

References

[1] Saadettin Erhan Kesen, Omer Faruk Baykoc ”Simulation of automated guided vehicle (AGV) systems based on just-in-time (JIT) philosophy in a job-shop environment”. *In Simulation Modelling Practice and Theory 15, 272-284,* 2007.

[2] Che-Fu Hsueh ”A simulation study of a bi-directional load-exchangeable automated guided vehicle system”. *In Computers and Industrial Engineering 58, 594-601,* 2010.

[3] H. Martinez-Barbera, D. Herrero-Perez ”Autonomous navigation of an automated guided vehicle in industrial environments”. *In Robotics and Computer-Integrated Manufacturing 26, 296-311,* 2010.

[4] Manish Shah, Li Lin and Rakesh Nagi ”A production order-driven AGV control model with object-oriented implementation”. *In Computer Integrated Manufacturing Systems Vol.10 N.1, pp.35-48,* 1997.

[5] http://www.playerstage.sourceforge.net

[6] Brian Gerkey, Richard T. Vaughan and Andrew Howard. ”The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems”. *In Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003), pages 317-323, Coimbra, Portugal,* June 2003.

[7] Roland Siegwart,Illah Reza Nourbakhsh. ”Introduction to autonomous mobile robots”. *MIT,* 2004.

[8] http://www.gtk.org.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

European Land Robot Trial (ELROB)

Towards a Realistic Benchmark for Outdoor Robotics

Frank E. Schneider, Dennis Wildermuth, Bernd Brüggemann and Timo Röhling
Research Group "Unmanned Systems"
Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE)
Wachtberg, Germany
{frank.schneider, dennis.wildermuth, bernd.brueggemann, timo.roehling}@fkie.fraunhofer.de

***Abstract* — The European Land Robotic Trial (ELROB), which was held for the fifth time in 2010, is designed to compare unmanned ground vehicles in realistic outdoor tasks. It addresses the need to create a benchmark that can reproducibly compare and evaluate different robot systems. While robot trials like the DARPA Grand Challenge or the RoboCup have proven to be adequate benchmarks to compare robots systems in specific scenarios, the ELROB provides benchmarking in a wide range of tasks, which are oriented at prospective use-cases from a large variety of applications. In this paper we describe the ELROB 2010, the rationale behind the scenario design and how the trial has been implemented. We present the benchmarking system used to evaluate the robots' performance in the different tasks and, finally, have a closer look at some exemplary results.**

Keywords* — *robot contest; outdoor; benchmark.

I. Introduction

The European Land Robot Trial (ELROB) was designed to demonstrate and compare the capabilities of unmanned systems in realistic scenarios and terrains. It was founded by the European Robotics Group and is organised by the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), formerly part of the Research Establishment for Applied Sciences (FGAN). The trial is held annually, alternating between a more military and a mainly civilian focus. Up to now, the so-called M-ELROB was held at the military school in Hammelburg, Germany, whereas the civilian C-ELROB is performed at changing locations throughout Europe.

One major aim of the ELROB is to get a deep insight into the field of ground robotics by testing existing solutions in practical trials. These trials are conducted with a focus on short-term realisable robot systems and are designed to assess current technology while solving real world problems. Thereby, scenarios are not limited to the abilities of today's robots, but focus on realistic missions demanded by experienced users in difficult environments.

The ELROB presents a variety of realistic user defined tasks. These tasks include, for example, security missions, convoying, or reconnaissance by day and night. Although robotic contests are widely accepted as valuable means for benchmarking real outdoor robot systems, it is generally a difficult task to compare results from different contests or to generate a reasonable ranking even within one of the quoted scenarios. Omitting all details of task design, it is still obvious that many different parameters might have an influence on the overall benchmark for a mission. Taking the convoying scenario as an example, average speed, totally driven distance, or degree of autonomy are only one possible choice from a wider range of feasible parameters. Each parameter has to be measured in a precise and reproducible manner, which often raises serious problems, and afterwards has to be weighted in its influence on the final benchmark.

This paper will mainly address the latest ELROB, which took place from 17th until 20th of May 2010 in Hammelburg, Germany. We present the rationale behind the scenario design, the special demands of the co-organising military user, and the structure of the participants. After a detailed description of the different tasks of ELROB 2010, the remainder of the paper deals with the chosen benchmarking approach, thereby discussing the typical problems in the field of ranking systems, namely choice, measuring, and weighting of the different benchmark parameters.

II. Related Work

Generally, it is a difficult task to compare different published approaches in the field of robotics [1]. Thus, robot competitions are recognized as valuable benchmarks for real robot systems [2]. Several different competitions were held in the last years. Two of the largest and best-known competitions are the RoboCup [3] and the DARPA Grand Challenge [4], which are also recognized outside the robotics community.

While the RoboCup is currently targeted at indoor robots, the DARPA Grand Challenge aims to test and compare driverless cars. It started in 2004 with the rather simple task of following a 241 km long path, defined by several thousand UTM waypoints. Due to the difficult terrain and some teething problems, no participant was able to solve this task. In 2005, the task remained basically unchanged, and four participants successfully completed the race. In 2007, the DARPA Grand Challenge modified its goals from driving autonomously on difficult terrain to interacting with other vehicles in an urban scenario. Three teams could solve this very demanding challenge.

The ELROB is somehow comparable to the DARPA Grand Challenge in its attempt to gauge the functionality of outdoor robots. However, as already mentioned the ELROB presents a wider choice of user defined tasks instead of only one single scenario. Different users often express completely different requirements and specifications for robot systems depending on the possible fields of application. Instead of combining demands into one large scenario, like in the DARPA Grand Challenge, it might be more meaningful to have different tasks, which correspond to the various application scenarios. The following chapters present an exemplary description of the tasks for ELROB 2010.

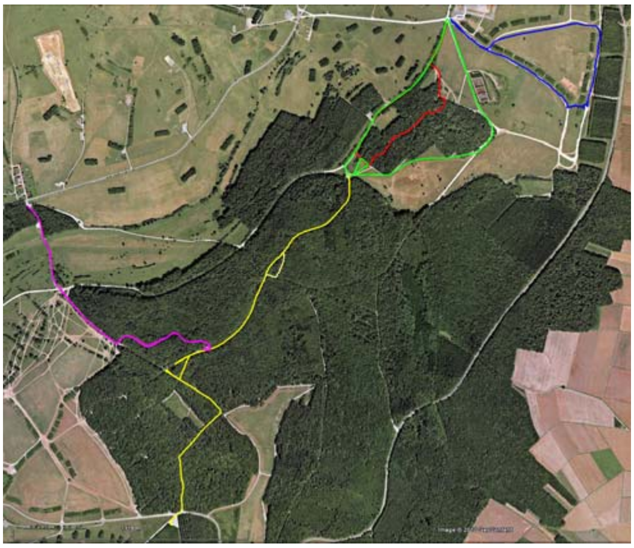


Figure 1. Overview of the movement oriented trials of ELROB 2010. The yellow track belongs to the approach part of the reconnaissance scenario, the purple one marks the mule scenario, the green, red, and blue tracks correspond to the different levels of the transport trial.

III. TRACKS AND TRAILS

The chosen area for ELROB 2010 lies within the training facility of the German military school in Hammelburg. Its size is of about nine square kilometres. The accessible roads have different qualities, ranging from well paved to heavy dirt roads. The environment is predominantly woody.

The different tracks on site were chosen to test specific aspects of robot deployment. Some challenges were common to all tracks; others were specific to certain scenarios. In preparation for the trials, every track was tested with respect to

- accessibility of the roads and paths,
- GPS reception, and
- radio reception between vehicle and control station.

By selecting areas with an elevation profile that does not support continuous radio communication from the control station, a certain level of autonomy was enforced. Thus, it was deliberately made difficult or even impossible to complete the missions in a purely remote-operated way.

Generally, the trials of ELROB 2010 have been divided into two major categories. On the one hand, there were scenarios with a focus on driving large distances of up to several kilometres. Due to the long distances in combination with the already mentioned hilly and woody character of the environment, it could be expected that solutions with a large degree of autonomy would perform best. Figure 1 presents an overview of the whole area. The different colours mark the different tracks and missions. In the following subsections, each scenario will be briefly described.

The second group of scenarios, on the other hand, had its focus on reconnaissance tasks. In these trials the robots had to search a given area, consisting of streets, paths, houses and grassland, for different kinds of targets, for example explosives, chemical or toxic waste, and radiation sources. Besides autonomy, other factors like manoeuvrability and a well-equipped sensor platform were of greater importance for this kind of tasks. The robots had to pass stairs and enter rooms through narrow doorways in order to reach all targets. In addition, although all targets could be seen with normal camera systems, additional hints like acoustic signals, heat or radiation sources had been installed for an easier identification. Therefore, systems with good sensor equipment had significant advantages during these trials.

Figure 2 illustrates some aspects of the reconnaissance scenarios. The leftmost picture shows a major part of the target area for these scenarios. From their starting point, the vehicles had to go there along some given, UTM-defined route. In the target area, open grassland with any kind of barricades, barriers or blockades had to be passed and different kinds of houses had to be entered and inspected. The middle picture presents an example target for the RSTA mission and the right one shows an exemplary radiation source in the NBC scenario. The details of the missions like the distances to be travelled and the exact kind of targets to be identified will be described in the following subsections.

A. *Movement Transport Trial*

Goal of this trial was to implement some kind of stable convoying in an outdoor, non-urban and off-road terrain with roads and paths ranging from asphalt streets to simple dirt roads in the forest. The convoy consisted of two vehicles of which only one was allowed to have a human driver. The second one had to be autonomous. The required path was defined by a small set of UTM waypoints, which were quite far from each other, so that the robot could not just drive straight lines between the waypoints but had to navigate along the roads and paths. The roads were part of the local testing ground for trucks, usually gravelled and led mainly through the forest. They were not marked, so mostly there was no clear distinction from the surrounding terrain. Sharp turns, dead ends and narrow passages occurred at several positions. No team member was allowed to inspect the trial area in advance.

The whole trial was divided into three levels of increasing difficulty, each consisting of a round trip with one common starting point. Looking at figure 1, one can identify the starting point at the connection of the green and blue track. The green track was the easiest one. The vehicles could use wide, well-paved roads, only with some very sharp turns to prove the robustness of the convoying algorithm. For the second level, a part of the original green route was replaced by a small dirt road in the forest; see the red track in figure 1. Level three, the blue track, was part of a special "off-road" truck testing site.

Figure 2. Illustration of the reconnaissance scenario of ELROB 2010. Left part – picture of the surroundings, streets, paths, houses, and grassland. Middle part – an example target in the RSTA trail. Right part – an exemplary radiation source in the NBC scenario.

Due to the very demanding character of this route and in contrast to the normally applied rules, the teams were allowed to have a look at the track in advance, in order to prevent possible harm from their vehicles. Each level was about 2.5 kilometres in length; the maximum time for completion of this trial was one hour.

B. Mule Transport Trial

The objective for this scenario was to let a vehicle serve as a “mule” and carry as much payload as possible between a loading and a turning point. Again, the terrain was woody and hilly with – partly very steep – roads of different quality. Instead of getting the UTM coordinates of the turning point directly, the robots had to follow a human who guided the vehicle from the loading point to turning point once. To simplify the mission for the teams this leader could be one of the team members, who himself was then led by someone from the organizing personnel. Thus, the leader from the team could wear, for example, specially coloured clothes or use special gestures.

After reaching the turning point for the first time, the robot had to shuttle the payload between the two points as fast and as autonomously as possible. In figure 1, the mule track is marked in purple. It was not known to any team member in advance. The distance between loading and turning point was about two kilometres; the maximum time for completion of this trial was one hour.

C. Reconnaissance Trial – Approach (Day/Night)

In contrast to last years’ approach, for ELROB 2010 the reconnaissance mission was split up into two independent parts. In the last years the objective was, first, to let the robot approach through unknown terrain into a designated target area and, second, search this target area for special, pre-defined targets. As already mentioned in the introduction for this chapter, the nature of these two subparts of a classical reconnaissance mission is rather different. The first part is more suitable for larger platforms with good and autonomous driving capabilities, whereas in the second part normally smaller robots with good manoeuvrability and special sensor equipment normally perform better. Consequently, nearly no participant was able to fulfil the complete trial during the former ELROB contests. Therefore, the organisers decided to separate the approach from the search in the target area.

Objective of the approach part of the reconnaissance scenario was now to reach a target point about three kilometres away. At that target point, an overview picture of a closely visible village should be taken. On its way towards the goal point, some intermediate waypoints had to be traversed. All these points were defined by their UTM positions. The yellow track in figure 1 marks one possible route for the approach, which passes all these intermediate waypoints. However, theoretically, the robots could choose their way freely. The track consisted of several narrow passages and even two dead ends, which can be identified by the small detours in figure 1. The area was completely woody and rather hilly, which notably complicated any attempt to maintain radio connection. The roads mainly consisted of forest paths with no clear distinction from the surroundings. As usual, no team member was allowed to inspect the area in advance. The maximum time for completion of this trial was one hour. The whole trial was first conducted under normal daylight conditions. The most successful teams had the chance to repeat the identical mission during the night.

D. Reconnaissance Trial – Target Area (Day/Night)

The terrain for the reconnaissance missions in the target area was an urban area within a valley. The urban area consisted of small buildings and homesteads, which are spread sporadically over the grassland of the valley (see left picture of figure 1). The buildings were connected with small roads and footpaths. Barricades, barriers and other blockades occurred at several places. From their starting point at the border of the valley the robots had to move along a given, UTM-defined route into a specific target area about 300 metres away. The relevant area for inspection was defined by a set of UTM boundaries. As for all other missions, no inspection of the operational area was allowed in advance. The maximum time for the completion of a trial was one hour.

The participants could attend up to three different kinds of such reconnaissance missions, according to their specific sensor equipment. The main difference between these possible missions was the type of targets the robots had to search. In the more general “reconnaissance, surveillance and target acquisition" (RSTA) trail, targets could be suspicious persons and vehicles, weapons, barricades and blockades, but also special acoustic signals like weapon fire or agitated discussions or heat sources, for example from vehicles or fires. The middle part of figure one gives an example. Those numbered orange cones marked all targets. The letters on the small white sheet in the middle of the picture should have been readable in the images acquired by the robot. Some of the targets could be only acquired from distances of up to 500 metres.

The "nuclear, biological, and chemical" (NBC) reconnaissance scenario required special sensor equipment, because there was no distinct marker for the targets like the orange cones in RSTA. Instead, the special physical or chemical properties of the – simulated – chemical agents, toxic industrial chemicals, radiation sources or explosives had to be measured. Finally, during the "explosive ordnance reconnaissance" (EOR) mission the robot had to inspect along a pre-defined UTM route. The robot had to search for suspicious objects like possible Improvised Explosive Devices (IED), ammunition, explosives, or wires under, beside or on the road, but – of course – without touching them. For all three types of missions, imagery and exact position of each target had to be acquired and transmitted to the control station.

IV. PARTICIPANTS

Ten teams in total participated in ELROB 2010, six teams came from European universities and four participants were from German and US industry companies. This section will shortly introduce their robots and vehicles.

The Institute of Real-Time Learning Systems of the University of Siegen took part with the robot AMOR. AMOR is a modified quad equipped with laser line scanners, PMD cameras and a stereo camera system. It uses a 3D environment model and fully featured local and global maps to drive autonomously, for example to follow a person or to pass given waypoints [5]. The Real Time Systems Group (RTS) of the University of Hannover participated with a robot called HANNA. Based on an off-the-shelf transport car, HANNA is equipped with various sensors for tele-operation, semi-autonomous operation and fully autonomous operation. The main sensors are two 3D laser range scanners used for environmental perception. In addition, multiple cameras, Differential-GPS, and inertial sensors are used for vehicle control. The Robotics Research Lab of the University of Kaiserslautern attended with their Robust Autonomous Vehicle for Off-road Navigation (RAVON). It is able to move fully autonomously, driven by a behaviour-based control system. It uses three 2D laser scanners and two custom-built stereo camera heads, as well as several additional sensors like GPS or a magnetic field sensor for localization purposes [6].

The Team MuCAR from the University of the Bundeswehr Munich (UBM) developed and operated the robot MuCAR-3. It is a modified Volkswagen Touareg, which allows computer control of steering, brake, throttle, and automatic gearbox. The team focuses on use of a Velodyne 3D laser scanner. The high definition 360 degree Laser Scanner is mounted on the roof of the vehicle. The RoboScout Team of the company BASE 10 SYSTEMS Electronics took part with the large robot GECKO. It is a four-wheel driven vehicle of about 3000 kg. Its speciality is its high manoeuvrability, because of its four separately steerable wheels. The robot can be controlled can be controlled via satellite or terrestrial communication, and can use a special small airplane as a relay station. The company Telerob presented their robot teleMAX. It is a track robot with flippers and a robotic arm. It is equipped with several cameras and is able to climb stairs. The team of Università degli Studi di Catania used a track driven vehicle that is used as an experimental research platform for volcano inspection. For autonomous navigation, the system is equipped with stereo camera-system, IMU, GPS and a SICK laser scanner.

The University of Versailles used a new and self-developed robot. The team is based on a student project that used a commercial electro kid quad as chassis for the robot. While moving, the environment is perceived through a laser range finder, sonars, infrared thermal sensors and webcams. The project addresses searching and rescuing people after natural disasters such as earthquakes. The company MacroUSA attended with a small Teleoperated UGV. The vehicle is equipped with a COFDM based vision system delivering a 360-degree view using three cameras. For navigation a GPS, IMU and a compass are included. However, the vehicle is not design to operate autonomously. The company ELP presented the PackBot, which was originally developed by IRobot. The tele-operated robot came in a basic version having on board only a camera and a manipulator.

V. RESULTS

For the presentation of the results of ELROB 2010 and for the discussion of our benchmarking system we will consider only a subset of all the conducted trials. As already mentioned during the description of the scenarios, the trials could be divided into two categories, one that focuses on autonomously driving large distances and the other one that more concentrates on steering capabilities and specialised sensor platforms. Since from our point of view this first kind of missions is the more interesting and more important one for the robotics community, we will omit the results for those scenarios dealing purely with reconnaissance in the target area. Additionally, it can be stated that actually all vehicles in those trials acted fully tele-operated and none of them was equipped with any special sensor equipment apart from (high-resolution and sometimes heat image) cameras. A detailed examination of all omitted trials – including the missions at night – can be found at [7].

TABLE I. WEIGHTS OF THE RELEVANT MISSION PARAMETERS

	Transport Movement	**Transport Mule**	**Recon. Approach**
Degree of autonomy	1000	1000	1000
Total distance	100	--	100
No. of round trips	--	100	--
Total runtime	10	--	10
Delivery of digital map	1	1	1
Delivery of GPS log file	1	1	1

The evaluation of the remaining missions – the mule transport trial, the movement transport trail, and the approach part of the reconnaissance trial – concentrated on parameters which were clear to distinguish and easy to measure. Table I gives a short overview:

- Movement Transport Mission
  Degree of autonomy, total distance driven, total runtime, delivery of a digital map and a GPS log file of the vehicle's track.
- Mule Transport Mission
  Degree of autonomy, number of successfully completed round trips between starting and turning point, delivery of a digital map and a GPS log file.
- Reconnaissance Mission – Approach
  Degree of autonomy, total distance driven, total runtime, delivery of a digital map and a GPS log file.

Obviously, autonomy was of overwhelming relevance to achieve a good result. In contrast to the other influencing factors, for which it is clear how they can be counted or measured, our definition of autonomy has to be explained. We used the ratio of total driving time and the so-called "manual interaction time", which starts at the moment when anyone interacts in any way directly with the vehicle or, for example, via an operation console. It ends in the moment when this interaction is over and the vehicle continues its autonomous work. The measurements for all the influencing parameters are normalized into the range [0; 1] and afterwards multiplied by the factors from table I, leading to a team's total sum for each mission.

A. *Movement Transport Trial*

Unfortunately, the movement trial suffered from very heavy rain. Due to the weather conditions, two of the registered teams, the University of Kaiserslautern and the University of Hannover, withdraw their participation. The University of Siegen and their robot AMOR managed the easiest first track without problems and could follow the leading car without any necessary interaction at an average speed of 5.8 km/h. However, at the rougher terrain of the second level the robot had considerable problems and often had to stop. As a result, the maximum trail time of 60 minutes ended after about 1200 metres of the second track.

The MuCAR-3 from the University of the Bundeswehr Munich performed very well and completed the first two levels of the trial at an average speed of 14 km/h and without any intervention of their safety driver. The team tried – without evaluation – even the very demanding third level and finished it with only one necessary stop. The third and last participant, the robot GECKO of the company BASE 10, also managed the easiest first track, but with some interaction, especially at sharp turns. Afterwards, the team aborted the mission because they feared damage for their vehicle due to the expected worse road conditions in the next levels.

Before looking at table II for the numerical results of this trial, it is important to mention, that the MuCAR team regrettably could not be evaluated because their mission setup was not compliant to the rules of ELROB 2010. For safety reasons they insisted on a human driver inside their car, who had to observe and – in case of need – control the actions of the robot. Therefore, the team started out of evaluation. As a result, official winner of this scenario was the University of Siegen, followed by the GECKO of BASE 10.

TABLE II. RESULTS OF THE MOVEMENT TRANSPORT TRIAL

Team	Robot	Result	Rank
University of Siegen	AMOR	1011	**1.**
BW University of Munich	MuCAR-3	n.e.	n.e.
BASE 10 SYSTEMS	GECKO	648	2.
University of Hannover	HANNA	n.p.	n.p.
University of Kaiserslautern	RAVON	n.p.	n.p.

B. *Mule Transport Trial*

The mule scenario started with the robot RAVON of the University of Kaiserslautern. Unfortunately, the team had technical problems, which forced the robot to stop after only a few metres. The second participant, HANNA of the Hannover University, successfully managed to follow its human leader and reached the turning point without larger problems. During the following autonomous shuttle mission, the robot had problems reaching the starting point again due to some technical faults. Shortly before the trial time was over, the team had to give up, only a few metres before arriving at the starting point again. However, corresponding to the benchmark parameters and since the vehicle was running autonomously most of the time, this result lead to the first rank in this scenario.

The University of Siegen and their robot AMOR reached the second place with a slightly worse performance. For a shorter totally travelled distance of about 2600 metres, the team needed longer and more frequent manual interaction with the system. The GECKO of BASE 10 SYSTEMS only drove a few hundred metres and then lost its way in the forest. Table III shortly presents the results of the mule transport trial. The team of the Munich Bundeswehr University was not evaluated for the same reasons as explained in the last section. Nevertheless, it is worth mentioning that the MuCAR-3 managed to shuttle between starting and turning point several times nearly without any intervention of the safety driver.

TABLE III. RESULTS OF THE MULE TRANSPORT TRIAL

Team	Robot	Result	Rank
University of Kaiserslautern	RAVON	206	4.
University of Hannover	HANNA	1000	**1.**
University of Siegen	AMOR	561	2.
BW University of Munich	MuCAR-3	n.e.	n.e.
BASE 10 SYSTEMS	GECKO	383	3.

C. Reconnaissance Trial – Approach

The scenario design for the approach part of the reconnaissance mission required a high degree of autonomy, because the very hilly and woody terrain made a permanent radio connection between vehicle and control station nearly impossible. Nevertheless, two teams tried to run fully tele-operated by using a fibre optic cable. The first of them, the company Telerob with their small robot teleMAX, reached the first dead-end about 500 metres away from the starting point. While trying to turn it cut the cable, which meant an immediate end of the trial. The GECKO of BASE 10 Systems performed better, because after the loss of the fibre optic cable it made use of the special radio relay airplane for transmitting the signals. However, due to the nature of the benchmarking system with its special emphasis on autonomy, it is clear that a tele-operated approach could not lead to good rankings.

The University of Kaiserslautern had to withdraw the participation because of technical problems with their robot RAVON. The remaining teams, the University of Siegen with the robot AMOR and HANNA from Hannover University, both tried an autonomous approach. AMOR only moved about 600 metres and then stopped due to some problems with a very steep forest path. When trying to free the robot, it unfortunately had a complete system crash. HANNA and the team from Hannover autonomously reached a distance of nearly two kilometres, but then faced some serious orientation problems. Since the team had no means of communication between the vehicle and its control station over this far distance, it had to give up at this point. Accordingly, table IV reflects the results of this trial. HANNA was the winner, and the University of Siegen reached a second place, although the GECKO drove faster and further, because of their autonomous approach.

TABLE IV. RESULTS OF THE APPROACH PART OF THE RECONNAISSANCE TRIAL

Team	Robot	Result	Rank
Telerob	teleMAX	368	4.
University of Siegen	AMOR	877	2.
University of Kaiserslautern	RAVON	n.p.	n.p.
University of Hannover	HANNA	1005	**1.**
BASE 10 SYSTEMS	GECKO	374	3.

It should be obvious, even through this very broad overview on the results of ELROB 2010, that the presented benchmarking system can be considered as fair only with regard to the special requisition of autonomy. Looking at the contest with a slightly different accentuation – for example on manoeuvrability, velocity or sensor equipment – immediately leads to different results. An appropriate extension and modification of the weighting parameters in table I might change the resulting ranks completely. Thus, it is important to keep in mind that a benchmark for robotic contests only reflects the organisers' demands and cannot reflect a robot's general suitability for outdoor robotic tasks.

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

The purpose of ELROB is not to get an overview over technological possibilities but to test outdoor ground robots in real world scenarios without regard to current limitations of these systems. Thus, the scenarios had to show the gap between desired and possible applications for today's robots. As could be expected, not every participant could cope with the designed missions. So the results were not unexpected and definitely not disappointing. In retrospect, two main problems could be singled out – reliable hardware, including reliable communication, and innovative autonomous software controller.

It is noticeable that while the industry generally had hardware in excellent quality available, they lacked the innovative autonomous control algorithms developed by the university teams. On the other hand, the university teams had most of their problems due to their restrained hardware budget and the required trade-off between functionality and cost. The combination from the robots used by industry and state-of-the-art control algorithms developed at universities might achieve much better results.

B. Future Work

From the 20th to the 24th of June 2011, the sixth European Land Robot Trials will take place in Leuven, Belgium [7]. The current working title is "Robotics in Security Domains, Fire Brigades, Civil Protection, and Disaster Control". Therefore, the missions will be designed having typical scenarios of those fields of application in mind. Again the trials will be designed to present scenarios as close to real world applications as possible.

REFERENCES

[1] A. P. Del Pobil, "Why do we need benchmarks in robotics research?", in Proceedings of the IROS-2006 Workshop on Benchmarks in Robotics Research, Beijing, October 2006.

[2] S. Behnke, "Robot competitions - Ideal benchmarks for robotics research", in Proceedings of the IROS-2006 Workshop on Benchmarks in Robotics Research, Beijing, October 2006.

[3] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara, "RoboCup: A Challenge Problem for AI", AI Magazine 1997.

[4] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge", Journal of Field Robotics, vol. 23, no. 9, pp. 661-692, June 2006.

[5] W. Seemann and K.-D. Kuhnert, "Design and realisation of the highly modular and robust autonomous mobile outdoor robot AMOR", In Proceedings of the 13th IASTED International Conference on Robotics and Applications, Würzburg, Germany, August 2007.

[6] M. Proetzsch, K. Berns, T. Schuele, and K. Schneider, "Formal Verification of Safety Behaviours of the Outdoor Robot RAVON", in Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO), pp. 157-164, Angers, France, May 2007.

[7] European Land Robot Trials (ELROB) website at http://www.elrob.org

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Subject „Robots“ at the CTU FEE in Prague – using LEGO robots to teach the fundamentals of feedback control

Martin Hlinovsky
Department of Control Engineering
Czech Technical University in Prague
Prague, Czech Republic
hlinovsm@fel.cvut.cz

Tomáš Polcar
Department of Control Engineering
Czech Technical University in Prague
Prague, Czech Republic
polcar@fel.cvut.cz

Abstract—**Since the beginning of the school year 2009/2010, new bachelor's program Cybernetics and Robotics offers a compulsory subject A3B99RO Robots in the first semester of study. This is a completely new type of subject provided jointly by three Departments at FEE CTU: Control Engineering, Cybernetics, and Measurements. The course with a limited number of lectures supporting required theoretical knowledge is focused mainly on independent laboratory work. The lectures are alternated by the departments mentioned above with laboratory exercises running in parallel. The assistants are supplied in laboratory teaching by promising and experienced master program students strengthening the concept of "learning by teaching".**

Study of technical subjects could be very difficult for new students entering the faculty with different background. Traditional study of theoretical disciplines without any clear relevance to "real" problems decreases motivation of many students, particularly those having difficulties with advanced mathematics. Many students feel almost betrayed when they have to learn three years just lemmas and theory since they want to study robotics, not mathematics. The core objective of our new subject Robots is to explore, in friendly way, students' independent thinking, and creativity and work in team. Although the theory is limited we believe that our students will eventually improve their theoretical knowledge as well. They are not obliged to memorize theoretical formulas but they have to learn how to use them. They start with something like child game trying to move robot from one place to another. Then, to succeed in more complicated tasks, they soon recognize that there is something behind the curtain and they start to ask how to solve the problems looking for a solution actively. If they reach this stage they are "trapped" - and become excellent and highly motivated students of control engineering.

Keywords- LEGO robots, feedback control, hardware, software

I. A3B99RO Robots: course organization

At the beginning of the semester the students are divided into small teams (4 to 6 students). Each team uses the basic set of the LEGO Mindstorms Education 9797, the set of the technical parts 9648 (additional passive components) and the network adapter 9833 (see the Figure 1). The teams design and complete the mobile robot with implemented control and program it to fulfill the specified and well-revisable tasks. Eventually, the teams prepare for the final competition with their robot directly fighting the opponents in activities attractive for broad audience.

Figure 1. Basic set of LEGO Mindstorms Education 9797, set of technical parts 9648

An essential element of the set LEGO Mindstorms Education 9797 and at the same time the "brain" of the robot is the central control unit known as LEGO ® NXT Intelligent Brick (see Figure 2) with a matrix display 100 x 64 pixels, 4 input ports for connection of the sensors, 3 output ports for connection of the motors, a speaker with 8kHz sampling frequency, having a possibility of Bluetooth wireless communications or an ability connecting to a USB 2.0 port. The intelligent brick and connected devices can be tested and partly controlled with the help of 4 buttons. Up to 3 servomotors can be connected to the LEGO ® NXT Intelligent brick which can be used as sensors for rotational speed measurement as well. The touch sensor, the light sensor (giving the robot an ability to "see" by measuring the intensity of the light and even recognizing different colors), the sound sensor or ultrasonic sensors (enabling the robot an orientation in the space, to find obstacles and to determine the distance from them) can be connected as well.

Figure 2. Intelligent LEGO ® NXT brick and connected sensors

II. PROGRAMMING LEGO ROBOTS

A. NXT-G

The programming language was named according to the programming language used by the LabVIEW program, developed by National Instruments, which is called only G. Abbreviation "G" comes from the fact that the programming language is graphical. Programs written in the NXT-G are thus built up of graphic blocks, with set up properties and subsequence, connected together. NXT-G is a joined product of LEGO and National Instruments and it is the basic programming tool for the LEGO MINDSTORMS NXT. The emphasis of the NXT-G is put on intuitiveness and simplicity of development environment including the programming process so that it can be used by primary school pupils with little experience in programming.

B. NXC

This text language derived from C language runs in the BricxCC on the standard firmware LEGO Mindstorms. It is very comfortable for those who want to program in both the NXT-G and the NXC because they do not need to upload new firmware after each change in programming environment. Working with the language abbreviating the phrase "Not Exactly C" is very comfortable and a programmer understanding at least the basics in C language becomes quickly familiar with the environment due to the almost identical semantics. Another advantage is that it is a freeware application. A disadvantage consists in complicated debugging of the programs. Unlike the NXT-G it is a purely textual programming without any graphics.

C. LeJOS-NXJ

The programming language distributed by Sourceforge is free and is available for Windows, Linux and MAC OS. Due to widespread expansion and knowledge of Java many users chose the LEGO MINDSTORMS LeJOS NXJ with its extensive libraries, which support interesting functions of the robot. The disadvantage is the necessity to change the firmware NXT which includes Java Virtual Machine replacing the standard LEGO firmware. LEGO firmware may be loaded into the NXT brick back using the LEGO software.

It depends on the students if they use one of recommended programming languages or use other ones (e.g. MATLAB toolbox developed at the University of Aachen (a product for users accustomed to programming in Matlab), RobotC (programming language based on C programming language), LeJOS OSEK (programming in ANSI C / C + +), or another one).

III. SOLVED TASKS IN THE CURRENT SCHOOL YEAR

In the current academic year 2009/2010 students solved two tasks:

A. Follow the line

The aim of the students in this task was to build and program a robot that would independently, without any further assistance, pass along the black line marked on the mat as quickly as possible and stop at its end (see Figure 3).

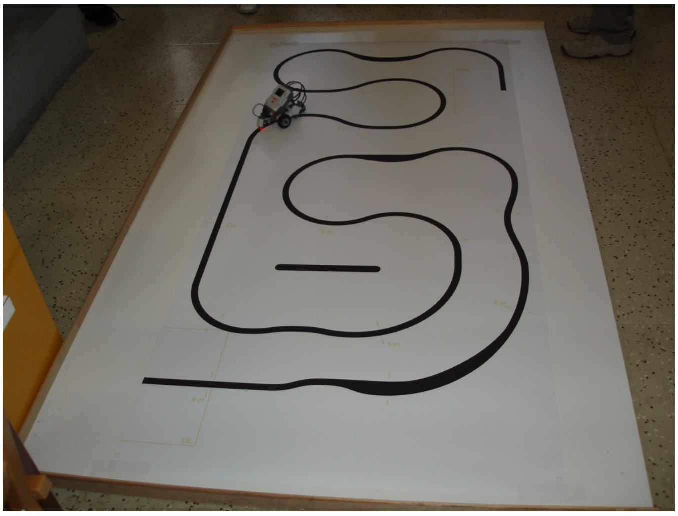
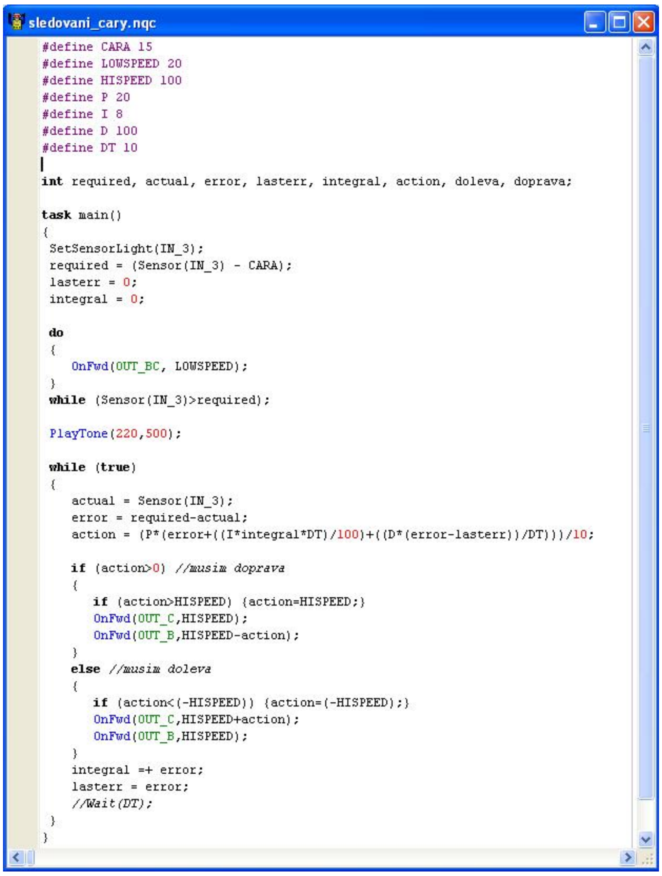


Figure 3. Task "Follow the line"

Students do not know the path ahead, they know only the basic parameters of the runway and that the total length of the line will be approximately 10m. The line may be arbitrarily extended not crossing itself with a minimum curve radius 20 cm.

The students are completely free to design the robot provide they use only the parts from the borrowed sets. After three weeks of preparation, testing and software debugging two-rounded competition of all teams followed. Four teams that had reached the best time had advanced directly to the final competition (held at the end of the semester), where they subsequently competed for interesting and attractive prizes.

The teams Jamais Contentés (Forever Dissatisfied) and Beer-go-home achieved the best times with 20.92s and 21.00s, respectively. The secret of success to achieve the best times was using PID controller for monitoring the line (example program in NXC see Figure 4).



```
sledovani_cary.nqc
#define CARA 15
#define LOWSPEED 20
#define HISPEED 100
#define P 20
#define I 8
#define D 100
#define DT 10

int required, actual, error, lasterr, integral, action, doleva, doprava;

task main()
{
 SetSensorLight(IN_3);
 required = (Sensor(IN_3) - CARA);
 lasterr = 0;
 integral = 0;

 do
 {
    OnFwd(OUT_BC, LOWSPEED);
 }
 while (Sensor(IN_3)>required);

 PlayTone(220,500);

 while (true)
 {
    actual = Sensor(IN_3);
    error = required-actual;
    action = (P*(error+((I*integral*DT)/100)+((D*(error-lasterr))/DT)))/10;

    if (action>0) //musim doprava
    {
       if (action>HISPEED) {action=HISPEED;}
       OnFwd(OUT_C,HISPEED);
       OnFwd(OUT_B,HISPEED-action);
    }
    else //musim doleva
    {
       if (action<(-HISPEED)) {action=(-HISPEED);}
       OnFwd(OUT_C,HISPEED+action);
       OnFwd(OUT_B,HISPEED);
    }
    integral =+ error;
    lasterr = error;
    //Wait(DT);
 }
}
```

Figure 4. Example of the program in NXC for the task "Follow the line"

B. Labyrinth

The aim of the second task was to build and program a robot that would independently, without any further assistance, pass through the maze from its beginning to its end as quickly as possible (see Figure 5).

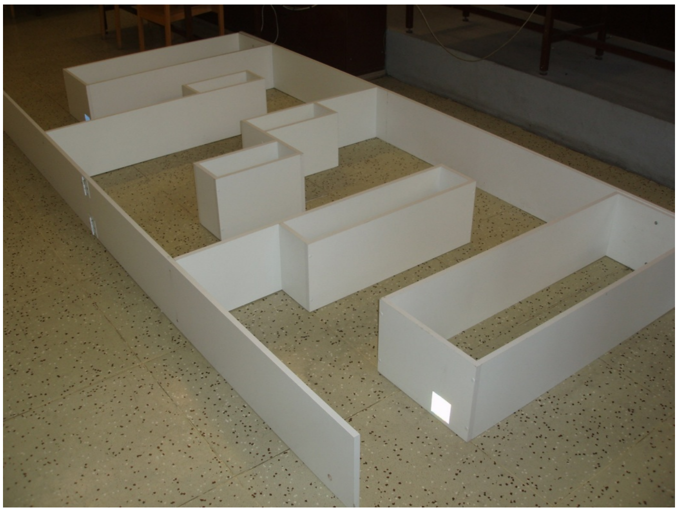


Figure 5. Task "Labyrinth"

The students were allowed to design robot quite arbitrarily only restricted by using the parts from the borrowed sets. Each robot had to pass through the maze from the start to the finish without any further assistance and external control. In the case of rules violation the team was immediately suspended from the competition. The minimum distance between any two maze walls was about 40cm. All maze walls were straight-line, 28 cm high, absent from any unforeseen bends and perpendicular to the bottom, i.e. there were no inclined walls. The time was measured by two light sensors located in the starting and finishing area. The total size of the maze was 330 x 160cm. The maze was built so that the shortest path between starting and finishing area was never coming back to the starting area, the passage led all the time to the target area. The choice of using sensors and control strategy depended solely on the individual teams. Of course, for the sake of orientation in the maze the robots could touch the walls. After four weeks of preparation, testing and software debugging again followed by a two-rounded competition for all teams the best fourteen went into the final contest (held at the end of the semester); their competitiveness was again strengthened by attractive prizes.

The students solved the passage through the maze using different ways; here we describe three of them. The team called DREAM TEAM constructed a robot according to the motto "The power is in simplicity" because they were aware of the fact that the decision making of the robot represents the largest waste of time. During construction of the robot they focused on the hardware part so that the code was as simple as possible and thus faster processing the information from sensors. They created a mobile robot (see Figure 6) that touches the wall by one wheel all the time.

The program (see Figure 7) activated all three motors at one time (this ensured that the robot was permanently crashing into the wall which it went along). There was one decision element (ultrasonic sensor) in the program analyzing whether the robot went along the wall or not. If the robot went along the wall the motor, which had been slowed down due to turning round, accelerated and the robot was speeding up. Conversely, if the robot went away from the wall, the program had slowed the motor so that the robot was able to pass the curve most efficiently.

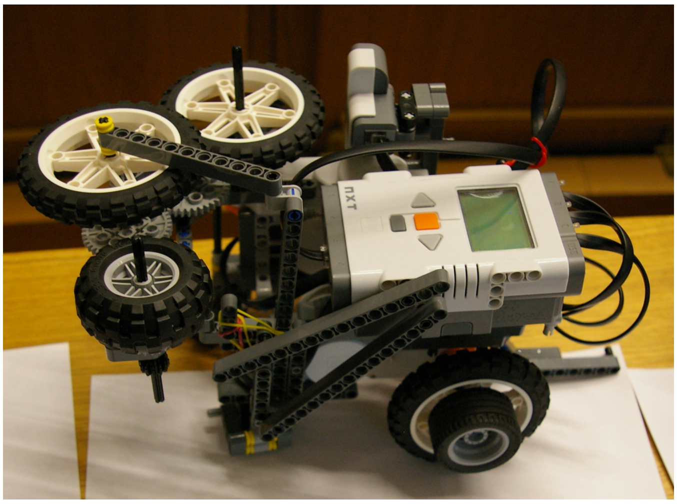


Figure 6. The robot of the team called DREAM TEAM

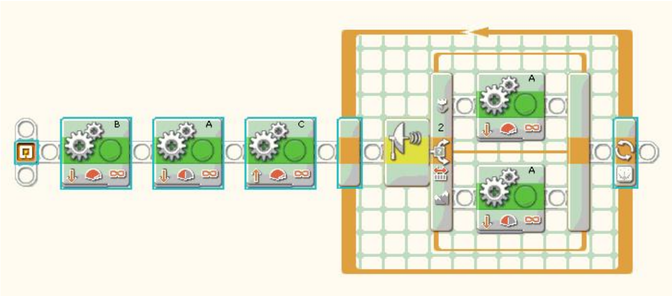


Figure 7. The program in NXT-G of the team called DREAM TEAM

The robot of the team called CENCUĽE was designed in a similar way (see Figure 8). Behavior of the robot was very simple (see program Fig. 9). The robot went straight on (motors A and B were switched on) and after hitting the wall it turned right using the motor C located in front. The ultrasonic sensor was used to detect a left curve. If the distance from the wall was more than 20cm the robot turned left (by decelerating the motor A). After re-approaching the wall the motor A was switched on again.

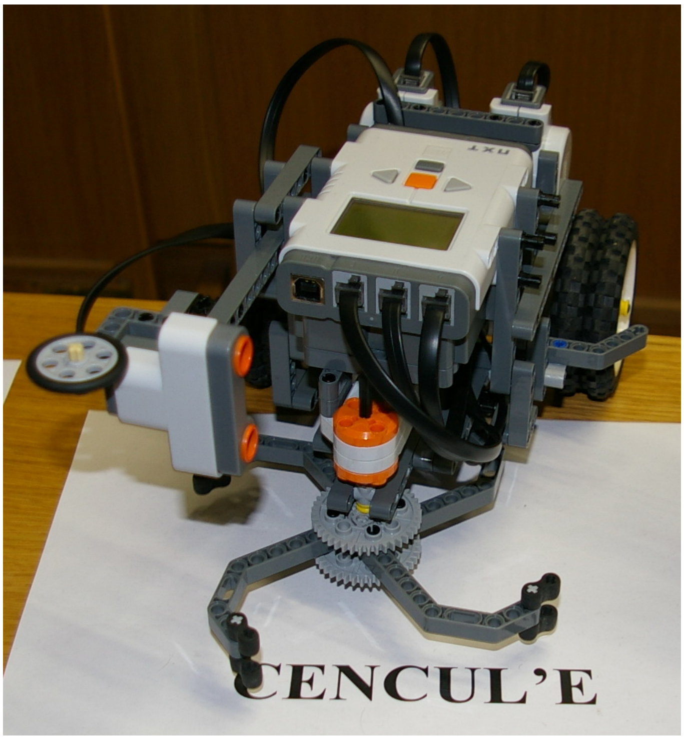
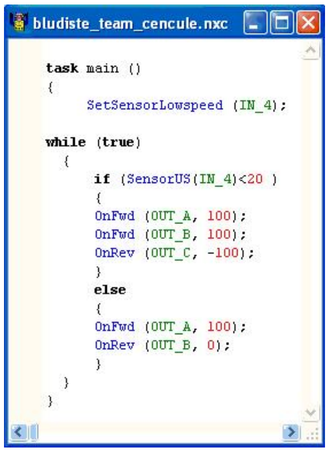


Figure 8. The robot of the team called CENCUĽE

The best design from both construction and software point of view was developed by the team Jamais Contantés (see Figure 10). They used two driven wheels with individual drives (the motors were located in the opposite directions for the compact shape and the smaller inertia around the vertical axis), two focal points, the gear of 2, 4 to fast on big wheels and the side guide wheels for the case of an impact. The robot used ultrasonic sensors (measuring the distance from an obstacle in front of the robot) and the light sensor (keeping the distance from the side wall, placeable on either side).




```
task main ()
{
    SetSensorLowspeed (IN_4);

while (true)
  {
     if (SensorUS(IN_4)<20 )
     {
     OnFwd (OUT_A, 100);
     OnFwd (OUT_B, 100);
     OnRev (OUT_C, -100);
     }
     else
     {
     OnFwd (OUT_A, 100);
     OnRev (OUT_B, 0);
     }
  }
}
```

Figure 9. The program in NXC of the team called CENCUĽE

The drive along the wall was solved in the code as one-level task with about 120 lines, due to the rapid passing a cycle using firmware 1.28 (allegedly many times faster than the original one) leading to the average cycle time of about 5ms. The program provided different speeds for direct drive, right and left turn and slowing down before the curve. Fast driving and right turn were controlled by two different settings of the PD controller using the light sensor. The code was optimized - due to long response the ultrasonic sensor measured the distance in front of the robot only if needed.

The following problems occurred during solving the task: low computing power, long reaction time of the ultrasonic sensor, small light sensor resolution, changing light conditions and big gap in the motor gear. Tuning the controller was possible only visually according to the behavior of the robot (it was not possible to store the values due to low computing capacity).

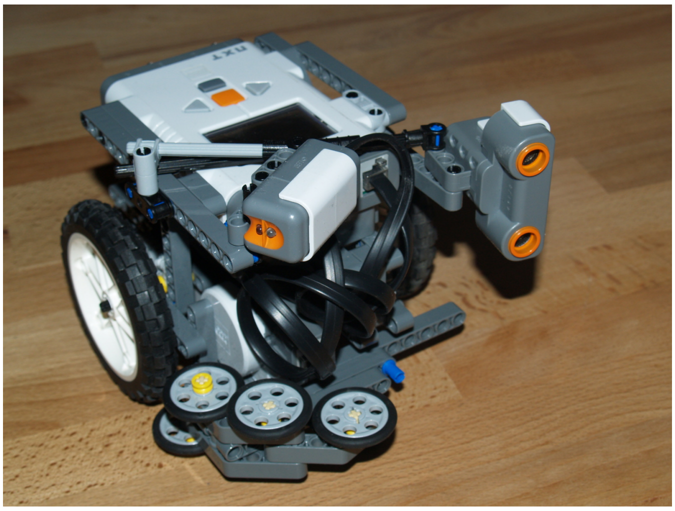


Figure 10. The robot of the team called JAMAIS CONTANTÉS

IV. Final Competition within the Subject Robots

The final round which was a part of the introductory course Robots of the new bachelor's program Cybernetics and Robotics, was held on Friday, 11/12/2009 from 3PM in the Zenger's lecture-room at the CTU FEE in Prague (see Figure 11 and Figure 12). For the final competition the task "Labyrinth" was selected. Each team was allowed to use once again the basic set of LEGO Mindstorms Education 9797 and the set of technical parts 9648. Best eighteen teams from CTU FEE in Prague with two teams attending the robotic seminars in Gymnasium Voděradská (high school) measured their power in the final contest for attractive prizes. The winner was the robot which passed through the maze as the fastest one. The entire final competition was broadcasted on-line via Internet. The parameters of the maze were announced just an hour and a half before the beginning, in order to allow finely tuning the programs of the mobile robots. In a two-rounded race both applause for the successful completion of the runway, bursts of laughter at the helplessness of a robot in the corners of the maze or even disappointment from the failure to complete the passage formed an exiting atmosphere.

Figure 11. Final competition within the subject Robots

Figure 12. Final competition within the subject Robots

Each team introduced some trick. The biggest hit was the robot which got over the walls (see Figure 13).

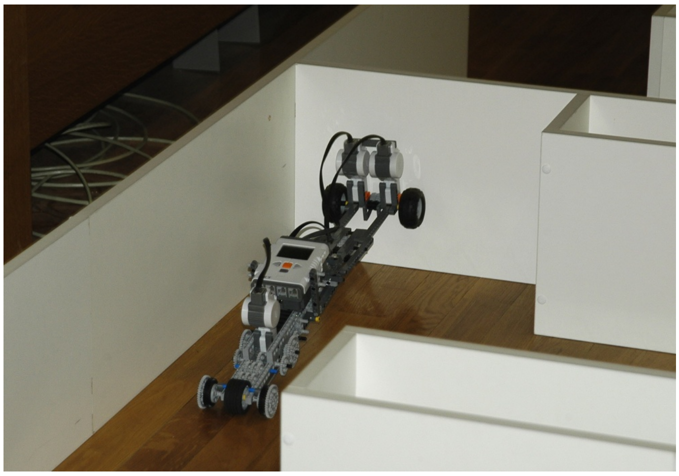


Figure 13. The robot which got over the wall

No one expected that the competing teams would reach such great times and the biggest surprise for us was when it became clear that students themselves have studied the basic theory of process control that would be lectured in later years, in order to gain an advantage over the other teams. We believe that without any doubt the subject makes learning more attractive. The whole atmosphere of the final round was quite exceptional. The winner of the contest became a robot team called "Cenculľe" with the time of 11.455s, second place earned the team called "Jamais Contentés" and third place took already mentioned robot, which got over the walls. Interestingly, the first and second teams were separated only by two tenths of a second. And how did finish the teams from the high school? Team "Robíci" finished the competition at an excellent ninth place just one second after the winner.

And what were the prizes for the winners? The winning team received a trophy, a barrel of beer and a certificate for a thirty-minute sightseeing flight by a helicopter starting in the airport Praha-Točná going in the direction of Karlštejn, quarries Great America, Mexico and Little America. The second placed team received external hard disks and a barrel of beer, the team on the third place received flash disks and a barrel of beer, the teams on the fourth and the fifth place were awarded too.

Information about the current subject Robots, videos and photos from the final competition including the flight of the winning team can be found on the website http://dce.fel.cvut.cz/roboti.

V. CONCLUSION

The subject Robots is timed in the very beginning of the study, deliberately at the time when the students ”know nothing”. However, it represents a playful way how to learn the basic ideas of automatic control, cybernetics, robotics, measurement and signal processing. The initiative is raised by themselves during solving the practical tasks. Right at the beginning of study the students recognize the principles of the creative engineering and research work.

The aim of the course Robots is to excite an interest in the branch, its main ideas and opportunities, while encouraging students to ask and study. We hope that the course will give them enough motivation to pass the difficult mathematical and technical courses during their studies. Moreover, the course and final competition is a very effective way to show FEE as the one of the most progressive faculties in the field of Robotics and Control Engineering in the Czech Republic.

Finally, we are preparing an international competition with our German colleagues for the next year.

ACKNOWLEDGMENTS

This research has been supported by the MŠMT FRVŠ project no. 33/100035/13135 Robots - the motivation subject of the new bachelor's program Cybernetics and robotics.

REFERENCES.

[1] LEGO.com Mindstorms – online. Avaiable: http://www.mind-storms.com.

[2] ROBORACE (in German) – online. Avaiable: http://www.ist.uni-stuttgarth.de/roborace.

[3] P. Trojánek, Usage of the LEGO robots in education, Prague 2009, Bachelor study, CTU FEE in Prague.

[4] J. Rieber, H. Wehlan, F. Allgőwer, “The ROBORACE CONTEST – Using LEGO robots to teach of the fundamental of feedback control”, IEEE Control Systems Magazine, October 2004

[5] D. Martinec, Usage of the LEGO Mindstorms Robot – Design and realization of the tasks, manual for programming in LeJOS-NXJ, Prague 2010, Bachelor study, CTU FEE in Prague.

[6] I. Moc, Usage of the LEGO Mindstorms Robot – Design and realization of the tasks, manual for programming in NXC, Prague 2010, Bachelor study, CTU FEE in Prague.

[7] T. Belik, Usage of the LEGO Mindstorms Robot – Design and realization of the special tasks, Prague 2010, Bachelor study, CTU FEE in Prague.

The Regional Center Concept for RoboCupJunior in Austria

Alexander Hofmann
Fachhochschule Technikum Wien
Höchstädtplatz 5
A-1200 Wien, Austria
Email:alexander.hofmann@technikum-wien.at

Gerald Steinbauer
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b/2
A-8010 Graz, Austria
Email: steinbauer@ist.tugraz.at

***Abstract*—The future wealth of western industrialized countries like the USA and the European countries will strongly depend on the availability of highly educated people for research and engineering. But already today these countries face a lack of such people. One reason for this is that science and technology is not very attractive for young people. Hand-on experiments and education had been proposed as a promising solution to this problem. In particular educational robotics has been very successful. In this paper we present the structure and results of the RoboCupJunior initiative in Austria which promotes science and technology by hand-on tasks in the robotics domain.**

I. Introduction

One of the most urgent challenges beside the aging society in western industrialized countries like the USA and the European countries face is that the future wealth can only be created by a knowledge-based society, research and technology. This leads to the fact that Europe already face a lack of several hundred thousand of researchers and engineers [1]. The lack is not only caused by the increased demand on well educated people by the industry or research institutions but also by the insufficient ability to educate enough young people as well by the lack of interest of young people in science and technology.

This story is not really new and has challenged people from academia, industry and politics for some years. One of the proposed concepts to cope with these phenomena is to use hand-on experiments or classes from the areas of physics, chemistry or technology. The advantage of these activities is that if the experiments and classes are well designed and exciting it's easy to get the attention by young people and to stimulate a sustainable interest in the fields.

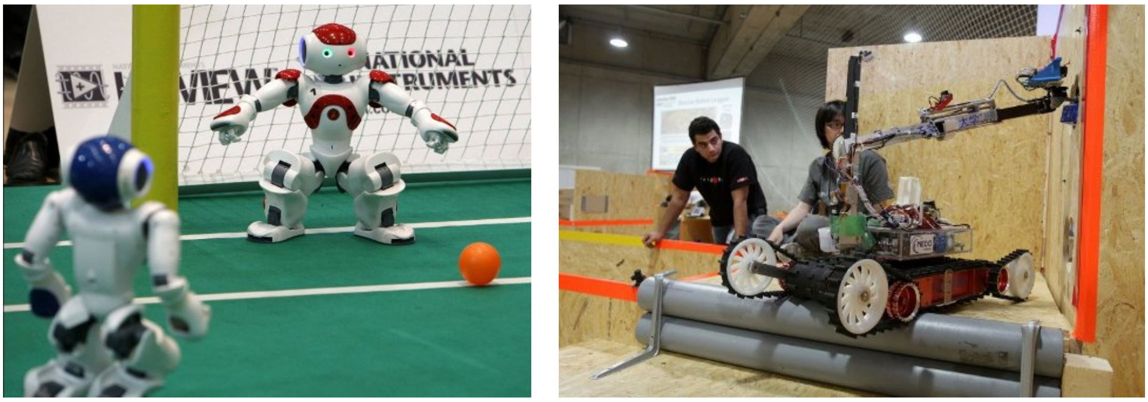
In order to challenge, inspire and motivate young people also robotics and robots has been used very successfully. The idea is simple but effective. Let the young people build and program little robots to solve a given task. This endeavor can be also done in teams and the results can be presented to the public and used in competitions. This playful work with an interesting and real problem allows exciting acquisition of knowledge and skills and a sensitization for challenges and opportunities of a profession in research or engineering.

During the years several different organizations and competitions have been developed which use the vehicle of building robots for educational purposes. These initiatives are quite heterogeneous and differ in the age of the target participants, the educational background of the participants, the organizations behind and the educational and social goals. The most popular and well known initiatives in particular in Europe are among others: (1) Roberta - Learning with Robots, (2) First Lego League [2], (3) Eurobot Junior and (4) RoboCupJunior [3].

The Roberta initiative tries in particular to engage young girls in science and technology. The initiative explicitly avoids competitions and is trimmed towards a course system used in schools. In order to engage in particular girls the initiative is very sensitive to gender issues and designs the courses and the course materials according to the special needs of young girls. The First Lego League is one of oldest and largest initiatives in the field. The advantages of this initiative are that it is a large world-wide initiative with several hundred thousand participants every year and that the task of the competition changes each year. The annual changed task allows focusing on actual topics relevant for the society like, e.g., climate change. The major drawback of the initiative is its close coupling to the products of Lego which are the only products allowed in the competition. The Eurobot Junior competition pursuit the same goal of a changing task every year but allows using arbitrary robots. RoboCupJunior is a worldwide initiative as well. RoboCupJunior is part of the research-oriented RoboCup initiative [4] and tasks in RoboCupJunior are related to the research focus of RoboCup like for instance rescue robots. The tasks of the competitions are only minor changed every year in order to make progress in the skills and solutions visible.

Beside this major initiative there are a number of national robot competitions which pursuit similar goals. These competitions are also popular in their proximity but act only on a regional or national level but sometimes with international participants.

Because the fact that the institutions of the authors already have been involved in RoboCup on the undergrad, grade and PhD education level and the fact that RoboCupJunior provides a long-term perspective for young people and a seamless migration from the secondary to the undergrad level we decided to work on the RoboCupJunior initiative in Austria. As no actions in the direction of RoboCupJunior had been set at that time nor an according infrastructure existed we started to promote and build up a RoboCupJunior network in



(a) Humanoid robot Nao playing soccer in the RoboCup Standard Platform League.

(b) A robot from the RoboCup Robot Rescue League examines a hole for possible victims.

Fig. 1. RoboCup Major Leagues

Austria. In 2007 we started with a set of activities to establish RoboCupJunior in Austria.

In the remainder of this paper we will introduce the structure of RoboCup and RoboCupJunior will present our activities to set up RoboCupJunior in Austria, will report results of the last three years and will discuss advantages and drawbacks of our activities.

II. The RoboCup Initiative

During the early 1990s research in Artificial Intelligence (AI) was trapped in a dilemma. On one hand the researchers were not able to fulfill their promises from the previous decades that if enough computational power is available machines can be developed which will execute relative intelligent task like the daily grocery shopping. On the other hand task which in general were assumed to be solvable only with intelligence were solved rather easily with enough computational power and showed no potential for research anymore. An example is the computer Deep Blue which defeated a grand master in chess [5].

After this, researchers thought about to define new standard benchmark tasks where intelligence is a precondition for a solution. Finally, researchers conclude that the best challenge for research on intelligent machines are those where elements of the real world like interaction, dynamics and uncertainty play a role. In 1995 Kitano and Asada proposed robot soccer as a challenge. Later in 1997 the first World Cup Robot Soccer, RoboCup for short, was conducted and the cornerstone of one of the largest and most important initiatives for the promotion of research, development and education in the area of AI and robotics was laid [6].

The long lasting success of RoboCup [4], 2010 the 14^{th} edition of RoboCup takes place in Singapore expecting about 2500 participants, is based on five major facts. At a first place the vision of RoboCup is a long-term, ambitious and inspiring one: *By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human soccer world champion.* Moreover, soccer playing robots are very attractive and exciting for a general public and allow an easy communication of the goals and challenges of research. Furthermore, the development of soccer playing robots raises a huge number of interesting research questions which are not only relevant for the RoboCup domain. Moreover, RoboCup takes care about the education of young academics and provides already students from schools a long-term perspective. Last but not least the different RoboCup leagues provide benchmark tasks for a great number of research groups with very different research focuses.

At the beginning of the RoboCup initiative robot soccer was the only research domain. But in the recent year further research areas has been established in order to accommodate actual development in the research of AI and robotics. Robots for search and rescue missions are one of these new domains. The focus lays on the development of robots as well as software which can support responders in the mitigation of disasters. This area does not only work on basic research but also work to convert ideas to products which can be used by responders. Another immense actual topic is service robots. The youngest RoboCup@Home league focuses on the development of robots which can assist people in their daily life. In sum RoboCup comprises 11 different leagues: five for robot soccer with different agents and robots, two for search and rescue robots, one for service robots and three for juniors. Figure 1(a) and 1(b) shows robots of two RoboCup major leagues.

III. The RoboCupJunior Initiative

The RoboCupJunior initiative came up in early 2000 and is based on different attempts to encourage children to work on robots and computer science [7]. In order to achieve the long-term goal of RoboCup, to defeat the human world champions in 2050 with humanoid robots, young researchers and even primary and secondary students became interested in science and technology. Moreover, this fact is even more important for the whole society as outlined in the introduction. RoboCupJunior is part of the international RoboCup research initiative and

(a) Presentation at RoboCupJunior Dance. (b) Setup at RoboCupJunior Rescue. (c) Before kickoff in RoboCupJunior Soccer.

Fig. 2. RoboCupJunior Leagues

therefore acts globally and provides competitions, networking, exchange of ideas and knowledge on a worldwide range.

The stated mission of RoboCupJunior as described in [8] is: *To create a learning environment for today, and to foster understanding among humans and technology for tomorrow.* The concept of RoboCupJunior is to use ”state of the art” teaching material, let students work in teams and to foster international exposure, exchange and contacts.

The RoboCupJunior initiative provides 3 different leagues for students to compete. Each league provides different challenges to solve, some are common, and some are league specific. The leagues are: (1) Dance, (2) Rescue and (3) Soccer.[1] Almost all leagues provide a further age differentiation. Primary participants, in the age from 10-14, and secondary participants, in the age from 15-18, are competing separately to guarantee fair competitions. Figure 2 shows examples for the different leagues.

The RoboCupJunior Dance league is known as the most creative one. The idea of this league is to provide a low-barrier entry competition. The task of the participating students is to perform a show together with one or more robots. The focus lies not only on technical skills and solutions, students still have to build and program robots, but creativity and fantasy play a major role. This league is less ”technical” than the other two. Since 2010 it is separated between *Dance* and *Theater* performances in order to allow and to emphasize different aspects of creative performances. The rules only apply few restrictions like the duration of a performance, size of the robots, security or organizational constraints. The students can combine music, stage design, costumes, spoken dialogs, robots and themselves into a stage performance. A jury awards points for specific criteria like creativity and entertainment value and for technical soundness as well.

The RoboCupJunior Rescue league is the most popular one. The reason for this fact is that only one robot is needed and the rules are very clear. Moreover, the background of the league is a realistic one. The task is inspired by the vision of RoboCup to develop search and rescue robots which assist human first responders in mitigating a disaster like an earthquake. Therefore, the scenario is inspired by a house with several rooms where a disaster happened. The robot has to autonomously navigate through the different rooms and to search for victims and to rescue them. Since 2010 two different types of competition arenas are provided, namely Rescue A and Rescue B. At RoboCup 2010 Singapore Rescue B will be only a demo. Rescue B uses heated victim dummies instead of a soft-drink cans that are used in Rescue A and is therefore closer to the rescue competition of RoboCup. This development should ease the migration of students from the junior to the major leagues.

The RoboCupJunior Soccer league is the most challenging one. Two teams of two robots play soccer against each other. As several robots (teammates and opponents) are involved and because of the dynamics of the game this competition is very challenging. In order to close the gap to the RoboCup soccer leagues the rules have been recently reworked completely. Now the soccer field of the juniors looks similar of them used in the major leagues, e.g., green carpet, colored goals and white field lines. The only differences beside the size of the field are that in RoboCupJunior soccer there are walls around the fields and an IR-emitting soccer ball is used. These modifications ease the sensing problem of the robots. Because of the challenging nature of this league a great number of students develop very advanced robots, e.g., omni-directional robots, by themselves rather to use standard robot kits like the Lego NXT.

One of the advantages of the RoboCupJunior initiative is that the students participate alongside the senior teams and world's top robotics researchers and engineers at the international RoboCup events. Moreover, in general the rules and tasks of the RoboCupJunior competition are only minor changed each year to allow students to improve their skills and to make these improvements visible.

Every year several thousand students around the world compete in national RoboCupJunior competitions where the best teams qualify for the annual international RoboCup competition. In this international competition around 1100 students in about 200 junior teams from approximately 40 nations compete.

IV. The Support Concept

Currently RoboCupJunior Austria comprises 7 regional centers spread in almost all regions in Austria. Centers are located in Dornbirn, Graz, Landeck, Salzburg, Vienna, Villach and

[1]The actual rules for the leagues can be found at: http://robocupjunior.org

Wels. These regional centers provide a set of standardized support actions to students, teams, teachers and schools in order to foster RoboCupJunior in Austria. First contact to RoboCupJunior is usually initiated by robotics experts from a regional center by giving a short RoboCupJunior presentation in front of students and teachers at their school. The presentation gives a first view on the topic and introduces RoboCup in general. The majority of the student and teachers become curious and want to know more about RoboCupJunior. Some of them even want to start immediately to program their own first robot.

In a follow-up the teachers can contact the regional center or visit the RoboCupJunior Austria website[2] to get further information on courses, material, support actions and competitions. The website has been recently refurbished and will further extend to new communication and networking tools like forums for teachers and students.

The next support actions we provide are introductory courses. These courses are designed for school classes and last from 3 to 4 hours. The courses mainly use Lego NXT robots to give a quick introduction in robotics, RoboCupJunior and robot programming. In order to ease the first contact Lego's own graphical programming language NXT-G is used for this first introduction. Hand-on experiments allow the student to use their new knowledge immediately in practice. After this introduction course the students are usually already able to solve small task related to RoboCupJunior.

In order to motivate and support teachers to use RoboCupJunior for teaching we designed special teacher's courses on robotics and robotic programming. The goal of these courses is to gain confidence on the robots and their programming as well to provide techniques like testing and debugging. These courses are well accepted by teachers of the different schools (different age groups and focuses) and we offer these courses once a semester in each regional center. For the long-term success of an initiative it is crucial to support and train teachers as they act as multipliers.

Advanced courses with other programming languages like Java or C have already been held at regional centers in Vienna and Graz and will be further improved in the future.

After the introductory course the students are able to start working on a particular RoboCupJunior league in their school. The usual plot is that the students work on a particular challenge from RoboCupJunior, solve this challenge, present their achievements during presentations at their school and finally compete in a national or even international competition.

If schools do not possess or cannot afford robot kits they can rent robot kits for free from the regional centers. We build up a considerable pool of equipment for this purpose. Teachers pickup the robot kit, i.e. Lego Mindstorm NXT kits, at the regional center and have to declare the intention to participate the next national Austrian RoboCupJunior competition. After the competition the teacher returns the rented hardware. Only lost or damaged parts have to be replaced by the school.

[2]RoboCupJunior Austria website: http://robocupjunior.at

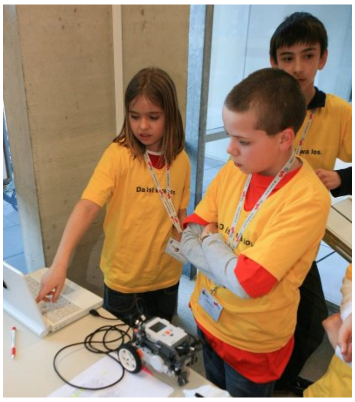


Fig. 3. RoboCupJunior open days does not only improve technical skills but also social skills like arguing for an idea.

During the project phase teachers work as mentors with their teams in classes. They work to build all necessary components and to develop software for their robots. The regional centers support them by answering questions, providing knowledge and communicating actively with the mentors and students.

Moreover, so called *Open Days* are offered to the teams where they can visit their regional center. On one hand it allows to test their robots on real competition arenas, i.e. soccer fields and rescue arenas, which are available in each regional center. On the other hand it allows asking experts questions about actual problems and programming hints. Experts can clarify misunderstandings and show examples on how they would solve similar problems. Mentors and experts have to take care not to give a ready solution to the students but to provide enough support to allow the students to solve the problem on their one. Figure 3 shows a scene at an open day.

The regional centers are also equipped with additional hardware, different sensors and various robot kits useful for a broad range of applications. The most important opportunity for students at the regional center is the testing in a real competition environment. For instance they usually encounter difficulties caused by the uncertainty of the environment. They discover that lighting conditions are different compared to their classroom and that they have to take care about it, e.g. calibrate sensors. These are very valuable lectures for them and improves the knowledge of the students and the quality of their solutions.

V. Results

Our attempts to establish a vital stable RoboCupJunior community in Austria have impact in at least three areas: (1) the RoboCupJunior community itself, (2) attractiveness of science and technology as a career opportunity and (3) the public awareness and understanding of education and research issues.

The impact can be evaluated in a qualitative and a quantitative manner. We collected the numbers concerning RoboCupJunior in Austria for the years 2007 to 2010 so far. Moreover, we did selective surveys and interviews during our support activities and the national Austrian competitions. Unfortunately, a systematic detailed overall evaluation has to be done in the future to evaluate the impact of particular support actions more precisely. Moreover, we present only preliminary results here. In fact three years are too short to seriously evaluate the impact of an activity like RoboCupJunior to the society. In order to review the impact on the attractiveness of science and technology to young people or the impact on later careers long-term surveys are necessary.

First lets have a quantitative look on the development of the RoboCupJunior community in Austria. Table V depicts the development of the numbers of students and teams involved in RoboCupJunior for the years 2007 to 2010.

The school year in Austria usually runs from September to June or July of the next calendar year. Therefore, our RoboCupJunior activities are usually planned for this period of time and ends with the annual international RoboCup competition usually hold in June or July.

The school year 2006/2007 was the first year of Austria was active in the RoboCupJunior initiative. As 2006/2007 was the first year there existed only one regional center and no RoboCupJunior network was established in Austria. Moreover, the support concept was not yet developed. During this year we focused on one selected team of students which we trained throughout the year. Finally, we applied for the first starting slot at an international RoboCup competition for an Austrian RoboCupJunior team and supported the team financially and logistically in their participation in RoboCup 2007 in Atlanta, USA. For the students and their mentors the whole story including preparation, traveling to another continent and to participate in a large international competition was an invaluable experience and triggered nation-wide media coverage.

Encouraged by this results we develop the support activities and started to actively communicate RoboCupJunior and our concept to schools. These activities were immense successful and we faced the situation that our entire capacities were quickly bound by an increasing number of schools participating. In order to cope with this situation and to provide easy and quick support throughout whole Austria we developed the concept of regional centers which provided decentralized support.

In the school year 2007/2008 we extended our network to 4 regional centers and conducted the first national RoboCupJunior competition in Austria. In this first competition immediately a fantastic number of 36 teams with 139 students and 16 mentors participated. For the RoboCup 2008 Austria successfully applied for five starting slots. Although, five teams were qualified in the national competition to participate in the RoboCup 2008 only one Austrian team competed in Suzhou, China. The reason for this was that RoboCup 2008 was held only a few weeks before the Olympic Summer Games 2008 Beijing and mentors expected a number of troubles concerning traveling and visa. Therefore, four teams resigned from the participation. Nevertheless, RoboCupJunior in Austria gained increased publicity and quality.

The school year 2008/2009 was special as 2009 Graz was the host city of the international RoboCup competition. Therefore, the interest of the media, the researcher and the schools was very much focused on RoboCup and RoboCupJunior. Nevertheless, we increased the number of regional centers to 7 and stabilized our support activities and network. For the national competition 2008 held in Vienna we were able to welcome 81 teams with 290 students and 31 mentors. For the RoboCup 2009 Austria as a host country got the increased number of starting slot of 30. We qualified the best 30 teams in the national competition and 26 Austrian teams participated in RoboCup 2009.

After this year which meant a tremendous effort for all people involved in RoboCup and RoboCupJunior we concentrated on the stabilization of the initiative after this unique event. We mainly focused on the training of potential mentors and teachers and the broadening of the basis of schools involved. We kept stable the number of teams, students and mentors who participated in the 2010 national competition held in Villach. Moreover, eight Austrian teams are qualified and will compete in the RoboCup 2010 Singapore. Which is a very good number for a little country with the size of Austria?

From the increasing number of students, mentors and schools involved in RoboCupJunior in Austria it can be concluded that the concept of support activities and regional centers is successful and mostly meets the needs of participants.

In order to evaluate the impact of their participation in RoboCupJunior we did a survey for students and teachers at the national competition in 2008. According to the feedback the majority of the students said that after the participation in RoboCupJunior it is more likely for them to choose an education and profession in science and technology and the majority of the mentors said that RoboCupJunior enriched their teaching.

If one looks at the impact of the RoboCupJunior activities in Austria on a qualitative level at a first place the development of skills of the student is evidently. The students develop remarkable technical skills in construction, programming and electronics which are often beyond the knowledge of their teachers. Because of the fact that rules do change dramatically from competition to competition also a yearly considerable improvement of the skills of individuals is observable. Beside the technical skills participants develop also social skills like communication, presentation, team work, and conflict management. Finally, the participation stimulates the self-confidence of the students in particular girls or students with difficult economic or family background.

Moreover, the RoboCupJunior activities also increased the public awareness of the problem of the lacking interest in science and technology of young people. This puts the initiative in Austria at a place to be able to discuss these issues with decision makers from companies, school authorities and politics.

Finally, in Germany the first students who participated in RoboCupJunior are close to finish their master theses related to

	2007	2008	2009	2010
# Regional Centers	1	4	6	7
Host City National RoboCup Competition	-	Graz	Vienna	Villach
# Teams @ Austrian National Competition	-	36	81	72
# Students @ Austrian National Competition	-	139	290	268
# Mentors @ Austrian National Competition	-	16	31	47
# Schools @ Austrian National Competition	-	15	27	30
Host City RoboCup	Atlanta, USA	Suzhou, China	Graz, Austria	Singapore
# Austrian Teams @ International RoboCup	1	1	26	8
# Austrian Students @ International RoboCup	4	4	110	35
# Austrian Mentors @ Intention RoboCup	2	1	56	9

TABLE I
DEVELOPMENT OF THE NUMBERS OF AUSTRIAN TEAMS AND STUDENTS INVOLVED IN ROBOCUPJUNIOR.

the RoboCup major leagues and have opportunities to continue towards a PhD. This fact clearly shows that RoboCup and RoboCupJunior provide a long-term perspective for students. In Austria the first students involved in RoboCupJunior just started their studies in engineering and computer science. We will observe the development of their future career.

Last but not least we established a valuable network to our neighboring countries in particular Hungary, Slovakia, Slovenia and Switzerland and exchange ideas. Moreover, we supported South Africa to start their local RoboCupJunior initiative which leaded to the participation of the first African junior team in the international RoboCup in 2010.

VI. CONCLUSION AND FUTURE WORK

The lack of interest in science and technology of young people has been a major challenge for western industrialized countries. Hand-on experiments has shown a very positive impact on this issue. In particular education robotics is used by several initiatives.

In this paper we presented the world-wide RoboCupJunior initiative which use robotics competitions to inspire young people and motivated why we adapted the RoboCupJunior initiative for Austria. We developed a set of support actions like introduction classes, open lab days and training for teachers in order to encourage and support schools to participate in RoboCupJunior. Moreover, we emphasized why a network of regional centers is necessary and increase the impact of the initiative.

Furthermore, we presented the results of the initiative in Austria achieved in the first three years. Mainly we were able to significantly increase the number of students, mentors and schools which actively participated in RoboCupJunior. Beside the quantitative numbers a qualitative increase of the skills of students and the public awareness for the issue has been achieved.

In the future we will work on a further improvement of the support of the mentors and teachers in particular increased training courses, improved teaching material for robotics and tools for networking and exchange of ideas and knowledge among mentors. Well motivated and trained mentors will act as multipliers and ensures a wide stable basis for the initiative. Furthermore, we plan to intense the promotion of more advanced solutions, e.g., high-level programming languages and self-made electronics, and more challenging competitions, e.g., soccer, which lacked in participants in the previous years.

REFERENCES

[1] Jose Mariano Gago and John Ziman and Paul Caro and Costas Constantinou and Graham Davies and Ilka Parchmannn and Miia Rannikmäe and Svein Sjoberg. Report by the High Level Group on Increasing Human Resources for Science and Technology in Europe. *European Commission*, 2004.

[2] Doug Oppliger. Using FIRST LEGO League to enhance engineering education and to increase the pool of future engineering students (work in progress). *Frontiers in Education, Annual*, 2:S4D11–15, 2002.

[3] Elizabeth Sklar, Amy Eguchi, and Jeffrey Johnson. RoboCupJunior: Learning with Educational Robotics. In Gal A. Kaminka, Pedro U. Lima, and Raúl Rojas, editors, *RoboCup*, volume 2752 of *Lecture Notes in Computer Science*, pages 238–253. Springer, 2002.

[4] Ubbo Visser and Hans-Dieter Burkard. RoboCup: 10 Years of Achivements and Future Challenges. *AI Magazine*, 28(2), 2007.

[5] Feng-Hsiung Hsu. *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2004.

[6] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. RoboCup: A Challenge Problem for AI. *AI Magazine*, 18(1):73–85, 1997.

[7] Henrik Lund and Luigi Pagliarini. Robot Soccer with LEGO Mindstorms. In *RoboCup-98: Robot Soccer World Cup II*, volume 1604 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 1998.

[8] Elisabeth Sklar and Amy Eguchi. RoboCupJunior - Four Years Later. In *RoboCup 2004: Robot Soccer World Cup VIII*, number 3276 in Lecture Notes in Computer Science. Springer, 2005.

Eurobot Junior and Starter – A Comparison of Two Approaches for Robotic Contest Organization

David Obdrzalek
Charles University in Prague, Faculty of Mathematics and Physics
Malostranske namesti 25, 118 00 Praha 1, Czech Republic
Email: david.obdrzalek@mff.cuni.cz

***Abstract*—In this paper we present our experiences gained from organizing a robotic contest for school childer as an adition to the "Eurobot" contest for robot builders of the age up to 30. In 2008 and 2009, we have used and compared two different approaches; based on this experience, we have set the model we believed is the most suitable for us. In 2010, it has proved this was a proper decision.**

***Index Terms*—Robot contest, Edutainment, Education, Entertainment robotics**

I. Introduction

Robotics is a quickly evolving area. Together with advances in research and manufacturing, popularization also takes place. Among other tools, it includes organizing contests for young students as well as for "mature" professionals. Over years, they became extremely popular. There are many contests all over the world, some of them need high expertise in robotics, some of them are targeted to beginner level of robotic science. But commonly, most robotic contests serve not only as a way how to select the best team but at the same time (and not rarely as a primary goal) as a great educational and scientific tool, because the participants learn a lot while building their robots and usually are able to find many new contacts with other participants in the contest.

The author of this paper has a long-time experience with organizing one robotic contest at national level, the "Eurobot Autonomous Robot Contest" (see [1] and [2]). His involvement started by participation with his students in Eurobot as one of the many robotic contests, and evolved into organizing this contest at Czech National level and working in the Executive Committee of Eurobot Association – the international body organizing this contest in 26 countries.

In this paper, we present the Eurobot contest and how we have extended it in the Czech Republic also for younger participants. Then we give our experiences with the two different models we used for 2008 and 2009 editions.

The following text is organized as follows: in Section II, we present Eurobot contest with basics of its rules and in Section III its implementation in Czech Republic. In Section IV, second contest (for younger participants) is introduced as Eurobot Junior and in Section V, we present another new category called Starter. In Section VI we give our experiences from the two years when Eurobot + Eurobot Junior (2008) and Eurobot + Starter (2009) contests have been organized in Prague, compare these two experiences, and briefly inspect the 2010 edition. Then we give our conclusion.

II. Eurobot Contest

In this Section, we give basic information about the Eurobot contest and its rules. For further information (details about rules, participation conditions, discussion and reasoning etc.) see the Eurobot Association web pages [1].

Eurobot is an international autonomous robot contest for young non-professional robot builders of the age up to 30 years. Their task is to build an autonomous robot which is able to perform actions defined by the rules. The contest sequencing uses a well known model with several qualification rounds followed by the finale matches played on knock-out basis. Every match lasts 90 seconds. During this time, two robots from two teams compete on a playing field.

The organizers change the rules every year; it is therefore easier for the participants to start with the competition (in contrast with other contests where the rules are fixed and it takes years to reach the top level) and it also stimulates the teams that already participated. After the organizers publish the rules at the end of September, the participants have about 7 months time to build their robots: during spring, national cups are held followed by an international finale in May/June. The participants work in teams, the usual size of a team is 2-6 people, but larger teams are not rare.

The core technical rules of the Eurobot contest are:

- autonomous robots
- indoor robots with limited size (roughly a cube with 30 cm edge size)
- game on a table (roughly 2 x 3 m)
- little time for one match (90 sec)
- fair-play spirit of the game

As mentioned above, every year the rules are renewed and new "topmost" topic is chosen. For example, in 2009, the actual rules specified:

- General idea: "Temples of Atlantis" - the robots are helping people to build wonderful temples, using building blocks found on the playing field
- Playing elements: the robots have to gather two types o playing elements: wooden cylinders (30mm high, diam-

Fig. 1. Eurobot Contest Finale in La Ferté-Bernard, France

eter 70mm) and wooden lintels (prism 200mm x 70mm x 30mm)

- Game goal: the robots have to build columns using the cylinders and lintels
- Scoring: basically, the taller structure built, the more points gained + bonuses for complexity (see also [1]).

The Eurobot contest has started as a French contest in 1993. In 1998, it has been officially widened and the Eurobot contest has been organized for the first time as an international event open to all countries (not only European). Number of participating countries increased and in 2004, Eurobot Association has been founded to organize such events in a long time frame. Since then, it is the Eurobot Association which every year prepares the rules, organizes national cups and the international finale. The contest grew from 14 French teams in 1994 to more than 300 teams from 26 countries in 2009 (including 13 national cups).

III. Eurobot Contest at Charles University, Prague

It has proven that the participation in such contest gives much more than just showing in praxis what has been learned in the class. To name just a few profits, the students have to work in a team, therefore they have to learn how to cooperate, their goal is to build an autonomous robot based on given rules so they have to find practical solutions for problems they have never met before, and last, but not least, they have to finish their work by an inalterable and inextedible deadline. Therefore, the contest has been integrated into the curriculum at Faculty of Mathematics and Physics at Charles University in Prague, Czech Republic, and is today well established and used as a practical project type.

For the first time, students of Charles University took part in the contest in 2001. Since 2004, the Czech Cup of Eurobot is organized at Charles University for all Czech teams. After running it for several years, the numerous questions from the public visiting the contest matches as spectators led the organizers to think about adding an activity also for younger people interested in robotics that want to take part but are afraid they cannot compete with university students.

In 2008 and 2009, the Czech National Organizer of Eurobot contest implemented this idea in Czech Republic as part of the Eurobot National Cup, both years in a different way. Later in this paper, we show our experience with the two models, one running a different contest for each age range (with completely different rules sets), and another one running two categories of a single contest (where the two age groups work separately but under the same core of rules with differences making the work feasible and challenging for the respective category).

IV. Trophées de Robotique / Eurobot Junior Contest

The Eurobot contest started in France as a single-category contest. While preparing the 2006 edition, the organizers had decided to address also younger fans of robotics and invite them under the roof of Eurobot Association. So, organizers of another French robotic contest, Trophées de Robotique, agreed to joined the Eurobot movement and while they prepared their contest in France, they also opened it for other countries under the name of Eurobot Junior (see [3] and [4]). Since 2006, this second contest for younger participants is organized every year.

This category is limited for participants of the age up to 18 years (or the end of their secondary school studies). The task for the junior contestants is simpler than in the main Eurobot contest; the most significant difference is that the robots are not autonomous but remote controlled. Both contests target students and recognize education as one of its goals. However, the rules are prepared by two independent groups and are totally different. Only the very basic ideas are shared (education, fair-play etc.). The technical part is different - even the robots are of roughly the same size as the playing field size is similar, the playing field decoration, playing elements, game goals etc. are not connected at all.

In 2008, Czech organizers of Eurobot National Cup prepared Eurobot Junior and run it in the Czech Republic too (see more in Section VI-A).

V. Starter Contest

For 2009, the Czech organizers decided to run the contest for the younger category like in 2008, however they knew it was not feasible to organize two completely different contests like the year before (explained later in this text). Therefore, they decided to use another model: to have the younger category much closer to Eurobot. They adapted Eurobot rules so that the task is easier for the younger participants while not cutting away more than necessary. The most significant difference lies only in the non-autonomy of the robot (and attributes related to this), the rest of the rules is left intact. Therefore, the two categories are very similar at sight, because the playing field is the same and most other visual attributes too, and the match goals for the robots are exactly the same. The exact rules for the contest in different years can be found at Eurobot Czech web pages (see [2]). Basically, the Starter rules consist of 8 points, of which 2 define the conditions for participation (age, team composition), 4 concern the on-stage working details and safety, and only 2 point define the contest as such. These two points say that the robot is remotely-controlled, and present a list of notes in respect to Eurobot rules (because of the non-autonomy, some technical details are not applicable – see point 8 in the following rules list).

For example, the 2009 Eurobot rules specification was a 44 pages long document (downloadable from [1]) and the 2009 Trophées de Robotique / Eurobot Junior rules were defined in 33 pages long document. On contrary, the complete 2009 Starter rules were:

> The rules for the Starter category are based on Eurobot rules with the following exceptions / additions:
> 1. The participants must be below 18 years old (or until the last year of the high school).
> 2. The team may be supervised by one older person (the teacher, parent, club leader etc.), however the robot must be completelly designed and created by the team members, not by the supervisor.
> 3. The robot will be operated by one team member - the pilot, who will use a control panel attached by electrical wire to the robot. The only control may be done using this panel and the wire, no other remote control system are allowed.
> 4. The control cable must be at least 5m long and will be hold in the air by a co-pilot. The co-pilot must not interact with the robot or its power sources, and the cable must not be used mechanically for robot control (for example to tow or rotate it). Should that happen, the team would be penalized.
> 5. The teams must pay attention to correctly choose the electrical cable with respect to the power it transfers. We also strongly recommend to protect the electrical circuitry by fuses located close to the power source.
> 6. There will be one electrical socket (220V) for each team close to the playing field, which may be used to obtain the electricity needed to power the robot using an appropriate power adapter. The teams may also use the batteries, but we would like to note that only hermetic batteries are allowed (i.e. no ordinary car batteries).
> 7. No part of the control panel, the wire and the robot may contain freely accessible electricity (contacts, live wires). Especialy, the power source (e.g. adapter or battery) clips cannot be bare and must be covered.
> 8. As the robot is controlled by the pilot, it does not have to comply to the 4th paragraph of chapter 4.1 ("The robot is a fully autonomous machine. It shall carry its own power source, actuators and control system.") and does not have to have the Starting cord (chapter 4.5.1). The systems defined in chapters 4.5.3 (Automatic shut down) a 4.5.4 (Obstacle avoidance system) are not compulsory, but the pilot must drive the robot accordingly. Missing Robot localization beacon support (chap. 4.5.5) would cause disqualification only in case the opponent proves it is vital for correct work of their robot (however, we recomment the teams to build such beacon support, as it would allow playing friendly matches with autonomous robots). The technical poster (chap. 4.5.6) is compulsory, but it may be bilingual (team's native language + English). The English part might be briefer, but it has to provide sufficient information about the team and its robot.
> The rest of the Eurobot rules remains valid; in case of doubts, the referee's decision is final.

It can also be seen that these rules are in fact year-independent, which also makes contest preparation fairly easy in comparison to creating a new rules set every year from scratch.

VI. Experience

The Czech National organizers of Eurobot decided to extend their activities and open a contest also for younger participants. In 2008, they organized in addition to the Eurobot contest also the Eurobot Junior contest. Based on this experience, they changed the model and organized in 2009 the Eurobot and Starter contests instead.

A. Eurobot Junior in Czech Republic, 2008

In 2008, the matches for the two contests, Eurobot and Eurobot Junior, were held on the same day at the same place. From the visitors' point of view, the two contests were interesting and the whole event was "diverse and colourful". The 2008 theme for Eurobot Junior was "Nature's Forces", Eurobot contest theme was called "Mission to Mars". The playing fields were of the same size, but that is the only common attribute; their colours, playing elements and playfield parts were different, for example:

Fig. 2. Eurobot Junior (left) and Eurobot (right) contests in 2008, Prague, Czech Republic

- Eurobot Junior: playing field: “peppermint green” with red and yellow starting areas and black deposit areas at short sides; playing elements: blue table tennis balls, red and yellow control elements. Playing field sloped by 10% towards the public.
- Eurobot: playing field: “grey yellowish” with red and blue starting areas and the deposit area on the long side; playing elements: blue, red and white floorball balls. Flat playing field.

See also the Eurobot Association website [1] for both detailed rules and Figure 2 for a view of the scene.

For the organizers, adding Eurobot Junior to their activities meant organizing two completely different contests even some external attributes were the same. The implementation required to translate two sets of rules (the official rules are published in English and needed to be translated to Czech), build two different playing fields, train two groups of referees etc. All this in fact doubled the expenditures and troubles too – for example, the contest is organized on volunteer basis so it was needed to find more volunteers to help, which is not an easy task.

After all, during the debriefing the organizers had to confess they underestimated how different the two contests in fact were to organize. They decided to keep the idea to involve younger participants but to change the model.

B. Starter in Czech Republic, 2009

After announcing the Starter rules for 2009, the organizers immediately received grateful comments from the future participants: The idea of having the same goals as older and more experienced participants was challenging for them.

The number of participants also increased significantly: in 2008, there were 7 teams registered for Eurobot Junior in Czech Republic (5 of them from one particular school). In 2009, 6 teams from 2008 registered again plus 5 more new teams, all from different places.

As an example of a nice cooperation between Eurobot and Starter participants, one group of robot fans participated in both contests. Part of them (younger members) has built the mechanical parts and drive, and the rest of the group members (members over 18) has built the autonomous control. This control module was attachable to the human-driven robot made by the first subgroup and therefore making it autonomous and able to participate in Eurobot contest. However, it was necessary for the two subgroups to strongly cooperate for obvious reasons - without proper support from the hardware part, the autonomous module could not have been used.

Since 2006, “Eurobot Teams’ Workshop” is organized after the contest for participating teams. Workshop attendees present their projects and discuss the solutions which they used, considered and even rejected. The workshop is organized the day after the contest, to avoid contest stress and rush. The workshop is optional, but the organizers invite all teams to take part and share their knowledge. Unlike in 2008, when no Eurobot Junior team took part (while all were invited), several Starter teams attended this workshop in 2009.

C. Comparison of 2008 and 2009 models

In 2008, the participants groups were disjoint - Eurobot participants inspected other Eurobot teams, Eurobot Junior teams talked to Eurobot Junior teams. The two contests were held on the same day at the same place, but they did not mix. But during the matches and preparations for them in 2009, the situation was totally different. Even the teams’ pits were separated, it was clearly seen that participants from the two categories, Eurobot and Starter, merged together and examined in detail the results of work of the other teams without any distinction to their category. The main cause lies in the same goals the two categories had and therefore the two groups of participants had the same knowledge of what to expect and what to look for. Also, as mentioned in previous section, Starter teams partipated in the Eurobot Teams’ Workshop too.

The organizers consider it as a great result, fulfilling one of the goals of Eurobot Association – to foster scientific and educational exchange between young amateur robotic fans.

The second model was also much easier to organize – in fact nothing different to a situation if the total number of Eurobot teams increases.

D. The contests in 2010

After the 2008 and 2009 editions, it was very clear that in Czech Republic, the 2010 edition will follow the 2009 model, i.e. to continue with Eurobot + Starter and keep the two contest categories as close as possible. There was no significant drawback noticed or experienced in 2009 and so the rules for Starter 2010 were defined the same way as in 2009. In fact, the short list used for 2009 (see Section V) did not need any change at all and was directly copied and used for 2010.

Number of Starter teams increased: there were 11 teams registered for Czech Cup in 2009 and 17 in 2010. Two Starter teams closely worked together with an Eurobot team (these pairs came from the same home structures), one team participated in both categories (mentioned in Section VI-B). About half of the robots was designed and constructed in a way that would allow consecutive development into a fully autonomous robot.

In 2010, most of the Starter teams participated on the Eurobot Teams' Workshop, making it trully integrated part of the event.

VII. Conclusion

In this paper, we have described the evolution of the Eurobot autonomous robot contest in Czech Republic in the last two years. The experience clearly shows several conclusions:

Firstly, the idea to extend the contest also to involve younger participants was good and eligible.

Secondly, adding the younger category did not only allow younger people to take part and learn from the experienced contestants in the main category, but also has shown that older participants can (and do) gain from the youngsters. Originally, the category for younger participants was introduced because the younger scholars were afraid of competing with university students. It has shown that in many cases this fear is unjustified and even more, in some cases the older students from universities can learn from the young scholars coming from secondary (and maybe even primary) schools.

At third, the organizers of Eurobot contest in Czech Republic have learnt that for small- and middle-sized national cups, the contest model used in 2009 serves better than the model used in 2008 when the two age categories were separated. It might not be the case for the very big national cups where tens or hundreds of teams participate in both categories, because it is not possible for them to organize it at the same time on the same place anyway. But for all other national organizing groups where the contests are held together, the advantages are clear and expressive.

Acknowledgement

This work was partly supported by grants GAČR 202/10/0761 and GAUK 76109.

References

[1] Eurobot Association: Eurobot Autonomous robot contest: http://www.eurobot.org

[2] Eurobot Association: Eurobot Czech Cup: http://www.eurobot.cz

[3] Eurobot Association: Eurobot Junior: http://www.eurobot.org/eng/junior.php

[4] Planete Sciences: Trophées de Robotique: http://www.planete-sciences.org/robot

Robotour - robotika.cz outdoor delivery challenge

Jiří Iša
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25, 118 00 Praha 1
Czech Republic
Email: isa@ktiml.mff.cuni.cz

Martin Dlouhý
MapFactor s.r.o.
Štefánikova 24, 150 00 Praha 5
Czech Republic
Email: md@robotika.cz

***Abstract*—In this paper, we present an international contest for autonomous robots: Robotour – robotika.cz outdoor delivery challenge. The main task is a navigation in real-world situations. First three years were held in park Stromovka, Prague, Czech Republic and raised an interest of many teams, media and general public. Last year, the contest started to migrate. To our knowledge, there is no similar European outdoor contest for fully autonomous machines. Note, that there are some common features with American Mini Grand Challenge and a younger Japanese Real World Robot Challenge. The rules of Robotour are described in more detail together with experience gained over the past four years – both from the organizers' and the participants' point of view.**

Keywords: autonomous robots, outdoor, international competition

I. Introduction

Competitions such as Eurobot [1] and DARPA Grand Challenge [2] have repeatedly shown that both young students and senior researchers are attracted by competitive research environments. Autonomous robotics is a multidisciplinary domain which offers educational opportunities and interesting real-world research topics.

In 2004, the American Defense Advanced Research Projects Agency (DARPA) organized the first Grand Challenge. The goal of DARPA was to foster a research in fully autonomous vehicles. In the first year, only 11.78 km of the 240 km long route were completed by the best team. Already in the second year of the competition (2005), five vehicles finished the 212 km long route. This shows a tremendous impact the challenge has had on the field of fully autonomous ground vehicles.

Since 1994, the Eurobot competition attracts many young people (more than 2000 in year 2010) [3]. Eurobot has successfully shown how an international competition can be used to teach young people how to cooperate and how to develop complex systems.

In 2006, the Robotour – robotika.cz outdoor delivery challenge has been founded. In our opinion, the large gap in complexity between Eurobot-like competitions (e.g. RobotChallenge [4], Istrobot [5] and other) and competitions like DARPA Grand Challenge needed to be bridged. In about the same time, other organizers felt similar insufficiency and more competitions were born. Since 2003, Field Robot Event focuses on the agricultural automation [6]. Since 2006, European Land Robotic Trial allows research teams and industrial companies to demonstrate their unmanned outdoor systems in realistic scenarios and terrains [7]. One year after Robotour – in 2007 – Tsukuba Real World Robot Challenge (RWRC) took place in Japan for the first time [8]. Since 2009, a similar straight line outdoor challenge takes place in Písek, Czech Republic [9].

Robotour – robotika.cz outdoor challenge is focused on autonomous ground vehicles and their orientation in the real-world outdoor environment. The robots perform a delivery task in complex environments of city parks. They are not allowed to leave paved roads. Participants of various background are welcome. In the previous years, students from high schools, university researches and hobbyists took part.

In this paper, we describe the Robotour – robotika.cz outdoor delivery challenge. General rules are covered in Section II. In Section III, we share experience obtained from the organizers' point of view. Reflections of the participants are captured in Section IV.

II. Rules

A. Historical Overview

The rules for each year change slightly and the contest becomes more and more challenging every year. The unified theme of all years is robot's ability to autonomously navigate in outdoor environments and to move payload from one place to another. The robots have to be fully autonomous, which means that after a task entry they have to control themselves.

Since the first year, the basic requirement is to navigate on paved roads in the park without leaving them – similar to cars not leaving the streets. In the second year, a possibility of robot cooperation was introduced. In the third year, obstacles were added and robots had to deal with them successfully. In the fourth year, robots did not know exactly their start position and had to deal with obstacles more carefully.

The fifth year of this contest should be a next step towards smarter and more autonomous robots. In contrast to the previous years, the robots get only a map and coordinates of the destination. The robots should be able to navigate around the park even if they have never been there before. The map and the destination should be the only information the robots get before the start. Robot successfully solving this task should be able to demonstrate its ability with a corresponding map in any park.

Fig. 1. A simple map of the Lužánky park in Brno given to the participants in 2009.

B. Detailed Rules

1) Task: The task for the robots is to deliver payload in a given limit of 30 minutes to a destination as far as 1 km. Robots must be fully autonomous, not leave a road and choose correct path on junctions. The starting place, starting time and the destination will be the same for all the robots.

2) Map: Vector map of footpaths in a park will be based on a vectorization of an ortophotomap and teams could improve it further. The basic idea is taken from Open Street Map [10]. A robot is allowed to use only this shared map – all other maps are prohibited!

3) Robots: A team can deploy multiple robots this year, but only a single designated one is used to compute a score. Every robot must have an emergency stop button, which stops its motion. The button must be easily accessible, red and must form a fixed part of the robot (aka Big Red Switch), so it could be used in a case of a danger. The team must show that it is easy to manipulate with the robot – two people must be able to carry it several tens of meters. There is also a minimal size – robot has to carry 5l beer barrel (at least an empty one).

4) Leaving the Road: The robots are expected to stay "on the road" which means to stay on the paved passage ways. If any robot leaves the road, its trial ends. The team has to take care of their robot and remove it immediately.

5) Obstacles: There could be obstacles on the road. Besides natural obstacles like benches there could also be artificial obstacles. A typical (artificial) obstacle is for example a figurant, a banana paper box or another robot. Robots must not touch an obstacle. Contact with an obstacle means an end of a trial. The robot may stop in front of an obstacle and visually or acoustically give a notice. Note, that the robot has to detect, that the obstacle is no longer present.

6) Robots Interaction: Situations, in which a faster robot catches up with a slower one, will not be explicitly handled. The faster robot can handle the slower robot as an obstacle, i.e. avoid it or wait until the "obstacle" disappears. In general, the road rules will be respected: right of way, avoidance to the right, passing on the left.

7) Start: All robots will start from the same park road simultaneously. A minimum width of this road is 3 meters. The starting area for each team will measure approx. 1.5×1.5 meters. Starting areas will follow one after another on one side of the road. Within the starting area, each team can place its robot as they see fit. The order of the robots on the start is given by their results in the previous round (a better robot will be closer to the destination). The order in the first round will be given by the order of successful homologation. Robots start automatically via their internal timers. During the last minute before the start, no interaction with the robot is allowed.

8) Score: The team, whose robot manages to proceed along the route best, wins. The aerial distance of the last position of the robot (leaving the road, a collision or a timeout) to the destination is critical. For every meter towards the destination, a team gets one point. If the team carries a payload, its score is doubled ("points for the payload"). Each robot can carry only one "payload". A 5l beer barrel (full) serves as a payload. In every round, a robot can obtain points at most equal to twice the aerial distance of the start and the destination.

9) Organization: The contest will consist of four trials for each team. The start and destination will be different for every trial. The selected destination will be announced to all teams 10 minutes before the start. The speed of the robots is not important (actually, it is limited to 2.5 m/s). All points gained during all trials will be summed together. The trial starts at a specified time and ends after 30 minutes. The robot must leave the starting area within 10 minutes of the start. If the robot does not move for 60 seconds its trial ends. Each team has to arrange for one person familiar with the rules that will be part of the referee team during the competition.

10) Homologation: A team can participate in the contest only if it is able to score at least one point. Another necessary condition is an ability to travel along a 10 meters long route fragment without a collision with any obstacle. The starting procedure will be tested (the automatic start) as well as the functionality of the emergency stop button. Usage of liquids, corrosive or pyrotechnic material as well as live beings is strictly prohibited. Every robot has to be accompanied by a team member, older than 18 years, who is fully responsible for the behavior of the robot.

11) Technical Documentation: Every team has to provide basic technical documentation about their robot (for presentations, general public and journalists). Three winning teams will be asked for a more detailed description for a website presentation and to make the entry of novices in the following years easier.

III. Organization

Robotour is organized as a three-day event (Friday to Sunday). Friday is dedicated to the testing, clarification of rule details and homologation. During the homologation, we want to make sure that robots are not dangerous, have a functional emergency stop button and are able to gain at least one point in the contest. Saturday is the contest day. Finally, there is a workshop on Sunday. It is after the contest, so the competitors have a fresh experience with their robots and algorithms. They are also not stressed any more and thus this is a good moment for sharing knowledge.

We started to enforce this three-day template after the first competition in 2006. That competition ended on Saturday and most teams left without letting us and other teams know what has worked and what has not. What was even more important was that teams left exhausted from the programming marathon and one team had a car accident on the way home. Since the following year, the workshop is mandatory.

The Robotour contest is relatively self-supporting and the expenses are minimal. There is no special playground – a public park is used instead. There is no need for renting a hall because the event takes place outside. To be precise, some room is necessary as a base for the teams especially in bad weather conditions. It is recommended to have a partner who provides this place, like Planetarium Praha in the first park Stromovka did. A good idea is also a combination with an exhibition of robots and a related technology parallel to the contest.

There is no registration fee, but the teams have to take care of catering and pay an accommodation.[1] Small items remain on the bill: leaflets printing, diplomas, cup for the winners, and a Saturday night dinner. The dinner is usually sponsored and the goal is to unite the teams and give them a chance to relax a little bit after the contest. Note, that prices are rather symbolic, which lowers expenses on one side and also reduces a potential rivalry between the teams.

A. Duties over the Year

The first task of the organizers is a precise specification of rules for the next contest. They are presented on the robotika.cz website in Czech and English languages. The core remains the same (autonomously navigate in a park) and the changes are usually a consequence of a discussion at the workshop and experience gained.

The second task is to ensure an affordable accommodation for a relatively large group of people (50 people needed accommodation in 2009). An agreement with a university dormitory serves well. The reservation must be performed usually a month in advance and that defines a clear deadline for the registration of the teams.

Finally, it is necessary to find an interesting park, manage permission for the contest day and find building with large enough room(s) for team base with many electric outlets.

B. Experience of the Organizers

There were couple lectures we have learnt over the last four years organizing Robotour (and previously several years of organizing Czech Cup of Eurobot). The basic scenario was already mentioned and serves good and is worth a recommendation. What has changed over the years are two major trends: the number of teams is increasing and the task is getting more difficult. In the first case, we tried to find some optimal timetable of the rounds and we are still not satisfied. What suits the teams does not suit a general audience and vice versa. This year, we will start all the robots from one place simultaneously, which could be attractive for spectators, but may cause problems to many teams.

The task complexity is another issue. Beginners have a harder position to enter the contest every year. For 2010, we discussed a new category (WagonOpen), but we will probably cancel it. The reason is a new, for the beginners with outdoor robots highly recommended contest “Robotem rovně” (Robot, go straight!) in Písek. In Písek, the task is to navigate as far as possible on a 3 meters wide and 300 meters long park road. This is exactly the first stage which is necessary to enter the Robotour contest.

[1] Accommodation is usually partially or fully sponsored.

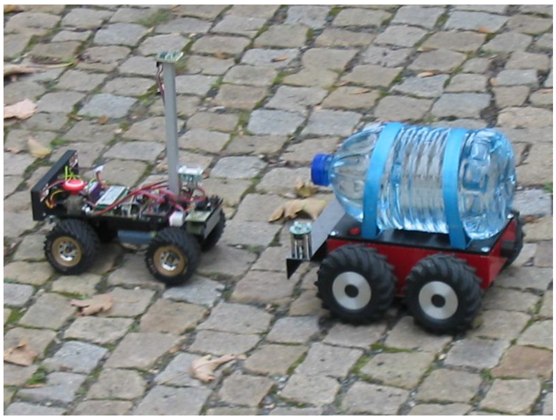


Fig. 2. Robot of the R-team (left) leading the allied robot of RobSys (right).

IV. Reflections

A. Questions

To reflect an influence the competition has had on its participants, we have asked some of the past successful teams few questions:

1) What did you expect from the competition?
2) What did the competition give you?
3) What were you disappointed with?

B. Asked Teams

The following teams were asked:

- **Propeler-team**, Opava: A group of high school students, who placed 2nd in 2006.
- **LEE**, Prague: Researchers and students from Czech Technical University in Prague. Winners of the year 2008 and the year 2009.
- **R-team**, Rychnov nad Kněžnou: A team of a high school teacher. Since 2010, he organizes *RobotOrienteering* in Rychnov nad Kněžnou. R-team finished 2nd in 2008 (in a coalition with the RobSys team, see Figure 2).
- **Roboauto**, Brno: A self-funded group of researchers, which ranked 2nd in 2009.

- **Radioklub Písek**, Písek: Hobbyists and professionals, who also teach electronics in a club. Radioklub Písek got a 3rd place in 2009. Since 2009, the club organizes *Robotem rovně* (mentioned in Section III).

C. Answers

1) What did you expect from the competition?:

- Propeler-team:
  - The competition motivated us to build our first robot.
  - Having almost no restriction on the dimensions of the robot allowed for a simple construction – We could use a notebook, get an image from a camera and use a bought chip to control the motor and the servo (we did not understand microchips and servos at that time).
- LEE:
  - We wanted to see a comparison of several approaches to the mobile robotics.
  - The competition gives us an opportunity to have our solution judged in an unbiased fashion.
- R-team:
  - After Istrobot and Eurobot, I wanted to try something new.
- Roboauto:
  - The competition served as a motivation to finish a functional version of algorithms and of the robot.
  - We wanted to present our results to a general public.
  - We expected to meet with a like-minded community.
- Radioklub Písek:
  - After seeing the robots in 2007, we believed we could do better.

2) What did the competition give you?:

- Propeler-team:
  - We met people in the same domain of interest, saw their approach and other technology.
  - Every year, we have a motivation to catch up with our first result.
- LEE:
  - We have seen, how a relatively simple solution (by R-team) can solve a given task.
  - We realized that the increasing accuracy of hardware and sensors can have a huge impact on the accuracy of simultaneous localization and mapping.
  - We have been shown, how important it is to deal with the technical details and with the reliability of the robots.
- R-team:
  - I have learned that even the hardware is not fully reliable. Indoor robots do not suffer from such problems.
  - I realized how difficult the task is, even though I have expected some difficulties even beforehand.
- Roboauto:
  - It has fulfilled our expectation.
  - The competition gave us a practical experience with deploying a robot.
  - We have got an inspiration for further improvements of the hardware and algorithms.
  - We feel in touch with people with similar interests.
- Radioklub Písek:
  - We realized the competition is not as simple as it seemed for the first look and few others.

3) What were you disappointed with?:

- Propeler-team:
  - We are not really disappointed: When the robot works, everything is fine.
  - Answering the question "What does the robot do?" is difficult, when the task difficulty is not obvious.
- LEE:
  - Although there is a lot written by the competitors at robotika.cz, every year someone new comes and repeats previous mistakes.
- R-team:
  - In my opinion, the competition has become too difficult. Only one or two best teams can fully cope with the rules.
- Roboauto:
  - Problems with a reliability and with a robustness are bigger than we have expected.
  - We are disappointed with only a small media attention.
  - We hoped to get an attention of potential sponsors or future team members, which has not happened so far.
- Radioklub Písek:
  - We are sad that the cooperation of multiple robots is not encouraged any more. We have learned several interesting things doing that. On the other hand, as the competition evolves, it does not suffice to copy a solution from the previous year.

V. Summary

We have introduced Robotour – robotika.cz outdoor delivery challenge, its rules and their evolution over the time. We share experience gained while organizing several years of the competition and show several patterns worth following. The competition has been successful in attracting people to robotics and giving them an opportunity to learn. The contestants enjoy a chance to meet others, exchange ideas and compare their approaches in an independent manner. As the competitors note, while seemingly simple, the competition became difficult to participate in. This in turn led to a creation of two new robotic competitions in Czech Republic, which differ in the level of difficulty. Currently, there exists an evolutionary path for a person interested in robotics through these outdoor competitions up to Robotour and possibly even further.

REFERENCES

[1] Eurobot, "Contest for amateur robotics," http://eurobot.org/, 1997.
[2] DARPA, "Grand Challenge," http://www.darpa.mil/grandchallenge/, 2004.
[3] David Obdržálek, "Eurobot 2010," *Robot Revue*, May 2010.
[4] Robot Challenge, "Indoor robotic competitions," http://robotchallenge.org, 2010.
[5] Istrobot, "Contest for amateur robotics," http://robotika.sk/, 2010.
[6] Field Robot Event, "Agricultural robotic competition," http://fieldrobotevent.de, 2010.
[7] ELROB, "1st European Land-Robot Trial," http://www.elrob2006.org/, 2006.
[8] Real World Robot Challenge, "Outdoor robotic challenge," http://www.ntf.or.jp/challenge/, 2009.
[9] Robotem Rovně, "Outdoor robotic challenge," http://www.kufr.cz/view.php?nazevclanku=robotem-rovne-2010&cisloclanku=2010020002, 2010.
[10] Open Street Map, "Public mapping activity," http://www.openstreetmap.org, 2010.

A Visual Navigation System for RoboTour Competition

Tomáš Krajník, Jan Faigl, Vojtěch Vonásek, Hana Szücsová, Ondřej Fišer, Libor Přeučil
Czech Technical University in Prague, FEE, Department of Cybernetics, Gerstner Laboratory
{tkrajnik,xfaigl,vonasek,szucsova,fisero,preucil}@labe.felk.cvut.cz

***Abstract*—In this paper, we present our approach to the navigation system for the RoboTour challenge. We describe our intentions, ideas and main principles of our navigation methods which lead to the system that won in years 2008 and 2009. The main idea of our system is a simple yet novel method of position estimation based on monocular vision and odometry. Unlike in other systems, the monocular vision is used to determine only the robot's heading and the odometry is used to estimate only the traveled distance. We show that the heading estimation itself can suppress odometric cumulative errors and prove this statement mathematically and experimentally. The practical results of the proof is that even simple algorithms capable to estimate just the heading can be used as a base for "record and replay" techniques. Beside the navigational principles, practical implementation of our navigation system is described. It is based on image processing algorithms for path following and landmark-based crossings traversing. Moreover, an overview of experimental results is presented as well.**

***Index Terms*—visual navigation, robotic competitions**

I. Introduction

The RoboTour contest is aimed to outdoor autonomous robots capable of delivering a payload to a given destination. Although the competition rules have evolved over time, the basic idea is the same: an autonomous robot should travel a given path on walkways in a park-like environment. The contest is therefore closely related to the safe mobile robot navigation in large outdoor environments. The first year of the contest has been organized in 2006 and we have participated in each year til now. During our participation, we used the contest as an evaluation scenario for our navigational algorithms as it is great opportunity to test reliability of the whole system of autonomous navigation. At the beginning, we have been in position that none of our navigation algorithm can be directly used in an outdoor environment. In that time, our group at the Gerstner Laboratory has been strictly focused to indoor navigation techniques[1]. So, from a certain point of view, we have been in similar position to other teams participating in the first year. However, we have an advantage of know-how in navigational principles and sensors equipment.

One of the most important RoboTour rules is that a robot, which leaves the pathway, will be penalized. Considering the basic definition of the delivery task, we realized, that the main principle of the navigation should be based on path following. Many teams tried to solve the task by using high-precision GPS receivers. However, the GPS signal in park-like environments suffers from reflections and occlusions and therefore, the GPS precision is around thirty meters, which is insufficient to keep the robot on the pathways. The GPS can be complemented by an odometry and a compass to estimate the robot position using Kalman filtering methods. Even through this sensor fusion can improve position estimation, it does not provide sufficient precision to keep the robot on the pathway. The odometric error tends to accumulate over time and therefore these methods do not perform better than sole GPS-based localization in the long term. Although the odometry is precise for travelled distance estimation, it cannot provide sufficiently good heading estimation, thus it is unsuitable for long-term robot localization.

Moreover none of the aforementioned sensors provide any information about the robot surrounding environment. A robot using these sensors is navigated by a simple control rule to the desired coordinates and ignores the situation in its vicinity. The reliability of its navigation is determined solely by precision of its GPS receiver. Regarding to this issue, the aforementioned sensors should be complemented by extroreceptors like digital cameras or rangefinders providing information about the robot surrounding environment. Having an intelligent robot that will be able to recognize pathways, crossings and obstacles from its sensors, the precise position estimation is not needed at all. These evidences were main ideas in our choice of research direction in reliable robust autonomous navigation for outdoor environment, which is to build an intelligent mobile robot (possibly from cheap off-the-shelf components) that will be capable of recognizing its surrounding environment and to select the most appropriate action to fulfill its goal.

To build such an intelligent mobile robot the "only" thing is to "process" its sensory data. The most common sensors in mobile robotics are cameras and laser rangefinders. While laser rangefinders are precise and robust to changing illumination, they are prohibitively expensive to most of the teams. Digital cameras are cheaper, but image recognition in outdoor environments is not a simple task, because of the complexity of the outdoor environments. In particular, the pathways differ in color, texture, width and often are interrupted by ruptures with vegetation. The recognition is complicated by slops, fallen leaves, dirt, shadows of surrounding trees, image blur caused by robot movement and variable illumination. However, the problem of path recognition has been successfully solved years

[1]Here, it should be noted that the Intelligent and Mobile Robotics Group (http://imr.felk.cvut.cz) has been founded in late nineties.

ago [1].

Once the path following is solved, a more challenging problem of reliable and robust crossing recognition arises. Many teams have successfully achieved implementation of a reliable path recognition, but still have problems with the crossings. This was also our case in the year 2006. Our robot, equipped with the “GeNav” [2] algorithm, was able to follow the pathways with sufficient reliability, but lacked the ability to choose the right paths on large crossings.

Based on our experience gained in the first RoboTour contest, we have proposed to use different image processing methods for the path and crossing traversal. The path traversing algorithm is based on path recognition, while crossing navigation is based on visual landmark recognition.

Because our intention is to build a reliable and robust navigation system, we have decided to use map-based navigation methods, at it is known that such methods are more reliable than mapless techniques. Moreover, we have decided to not follow mistakes made in seventies in the field of Artificial Intelligence (AI), i.e. to use knowledge representation that is natural to humans, but not to machines. Instead, we considered the new AI concepts and let the robot use knowledge representation that is natural to its sensory equipment and reasoning abilities. Therefore the map, which is used by our robot, is not easily interpretable by a human at a glance.

Our approach lead to a minimalistic monocular navigation system capable of navigation within a known environment. The robot utilizes a map just to correct its heading and measures its position by a relatively imprecise odometry. We claim, that the heading corrections effectively suppress the cumulative errors of odometry. We formulate a particular instance of the navigation method mathematically and provide a proof of the claim in [3]. The practical implementation of the method won the RoboTour challenges in years 2008 and 2009. Beside our RoboTour participation, we examined the method in several outdoor experiments that confirm the system performance. In this paper, we present the main ideas of our methods used in the RoboTour challenges and described practical issues that have been meet in realization of the system for the RoboTour.

The paper is organized as follows. The principles of the proposed navigation methods are described in the next section. A mathematical model of the navigation methods is outlined and its properties are examined in Section III. An overview of the main experimental results evaluating the system performance is presented in Section V. The conclusion discusses the proposed navigation method and outlines possible future improvements.

II. Principles of the Navigation Methods

This section provides an overview of the main navigation principles used in our system based on two navigation algorithms. We assume that the robot has a differential, nonholonomic drive and its movement can be describe by equations

$$\begin{array}{rcl} \dot{x} & = & v\cos(\varphi) \\ \dot{y} & = & v\sin(\varphi) \\ \dot{\varphi} & = & \omega \end{array} ,$$

where x, y denote the robot position, φ is its heading and v and ω denote forward and steering velocities. In general, the task of a navigation algorithm is a computation of the required velocities v and ω from the actual and past sensory data. To simplify the equations describing the robot movement based on sensory data, we assume the robot is capable of fast turns and the steering actuator sets the robot heading φ directly.

The first navigation algorithm is a simple path following algorithm, which is capable to keep the robot on the path. Reliable path following has been solved by most teams, but the crossing recognition remains a problem. In our opinion, the reason for it stands in the fact that most of the robots are small and their cameras are not very high above ground. Moreover, the camera is usually firmly fixed on the robot body and therefore, the robot cannot “look around” when following a path. For a small robot, a crossing is more or less an open space, and therefore the robot cannot distinguish between crossings and wide paths, see Fig. 1 for an example of crossing. Moreover, it sparsely spots all the exiting paths. Due to these facts, we have decided to use an algorithm based on objects recognition for crossing traversal. The advantage of the algorithm is that it does not rely on environment structure, so it can also be used in areas that are not divided to paths and obstacles. On the other side, a possible disadvantage can be in the higher computational cost of object recognition and complexity of the map. Even through these algorithms seem to be different, they use the same navigational principle. Let us review the main principles of the path following and landmark-based navigation techniques.

Figure 1: A crossing, which is difficult to survey.

a) Path Following: The path following algorithm can be fairly simple. A robot controlled by the algorithm has to move forwards and keep itself on the path. Suppose, that the robot knows how to recognize the borders of the path in the image from its camera. This can be achieved by several ways, ranging from a simple perceptron working in RGB space to elaborate methods combining several computer vision algorithms [1].

If the borders are identified, the robot can compute which border is closer and steer in the other direction. Therefore, the robot adjusts its heading to keep itself in the middle of the path. Now, suppose that the robot is moving along a straight path. A 2D coordinate system $x - y$ can be defined such that the x-axis is at the path center. Then, the heading φ of a robot following the path can be expressed as

$$\varphi \approx -K_0 y,$$

where K_0 is a positive constant.

b) Landmark-based Navigation: Suppose that the robot has a map of the environment containing descriptions and positions of salient objects. The map allows computation of objects that will be visible from a particular robot position. While the robot approaches a particular position, it retrieves objects from the map and matches them to the actual set of seen objects. Because the objects are provided with the image coordinates, the robot heading correction can be directly based on the horizontal coordinates. Assume the expected robot position and heading are zero, but its true position and heading is $(0, y, 0)$. The actually seen objects are not in the expected positions in the image, but they are shifted to the right, for $y > 0$ and vice versa. In this case, the robot can steer in a direction that will minimize the differences in lateral coordinates of the expected and detected objects. In other words, the robot heading φ satisfy similar constrain as in the previous case:

$$\varphi \approx -K_1 y.$$

Even though the algorithm realized the so-called visual compass, it can be used also for the path traversing.

c) Making Turns: Here, it should be noted that both algorithms suppose the robot heading is close to the right direction. If a robot following a path is suddenly turned by 180°, it will just continue in the wrong direction. Similarly, if a robot does not see expected objects, the landmark-based navigation fails to correct the robot heading. Therefore, none of these algorithms are suitable to make sharp turns. In such cases, the robot has to stop and turn to the requested direction using its compass data.

d) Switching the Methods: Having these three methods: path following, landmark-based navigation and turn making a robot should be able to traverse paths and crossings in five steps. The robot would (1) approach a crossing by path following, then (2) traverse to the crossing center by landmarks, (3) turn towards a particular crossing exit, (4) use landmarks to approach the exit and (5) finally switch to path following to traverse to the next crossing.

The final problem stands in the mechanism of algorithms switching. Once the robot miss the crossing center, it will unlikely find the appropriate exit. The crossing is difficult to detect due to insufficient field of view and resolution of common cameras for achieving an overview of the complete crossing. Also GPS-based localization methods are not precise enough in parks. Although traditional image-based localization is more precise than the GPS, it is not robust to weather and seasonal environment changes [4].

On the contrary, the distances a robot has to travel between individual crossings are well known and can be measured by the odometry, which provides sufficient precision as it is used only locally between the crossings. Therefore, the robot can utilize odometry to decide if it has arrived at a particular crossing. The intended path of the robot may be split into several segments with a different length. A plan to traverse a simple path with one crossing can be as follows: *Follow the path by 46 meters, then go 4 meters towards the red tree, turn by 90 degrees, go 7 meters towards the blue circular file and follow the path for 43 meters.* One might argue that the cumulative nature of the odometry error would cause this plan to fail, because both of the aforementioned navigation algorithms compensate only the lateral position error. In the next section, we explain that despite their simplicity, the algorithms can compensate the odometry error.

III. Navigation Stability

In this section, we show how the aforementioned navigational methods compensate the odometry error. At first, we introduce a model of the robot movement along the plan consisting from a sequence of path segments with various lengths. Based on the model, we explore navigation properties and outline a proof of the navigation stability. The main idea of the proof is based on a robot position uncertainty as it moves along a segment. A navigation method is stable if the uncertainty does not diverge, which holds for our navigational methods used in the RoboTour competitions.

A. A Model of Robot Movement

Assume the same situation as in the previous section, i.e. the robot moves along a path, which lies on the x axis of the 2D coordinate system. A robot position evolution as it moves along the path can be described by relations of $y(t)$ and $x(t)$ in dependence of the controller's action values. The robot forward speed controller maintains constant speed until the robot has traversed path longer or equal to the segment length, i.e. $v = v_k$. Let us assume, that the robot steering controller sets the robot heading. So, the robot is driven towards the segment axis, i.e. $\varphi = -ky$, where k is a positive nonzero constant. Assuming that the robot heading φ is small, we can state that $\dot{x} = v_k$ and $\dot{y} = v_k\varphi$. Solving these two differential equations with boundary conditions $x(0) = a_x, y(0) = a_y, t = s/v_k$ gives the robot position (b_x, b_y) after it completes the path:

$$\begin{aligned} b_x &= s + a_x \\ b_y &= a_y e^{-ks}. \end{aligned} \tag{1}$$

Equation (1) would hold for an errorless odometry and noiseless camera. Considering the odometry and camera noises, the equation can be rewritten to

$$\begin{pmatrix} b_x \\ b_y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{-ks} \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix} + s \begin{pmatrix} 1+\upsilon \\ 0 \end{pmatrix}, \tag{2}$$

where υ is a random variable drawn from the Gaussian distribution with the zero mean and the variance ϵ and ξ is a random variable of the Gaussian distribution with the zero mean and the variance τ. A compact form of (2) is

$$\mathbf{b} = \mathbf{M}\mathbf{a} + \mathbf{s}. \tag{3}$$

To apply (3) for an arbitrarily orientated segment, the coordinate system can be rotated by the matrix $\mathbf{R}$ and then back by $\mathbf{R^T}$. Thus, (3) can be rewritten as follows

$$\mathbf{b} = \mathbf{R^T M R a} + \mathbf{R^T s} = \mathbf{N a} + \mathbf{R^T s}. \tag{4}$$

Using (4), the robot position at the end of the segment can be computed from its starting position. However, the absolute position does not concern us, to show the navigation stability we need to describe and predict the robot position uncertainty.

B. Position Uncertainty

Let the robot position $\mathbf{a}$ be a random variable drawn from a two-dimensional normal distribution with the mean $\hat{\mathbf{a}}$ and the covariance matrix $\mathbf{A}$. Equation (4) has only linear and absolute terms, and therefore at the segment end the position $\mathbf{b}$ will constitute a normal distribution with the mean $\hat{\mathbf{b}}$ and the covariance matrix $\mathbf{B}$. Denoting $\mathbf{a} = \hat{\mathbf{a}} + \tilde{\mathbf{a}}$, where $\tilde{\mathbf{a}}$ is a random variable of a normal distribution with the zero mean and the covariance $\mathbf{A}$ and similarly $\mathbf{b} = \hat{\mathbf{b}} + \tilde{\mathbf{b}}$, equation (4) can be rewritten to

$$\tilde{\mathbf{b}} = \mathbf{N}\tilde{\mathbf{a}} + \mathbf{R^T}\tilde{\mathbf{s}}.$$

Assuming $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{a}}$ are independent and do not correlate,

$$\tilde{\mathbf{b}}\tilde{\mathbf{b}}^\mathbf{T} = \mathbf{N}\tilde{\mathbf{a}}\tilde{\mathbf{a}}^\mathbf{T}\mathbf{N^T} + \mathbf{R^T}\tilde{\mathbf{s}}\tilde{\mathbf{s}}^\mathbf{T}\mathbf{R},$$

which rewritten in terms of covariance matrices is

$$\mathbf{B} = \mathbf{N A N^T} + \mathbf{R^T S R} = \mathbf{N A N^T} + \mathbf{T} \tag{5}$$

where $\mathbf{T} = \mathbf{R^T S R}$. Equation (5) allows determination of the robot position uncertainty after traversing one segment. A geometrical interpretation of the algebraic terms describing the uncertainty evolution is shown in Fig. 2.

C. Traversing multiple segments

Let the robot path is closed and consists of n chained segments denoted by $i \in \{0, \ldots, n-1\}$. A segment i is orientated in the direction α_i and its length is s_i. The robot positions at the start and end of the i^{th} segment are $\mathbf{a_i}$ and $\mathbf{b_i}$. The segments are joined, so $\mathbf{a_{i+1}} = \mathbf{b_i}$ and the movement model (5) for the i^{th} traveled segment is

$$\mathbf{A_{i+1}} = \mathbf{B_i} = \mathbf{N_i A_i N_i^T} + \mathbf{T_i}$$

The robot position uncertainty after traversing whole path consisting of the n segments will be

$$\mathbf{A_n} = \breve{\mathbf{N}}\mathbf{A_0}\breve{\mathbf{N}}^\mathbf{T} + \breve{\mathbf{T}},$$

where

$$\breve{\mathbf{N}} = \prod_{j=n-1}^{0} \mathbf{N_j}$$

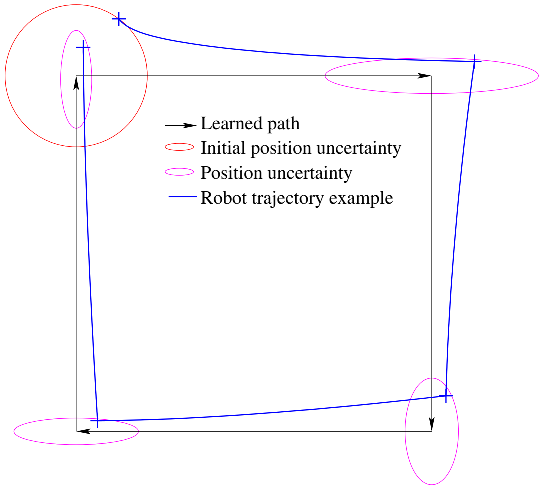


Figure 2: Position uncertainty evolution for a simple symmetric path.

and

$$\breve{\mathbf{T}} = \sum_{\mathbf{j=0}}^{\mathbf{n-1}} \left(\left(\prod_{\mathbf{k=n-1}}^{\mathbf{j}} \mathbf{N_k} \right) \mathbf{N_j^{-1} T_j} \left(\mathbf{N_j^T}\right)^{-1} \left(\prod_{\mathbf{k=j}}^{\mathbf{n-1}} \mathbf{N_k^T} \right) \right)$$

If the robot traverses the entire path k-times, its position uncertainty $\mathbf{A_{(k+1)n}}$ can be computed in a recursive way by

$$\mathbf{A_{(i+1)n}} = \breve{\mathbf{N}}\mathbf{A_{in}}\breve{\mathbf{N}}^\mathbf{T} + \breve{\mathbf{T}}. \tag{6}$$

Since (6) is Lyapunov discrete equation its limit for $i \to \infty$ is finite, because all eigenvalues of $\breve{\mathbf{N}}$ lie within a unit circle and $\breve{\mathbf{T}}$ is symmetric. It means, that if the robot travels the trajectory repeatably, its position uncertainty $\mathbf{A_i}$ will not diverge. A detailed analysis of $\breve{\mathbf{N}}$ and $\breve{\mathbf{T}}$ is beyond the scope of this paper and can be found in [3].

IV. The Navigation System

Our system runs on the P3AT robot equipped with the Unibrain Fire-i601c camera, the TCM2 compass and the HP 8710p laptop. The robot camera is aimed forwards and provides color images with resolution of 1024x768 pixels and field of view approximately 60°.

The system implements two navigation algorithms based on image processing that follow the principles described in Section II. The first algorithm, called “GeNav” [2], recognizes a pathway in front of the robot and corrects robot heading to keep it in the middle of the recognized path. Although quite simple, fast and reliable, its main disadvantage is its reliance on the environment structure. It can be used only in areas, where pathways are clearly distinguishable. The second algorithm is called “SUFNav” [3] and it is based on a salient feature recognition [5]. While robust and reliable, salient feature extraction requires a significant amount of processing power, therefore a GPU based future extraction algorithm is used [6]. Both algorithms require a prior knowledge about the environment. While GeNav needs a pathway color, SURFNav requires a detailed map of salient objects around the path being traversed. Therefore both algorithms require a suitable map of the operational environment. The map is created by manual driving of the robot by a human operator through the environment prior to the competition. Alternatively, a map created by another robot or publicly available maps like [7], [8] can be used.

The map consists of a set of straight line segments. Each segment is described by the initial robot orientation α, the segment length s, the landmark set L and the color table $\mathbf{G}$. The set L consists of salient features detected in images captured by the robot's forward looking camera. The color table $\mathbf{G}$ is a mapping of the RGB color space to N^+ indicating the likelihood of a pixel being on the path. Once the map is created, the robot can travel autonomously within the mapped environment using the algorithms. GeNav recognizes the path in front of the robot based on the color table $\mathbf{G}$. SurfNav computes steering by matching the currently detected landmarks to the landmark set L. Both algorithms decide only the robot steering, and the robot forward speed is set according to odometry measurements. A simple, sonar-based obstacle avoidance routine based on the Tangent Bug method [9] can temporarily override steering and forward speeds set by both GeNav and SURFNav algorithms in cases of detected obstacles in the robot course.

A. The Mapping Phase

The mapping procedure is initiated by a human operator. Before the robot starts to learn the segment, it reads compass data to establish the segment azimuth α and resets its odometric counters. After that, the robot starts to move forwards while measuring the traveled distance and processing the onboard camera image.

The onboard camera image is processed by two independent algorithms. Since its lower half contains the path, on which the robot moves, a small trapezoidal area at the bottom of the image is used to update the color table $\mathbf{G}$. The table $\mathbf{G}$ is implemented as a three dimensional array. Each cell of $\mathbf{G}$ represents a color in the RGB color space and contains an integer value. If a pixel of a particular color is detected in the trapezoidal area, the corresponding cell value is increased. At the end of the mapping phase, the $\mathbf{G}$ contains a color histogram of the pathway.

The upper half of the image is processed by the SURF [5] algorithm, which provides a set of point features. Each feature is described by its position in the image and a descriptor invariant to lighting and viewpoint changes. These features are tracked as the robot moves. If a feature is tracked long enough, it is saved in the landmark set L along with additional attributes. The attributes are the number of images, in which the landmark was detected, feature position in the image and the robot position as a distance from the current segment start, when the feature tracking was initiated and finished.

The feature tracking can be described in terms of manipulating three sets of the features: currently detected features S, tracked features T and saved features L. At first, features extracted from the current image are put to S. Then, similarity between the features in the sets T and S are computed based on the Euclidean distance of their descriptors. For each currently tracked feature $\mathbf{t_i} \in T$, two most similar features from S are found. If these two pairs are distinguishable [5] the best matching feature is removed from S and the tracked landmark description is updated. If the two pairs are not distinguishable, the landmark $\mathbf{t_i}$ is moved from the set T to the set L. After all the tracked landmarks have been updated or removed, the remaining features of S are moved to T. When the mapping is completed, each feature in the set L is described by its descriptor, position in the image and values of the robot odometric counter in moments of the first and last time the feature was tracked. The segment description consisting the azimuth α, length s and the set L is saved at the end of the segment and the operator can turn the robot to another direction and initiate mapping of a new segment.

B. Navigation Phase

In the autonomous navigation mode, the robot is supposed to traverse a sequence of mapped segments. The operator has to place the robot at the start of the first segment and indicate, which segments are to be traversed by which algorithm. Then, the robot loads description of the first segment, turns in the direction of the segment azimuth and starts to move forwards. Its forward speed is set according to the travelled distance until the odometric counter indicates that the segment length has been traversed. The robot steering speed is decided by either GeNav or SURFNav algorithms. After the segment is traversed, the robot loads description of the next segment and repeats the navigation procedure.

C. Pathway Recognition - GeNav

The bottom half of image is analyzed by GeNav. In particular, it recognizes a pathway in the image and determines the robot steering speed to keep it in the middle of the detected pathway. The algorithm uses the color table G to decide which pixels lies on the path and works as follows.

GeNav starts with the bottom row of the acquired image searching for pixels of the path color and determine the mean value of their horizontal coordinates. After that, the algorithm searches for the path boundaries. It starts from the mean position and searches for pixels with other than path color in both directions from the mean. The path center and width are then calculated out of the detected boundaries for the particular row, see Fig. 3. If the width is greater than a predefined threshold, the algorithm proceeds to a higher row. The search algorithm is completed when the current path width drops below the threshold or when it reaches the middle image row, see Fig. 3. The robot steering speed ω is determined from the mean of the computed path centers.

D. Landmark-based Navigation - SURFNav

SurfNav analyzes the top half of the onboard camera image. The robot uses the landmark set L and the currently detected features to determine the robot steering speed. A set of

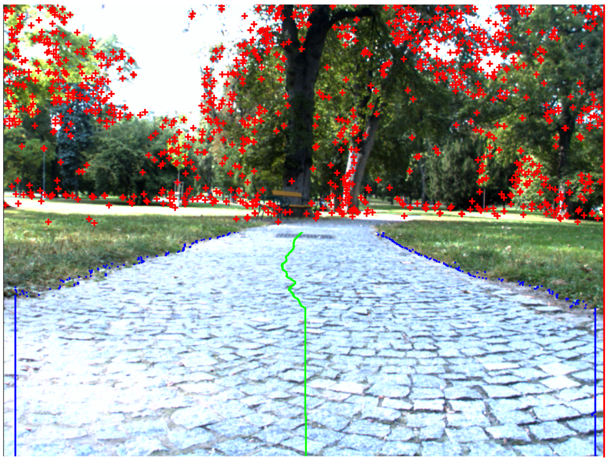


Figure 3: The image processed by both algorithms.

landmarks T, which are expected to be seen at the current position are selected from the set of the learned landmarks L. For each landmark in T, the best matching feature in the set of currently detected landmarks S are found in the same way as in the mapping phase. A difference in horizontal image coordinates of the features is computed for each such tuple. A modus of those differences is estimated by the histogram voting method. The modus is then used to compute the robot steering speed ω.

V. Experiments

The practical verification of our navigation system has been examined in a series of outdoor experiments with a P3AT mobile robotic platform.

The first set of experiments was aimed at verification of the navigation system and measuring its precision. The experiments have proved, that the proposed method is able to cope with diverse terrain, dynamic objects, obstacles, systematic errors, variable lighting conditions and seasonal environment changes. During these experiments, the robot has autonomously traversed over 25 km with an average position error lower than 0.3 m [3].

In the second set of experiments, we have tried to build the map from the Google Street view data. The parameters of the Google Street view images were set up to resemble the image a robot would see. After that, the SURF features have been extracted from these images and a landmark map was created. The experiment showed, that a mobile robot can use this landmark map for navigation in an urban environment. However, the SURFNav algorithm has to be complemented by collision avoidance.

Participating on the RoboTour competitions can be considered as an experiment as well. Contrary to the regular field tests, the competition is more challenging, because all system componets must work flawlessly. So, the RoboTour event is not only the navigational method examination, but a test of the whole system. Our navigation method evolved during the time when we started with it in 2006. In that year, we used only the path following algorithm and we had problems in crossing recognition. A year later, our system was complete, but contained a lot software bugs. The main milestone was made in 2008 and from that time we consider the navigational system complete, however we still improved it in particular aspects. In years 2008 and 2009, most of the software bugs were resolved. Event through the performance has not been perfect, the robot was able to travel the required trajectories and our team has reached the first rank for both events in 2008 and 2009.

VI. Conclusion

A simple navigation system based on bearing-only sensors and an odometry used in the RoboTour competitions was presented in this paper. The method navigates a robot using a map of the environment and a camera input to establish the robot heading, while the traveled distance is measured by the odometry. Using a mathematical model of the navigation method, we have shown that this kind of navigation is sufficient to keep the robot position error limited. Besides, the proposed method has been experimentally verified by a mobile robot with a monocular camera.

Even through our success in the last two RoboTour competition, the main disadvantage of our method is in the necessity of the detailed map in advance. The robot can create the map in a guided tour, however it takes a long time, because of its low speed. The mapping of the Stromovka and Lužánky parks took two, resp. four days. Our intention is to solve this issue while preserving the reliability of the method.

Acknowledgements

The work presented in this paper has been supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS10/185, by the Ministry of Education under by the project MSM 6840770038 and by the FP7-ICT project “REPLICATOR” No. 216240.

References

[1] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, “Vision and navigation for the Carnegie-Mellon Navlab,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362 – 373, May 1988.

[2] K. Košnar, T. Krajník, and L. Přeučil, “Visual Topological Mapping,” in *European Robotics Symposium 2008*. Heidelberg: Springer, 2008.

[3] T. Krajník, J. Faigl, M. Vonásek, V. Kulich, and L. Přeučil, “Simple, yet stable bearing-only navigation system,” *J. Field Robot.*, 2010.

[4] C. Valgren and A. J. Lilienthal, “SIFT, SURF and seasons: Long-term outdoor localization using local features,” in *Proceedings of the European Conference on Mobile Robots (ECMR)*, September 19–21 2007, pp. 253–258.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[6] N. Cornelis and L. V. Gool, “Fast scale invariant feature detection and matching on programmable graphics hardware,” in *CVPR 2008 Workshop (June 27th)*, Anchorage, Alaska, June 2008, June.

[7] D. Anguelov et al., “Google Street View: Capturing the world at street level,” *Computer*, vol. 43, no. 6, pp. 32–38, June 2010. [Online]. Available: http://dx.doi.org/10.1109/MC.2010.170

[8] M. M. Haklay and P. Weber, “Openstreetmap: User-generated street maps,” *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[9] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, June 2005.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Robotour Solution as a Learned Behavior Based on Artificial Neural Networks

Miroslav Nadhajský and Pavel Petrovič
Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University,
Mlynská dolina, 842 48 Bratislava, Slovakia, miroslav.nadhajsky@st.fmph.uniba.sk, ppetrovic@acm.org,

Abstract—**Our contribution describes a mobile robot platform that has been built for the purpose of the contest Robotour – robotika.cz outdoor delivery challenge. The robot is a standard differential-drive robot with a good quality consumer market digital video camera with a lightweight, but high-performance laptop computer used as the main control board. Supplementary board is used to control motors and sensors of the robot. The robot utilizes a behavior-based architecture and its vision module that is responsible for track-following is utilizing an artificial neural network that was trained on a set of images. This is a novel solution that has not been used in Robotour contest previously, and our early experiments demonstrate promising results.**

Keywords – robotour, navigation, artificial neural networks, learning robots

I. Introduction

Applications of robotics technology in both production and personal use are becoming possible with the development of new materials, motors, sensors and vision, ever decreasing cost of computing and memory capacity, and development of new algorithms and control strategies. Robots must be able to operate in dynamic and unpredictable environments. Therefore, one of the most important challenges to be solved reliably is robot navigation – in both indoor and outdoor environments. The robots must be able to localize themselves on a supplied map, create their own map representations of the explored environment, and they must be able to navigate their environments safely, without colliding with obstacles, or failing to follow the paths, roads, trails, and tracks. The real improvements in the technology typically occur when there is a large motivational pressure to produce a working solution. This might either be a goal to produce a final product, or alternately, with somewhat more relaxed requirements and settings, which are suitable for experimentation, and research, when the goal is to develop a robot to participate in a robotics contest.

Robotour – robotika.cz outdoor delivery challenge, organized by the Czech association robotika.cz, is an annual meeting of teams building and/or programming outdoor robots that navigate in a city park filled with trails, trees, grass, benches, statues, water ponds, bridges, and people. The task changes every year, but the main challenges are 1) be able to localize and navigate on a map supplied by the organizers, and 2) be able to follow the trails and paths without colliding with the obstacles or leaving the path without reaching the goal. See [1] for the exact rules of this year's contest.

Various solutions for the challenge were developed, however, in most cases, they did not take advantage of advanced artificial intelligence algorithms. In particular, only few different vision algorithms were developed until today, several teams shared the successful solution of [2], and many solutions rely on the use of odometry, compass, and GPS. We would like to address this area, and prepare a solution for the contest in 2010 or 2011 that will utilize AI algorithms. The second author has participated in the competition team several times in the past, and collected some experience and motivation for a new attempt. In this article, we describe the principles our solution is based on and is currently being built. In the following sections, we describe the mechanics and the hardware, robot overall architecture, the software components, and the AI methods that we aim to use. Finally we summarize the experience with building and programming the robot up to date.

II. Mechanics

The robot is a simple robot with differential-drive kinematics with one supporting free-rolling caster wheel. The length of the sides of its square base is 45 cm; the air-inflated wheels of a diameter 15.3 cm are mounted on the outside of the base, in the front of the robot. The total weight is about 6 kg without any load. The robot provides a storage space of ca. 20 x 20 x 45 cm to carry a heavy load (approx. 5 kg), which can be placed close to the center of rotation, above the propelled wheels, so that it does not have a negative impact on maneuverability of the robot. The main control unit is a portable computer, mounted in a flat plastic frame with a foam to compensate the shocks. The lead acid 12V 9Ah rechargeable battery, being the heaviest component, is stored under the base, between the wheels, keeping the centre of gravity low. Color camera with a true optical image stabilizer and CCD image sensor is mounted using anti-shock foam on a U-shape construction frame built of aluminum profiles, together with GPS and IMU sensor, see Fig.1. The camera is inclined 10° downwards. The IMU sensor must be mounted far from any sources of electric and magnetic fields, such as motors and wires. Placing GPS high compensates also for obstacles in the surrounding terrain, which may hinder the GPS satellites signal. The robot is built from raw materials, except of the motors, wheels and consoles that hold them, which are all part of a set from Parallax. The aluminium framework allows mounting a rain shield for the computer and the camera when necessary.

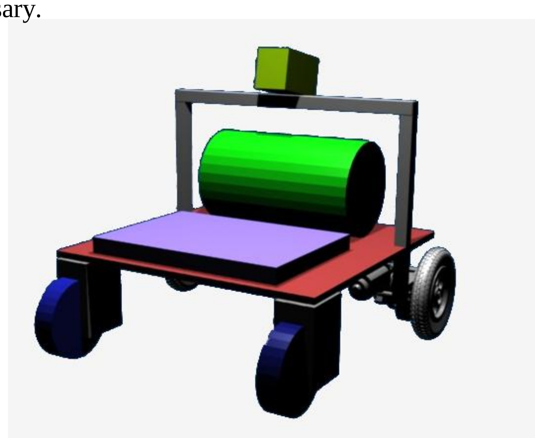


Figure 1. 3D Model of the robot showing main parts. In real implementation, we have mounted only one caster wheel as it proved to be sufficient, and allowed more accurate control.

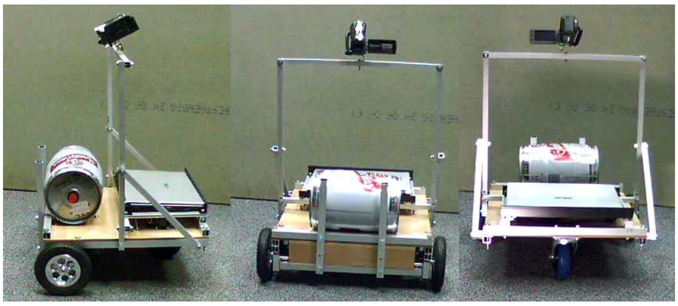


Figure 2. The resulting constructed robot from the side, front, and back. The control electronics is installed under the PC. The robot has already been tested in outdoor settings and has traveled a distance of several km.

III. Hardware Architecture

The robot is propelled by two 12V DC motors with built-in transmission, rotating at up to 150 rpm and consuming 1.5A at no load. The encoders with 36 ticks per rotation are used for speed and position feedback and are equipped with on-board microcontrollers that are directly connected to the motor drivers HB25, supplying them with the proper PWM signal to keep the requested speed. In this way, the main microcontroller board, which is the SBot control board, designed in our group originally for SBot mobile robot, is freed from the low-level motor control, and dedicates this task to both of the encoders that have an implementation of a standard P (proportional) controller and are connected using the same 1-wire serial bus. Unfortunately, we found that the original firmware for the encoders supplied by Parallax did not satisfy our needs for several reasons. Most importantly, the encoders were not designed for dynamic change of speed, but only for simple positional commands that accelerate from zero speed to a fixed predefined speed, and then decelerate after traveling the required distance. They do not allow to change the speed in the middle of such positional command. However, movements, where the speed and rotation is changed arbitrarily at any time, are required in the Robotour task, where the robot has to dynamically respond to the visual feedback when it has to align its movement with the shape of the path. Fortunately, Parallax makes the source-code for the encoders firmware available, and thus we could modify it to suit our application and support immediate smooth changes of the instant speed.

The obstacles are detected using the standard SRF-08 and Maxbotix LV EZ1 ultrasonic distance sensors that are connected to the main control board.

Outdoor robots are typically equipped with a global positioning device, i.e. GPS, and it is the case for our robot too. Information from the GPS module that is connected directly to the main computer using USB port, however, is not so reliable due to atmospheric and other occlusions, and serves only as a guidance for map localization. It is confronted with visual input and complemented by the current heading obtained from compass sensor. The compass sensor is part of the complex 9 DOF IMU sensor that includes several axes of gyroscopes, accelerometers, and magnetometers, thus compensating for various robot inclinations when traveling uphill or downhill. This is important since the simple compass sensors provide incorrect information once the robot and thus also the sensor is tilted.

Finally, for the visual input, we chose to use a standard video camera Panasonic SDR-T50, due to a very good ratio of parameters/price. The video camera is built around a CCD sensor, which has the advantage over the CMOS image sensors of taking the image instantly. Cheap CMOS cameras therefore suffer from a serious vertical distortion when the camera is moving, since the different rows of the image are scanned at different times. In addition, the camera has a built-in true optical image stabilizer, which further compensates for distortions due to the movement. Unfortunately, we found this stabilizer to be insufficient, and thus we have supported it with an anti-shock foam placed between the camera and the platform where it is tightened using flexible textile tape. The camera renders its image either as 16:9 or 4:3 image, however, it sends a wider signal down to its video output jack connector, which is further connected to a USB frame grabber card and the main computer. The main computer is a 2-core powerful PC with a GPU that can be used for the intensive image processing computation. The computer and the Sbot control board are connected using a serial port or a virtual serial port over radio BlueTooth connection. In debugging and testing applications, the robot can be controlled using a wireless gamepad connected using a proprietary 2.4GHz radio link.

In general, the robot is designed in such a way that it can be used in many different applications. For instance, a stereo vision system or an arm with a gripper can be installed in the cargo hold area. Additional sensors can be easily mounted on the aluminum profiles or wooden base. Fig. 3 shows overall system architecture.

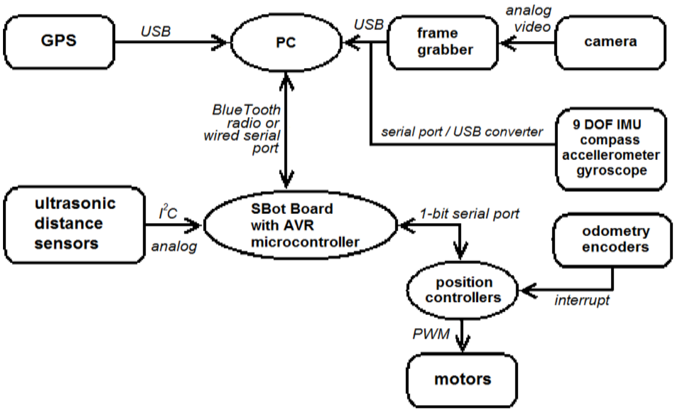


Figure 3. System hardware architecture.

IV. Software Controller Architecture

The software architecture is tailored for the Robotour contest. In this year's contest, the goal for the robot is to navigate to the target without knowing its starting location. It is only given the target coordinates and an official map of the park. It may not use other map information. The software controller is logically divided into five main components, see Fig.4.

The first component, planning, uses the map with the destination location and generates a path plan for the robot to follow. It tries to minimize the number and complexity of the crossings as these are the most critical places and candidates for navigational errors. The component outputs a sequence of locations that are to be visited by the robot. Whenever requested, the module can generate a new plan after a problematic place in the map has been reached.

The second component, localization using map, is responsible for the most accurate localization of the robot on the map. It is using the information from the compensated compass (IMU) for heading, from GPS for position estimation, and from the position encoders to estimate the distance traveled and turns made. All the information is integrated and with the help of the map and the path plan, the target distribution is determined using a probabilistic Monte-Carlo estimation. The output of the localization module is a probabilistic distribution over the expected heading in the very next correct movement, and the expected distance to the next crossing or target.

The third module, path recognition, is the most important one for the actual control of the motors, and has a priority over the localization module. It receives the image from the front camera and recognizes which parts of the image correspond to the path, and which of them correspond to other surfaces. The next section explains this procedure in more details. The output of this module is again a probabilistic distribution over the space of possible headings that can be projected to the input frame, where the headings leading to more “path” areas are more likely than those leading to less “path” area. Input from the odometry and gyroscopes helps this module to improve its estimation of the path using its previous estimations and the relative displacement of the robot.

The obstacle recognition module is responsible for detecting obstacles in the planned path of the robot and for stopping the robot in case of a possible collision early enough so that avoidance could be attempted by the coordination module. The robot is currently equipped with three ultrasonic distance sensors (front ahead, front left, front right), and thus the module reports on its output whether the path is blocked completely, or only partially, and also what is the size of the expected free buffer in front of the robot.

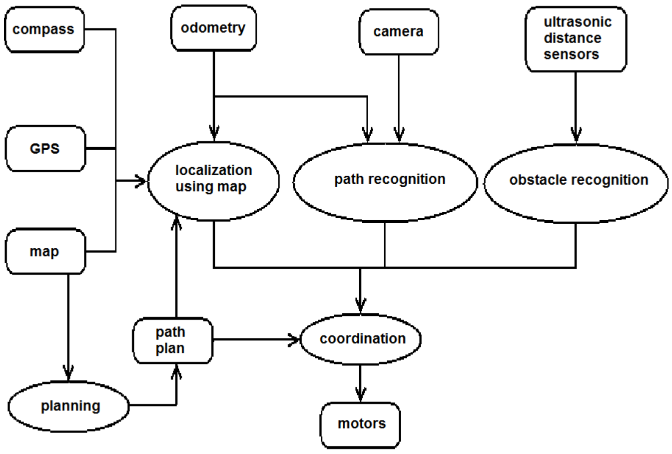


Figure 4. Overall controller architecture.

The most complex module is the coordination module. Its purpose is to take the prioritized outputs from the other three modules, and to determine the best possible angular and linear velocity for the next instant movement. When the confidence of the module is getting low, the robot slows down. If the confidence falls even lower, the robot stops, and starts rotating left or right, depending, which direction is expected to be more promising, until it finds a heading, where the module confidence is sufficiently high again. If such heading is not found, the robot attempts to return back in the reverse direction as it arrived to the problematic location, possibly moving in the reverse of the planned direction on the map. After returning back a short distance, it retries. The retries are repeated several times while gradually extending the back-up distance. If all attempts to pass the problematic location fail, the planning module is asked to generate a different path.

The controller is arranged in a behavior-based manner, individual behaviors are developed and tested independently before they are integrated in a common controller.

V. Path Recognition

Our goal was to use artificial neural networks in order to help the robot navigate and stay on the path. We obtained many images from a park with trails, and we have manually marked the regions in these images that correspond to the traversable path. This input was used to train the neural network (a standard multi-layer perceptron) to recognize the path. See figure 5 for an example of such manually classified image.

Figure 5. Manual preparation of training images.

Sending the whole image to the network as the input would obviously be infeasible. Instead, we first tried to scale the image to a lower resolution of 400x300 pixels, and divide it into 100 rectangular regions of equal sizes that covered the whole image. Each region formed an input to a neural network, and the whole region was about to be classified as "path" or "not path". However, the resulting resolution of the classified image was not satisfactory, even after a further reduction of the region size so that the image was divided into 2500 segments. Therefore, we decided to use a sliding region. For almost every pixel in the image, we define a corresponding region – it's larger neighborhood, which forms the input vector. The classification output produced by the network for each pixel in the image is then a real number from 0 to 1, estimating how much the network believes the pixel lies on the path. Two examples of images that were not used in the training phase are shown in the Fig.6.

Figure 6. Examples of path recognition.

We used the RPROP training algorithm for multilayer perceptron, in particular the implementation that is present in the OpenCV package. The training used tens to hundreds of manually classified images from various places in a park with various path surfaces, light and shadow conditions. Since this is still an ongoing work and only preliminary results are available, we restrain from a statistical analysis of the results at this moment, and refer the reader to the page dedicated to the project with detailed results and data [5].

Once the network is trained and produces the classifications for the image frame pixels, the path recognition module enters a second phase, when it tries to evaluate all possible travel directions (headings) with respect to the chances that the robot will stay on the path. For this purpose, the module analyzes a family of triangles of the same area with the base at the bottom of the frame and the third vertex placed in the middle of the image. For each such triangle, we compute an average path likelihood. The triangle for which the path is most likely, i.e. where most pixels lay on the path, is likely to be the correct new heading. However, the module outputs a full distribution over all possible headings so that the coordination module can take advantage of this information, for instance to determine different directions at a heading, or when trying to resolve ambiguous cases. Fig.7 depicts the analyzed family of triangles. Two example pictures are further analyzed in Fig. 8, where the bars show how "likely" it is that following in the various directions is a "good" idea in order for the robot not to leave the path.

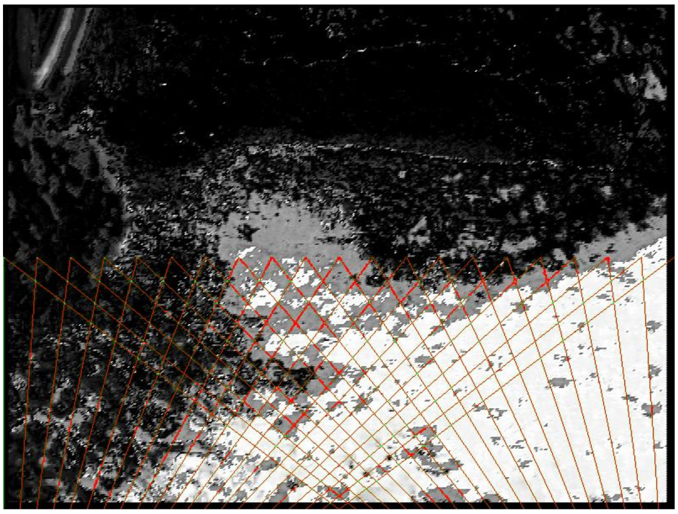


Figure 7. Triangles representing different turning projected to the image of recognized path.

VI. CONCLUSIONS AND FUTURE WORK

We have designed and implemented a robotic hardware and software platform to be used in the Robotour contest for outdoor robots navigating in park environment. The hardware platform is implemented in a general way and most components of the software platform can be reused in other applications, the robot can be extended with stereo vision or manipulator. We have designed, implemented and tested in this context a new method for path recognition, which is based on artificial neural network that is trained on a set of static images that are similar to the environment where the robot is to be operating. We are currently working on integrating all the components of our prototype so that it could perform in its first

Robotour contest this year. In the remaining 10 months of the project, we will analyze the results from our participation, and propose, implement, and verify improvements so that the robot can serve both as a competitive platform in the contest and as an educational tool in the course Algorithms for AI Robotics, which is provided at our department to students of Applied Informatics.

ACKNOWLEDGMENT

This work has been done with the support of the grant “Podpora kvality vzdelávania na vysokých školách“ from Nadacia Tatra banky.

REFERENCES

[1] Robotour - robotika.cz outdoor delivery challenge, Rules, online: http://robotika.cz/competitions/robotour/cs last accessed: August 1st 2010.

[2] K. Košnar, T. Krajník, and L. Přeučil, “Visual Topological Mapping,” in European Robotics Symposium 2008, Heidelberg: Springer, 2008, pp. 333–342.

[3] G. Bradsky, A. Kaehler, Learnning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, Inc., 2008.

[4] D. Guštafík, SBot v2.0 – Educational Robot for Clubs and Classrooms, User manual to Sbot robot, 2008, online: http://webcvs.robotika.sk/cgi-bin/cvsweb/~checkout~/robotika/sbot/2.0/doc/dokumentacia_en.pdf

[5] M. Nadhajský, Robotour diploma project page, 2010, online: http://virtuallab.kar.elf.stuba.sk/~mnadhajsky/

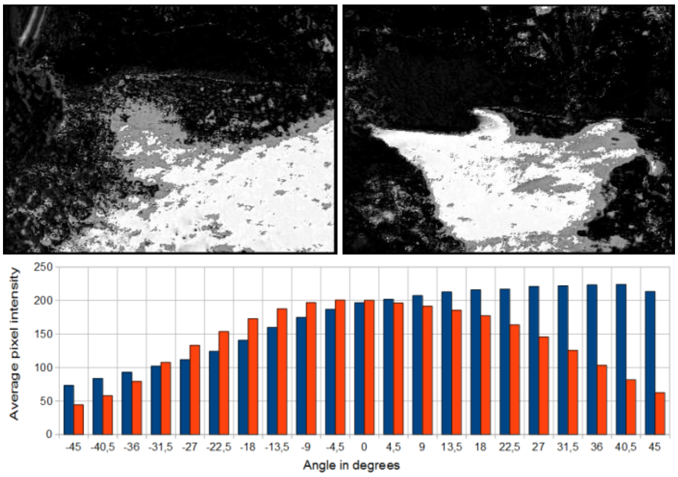


Figure 8. Two scenes after path recognition. The bars show the average pixel intensity of pixels inside of triangles for a range of different rotations for both of the resulting images (blue/dark for the left image, red/bright for the right image).

EDURO - Mobile Robotic Platform For Education

Martin Dlouhý
Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25, 118 00 Praha 1
Czech Republic
md@robotika.cz

Jan Roubíček
RobSys Czech Republic
jr@eduro.cz

Tomáš Roubíček
RobSys Czech Republic
tomas.roubicek@robsys.cz

***Abstract*—Eduro is a modular mid-size mobile robotic platform designed as both a teaching tool for higher education and a research platform for academia and industry. In this paper we describe the technology used within the Eduro (indoor) and Eduro Maxi HD (outdoor) product lines. Both platforms are designed around a tricycle base with two differentially driven wheels and one caster wheel. The on-board electronics consists of smart sensors and actuators connected by a CAN bus. The main controller module is implemented as a single board x86-based computer running Linux OS. This platform participated in several competitions including Eurobot, RobotChallenge and Robotour.**

Keywords: education robot, mobile platform, CANopen

I. Introduction

Eduro is a generic robotic platform intended for education and research. It was initially created as a teaching tool for Charles University, Prague in 2007, further development has continued independently at the initiative of the development team. At that time, none of the commercially available robots met the low cost/high performance requirements posed by the University's educators.

The Eduro platform is a successor of older platforms Berta, Daisy and Explorer. Berta - a triangle-shaped robot with vacuuming extension - won the 1st Annual Cleaning Contest in 2002, Lausanne, Switzerland [1]. The same triangular base was used in Daisy, a robot which ranked 7th at Eurobot 2003 in La Ferte Bernard, France [2]. Finally, the outdoor prototype Explorer - a 4-wheel waterproof robot - was demonstrated on Robotour 2006 in Prague [3].

The basic idea was to develop a platform that is highly modular on three levels: mechanics, electronics and software. The mechanics is designed as a construction kit with numerous mounting holes with pressed nuts. The electronic is based on a set of independent modules connected via a CAN bus. Finally, the low-level software modularity is achieved through the CANopen protocol and high-level modularity is facilitated through Player devices.

This paper is structured as follows: Section II describes the hardware platform in more detail. Section III outlines the software. Finally, examples of various configurations are provided in Section IV.

II. Hardware

A. Mechanics

The base of the robot is a construction module that includes the battery and motors. Modules such as the caster wheel or the control panel with buttons and indicators are attached to the base. These modules are made from aluminium profiles and sheets and have many mounting points for simple extending.

The Eduro is not waterproof by default but outdoor versions can be optionally sealed against dust and water.

Rugged plastic wheels are used in the indoor robot design. Such wheels have very good contact properties while they are still sufficiently sturdy for reliable encoder measurements. In outdoor scenario we tested two sets of wheels. The first approach involves smooth inflatable wheels with shallow tread pattern as demonstrated in Field Robot Event 2010 in Germany. These wheels were found suitable for park roads and other easy terrain. The second option utilizes arrow shaped wheels commonly used for small ploughing tractors. These wheels are recommended for rough terrain. They performed very well on muddy terrain when tested on RoboOrienteering contest. In the case of uneven surface with steep slopes, 4-wheel-drive configuration becomes a necessity.

B. Drive

Current members of the Eduro product line use SMAC (Stepper Motor - Adaptive Control) drives. This is original Robsys technology for gearless drives, which are based on closed loop controlled stepper motors. The motors are attached directly to driven wheel for indoor robots (Eduro) or by simple belt transmission for outdoor robots (Eduro Maxi HD). This gearless solution is very durable and easily withstands operations by unexperienced students.

Simple speed control with interpolation is presently used. The control system sends speed commands periodically, speed is represented by a 16bit signed integer, where 1000 corresponds to one shaft rotation. The drive sends back an encoder value - 32bit signed integer, where 65536 means one revolution of the motor. The drive has preset software limits for maximal acceleration and speed. Smooth motion can be obtained for speeds between 1 cm/s to 2 m/s given 150 mm wheels (diameter). It is recommended to maintain continuous flow of speed requests such that the wheels remain in permanent

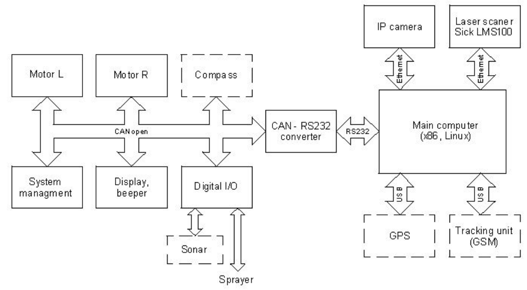


Fig. 1. Hardware structure

contact with the ground. The drives have implemented a communication watchdog which stops motors if speed command is not received within a predefined period.

C. Processing Power

The brain of the robot is a single-board computer running Linux OS. The computer is equipped with AMD Geode CPU running at 500 MHz, 256 MB RAM, compact flash card, wi-fi, 3 Ethernet, 1 RS232 and 2 USB ports. RS232 port is dedicated for CAN bus connection via transparent RS232-CAN bridge. High data throughput without data loss is secured by real-time serial driver.

D. Communication network

The Eduro uses CAN bus as its main communication network. All sensors and actuators with low data rate requirements are connected through the CAN. CANopen is the preferred communication protocol but other proprietary protocols can be used as well. CANopen is widely used in industry and hence many available sensors are directly compatible.

Cameras, laser range finders and other sensors with high data throughput are connected directly to the main computer via Ethernet or USB. Except for CAN and Ethernet, I2C and 1-wire Dallas buses are used in robots. I2C is a widespread interface for low-cost sensors, therefore it is supported. However I2C is not designed for large distances (I2C = inter-integrated circuits), therefore it is used only for short local buses. The 1-wire is used for a diagnostic network and advanced power management. Distributed power switches, thermometers, battery chips and other simple sensors and modules are connected via the 1-wire bus. I2C and 1-wire bus are connected to the CANopen network through the gateways.

E. Energy source and power management

The power supply is provided by sealed lead acid batteries. The outdoor version Eduro Maxi uses two 12 V/8 Ah batteries while for smaller indoor robot one third of the capacity is sufficient. The whole robot uses a single power source to simplify management. The motors are powered from 24 V supply branch, directly from batteries. The main computer, CAN network and most of sensors are powered from a stabilized 12 V branch. The auxiliary 5 V power supply is present for simple connection of the low cost sensors.

A standard off-the-self charger allows continuous charging while the robot is in operation (e.g., code debugging). When compared to other platforms it allows several hours of autonomous operation and swapping batteries is usually not necessary although it is possible.

An important part of any mobile robotic platform is power management. Energy is a limited resource and thus it needs to be monitored regularly. Two level power management is used in robots. The base is a standalone electronics providing basic function as charging, voltage monitoring and power distribution. An optional module is connected to the CAN bus and adds remote monitoring and advanced functions. The module sends messages about system voltages, temperatures and other important information. When the voltage falls below the given threshold, temperature rises or other exception occurs, the module can automatically blink LEDs, turn on the beeper or even turn off motors independently on the main computer. The thresholds as well as the consequent actions can be preconfigured from the control software via CANopen. RF remote key is an invaluable accessory to the power management module and allows the operator to turn the robot off in case of an emergency.

F. Sensors

This section describes sensors which are used in robots and are connected via the CAN bus. A robot often includes other sensors such as cameras, laser range finders, a GPS unit, etc. These sensors are connected directly to the main computer via Ethernet or USB.

1) Compass: A compass is a part of the inertial unit. Currently, we use a two-axis compass HMC6352 from Honeywell. It is a one chip solution with I2C bus, however the chip is not visible from CAN. The data from compass and other sensors are periodically polled by the CAN module, processed and only then forwarded to the central unit. The azimuth readings from the sensor are converted into 1/100th of degree and sent over CAN bus as 16bit integer. The update rate is 20 Hz.

The sensor itself is represented as one of the layers in the "sandwich" of inertial unit, other layers can include an accelerometer or a gyroscope. The HMC6352 is only a two-axis magnetometer, therefore tilt compensation is not possible. We plan to integrate a three-axes magnetometer to facilitate tilt compensation in the future.

The inertial unit including the compass is mounted on top of the pole away from sources of magnetic fields and ferromagnetic objects. The module itself is covered by a plastic case and no steel parts are used. During experiments we observed substantial changes in sensor readings caused even seemingly minor attachments to the pole such as a small umbrella, therefore caution is needed. A presence of ferromagnetic objects can be compensated by the system but that requires recalibration and usage of non-linear transformation.

2) "Sharps" distance sensors: "Sharps" distance sensors are cheap IR triangulating sensors for distance measurement. They are often used for obstacle detection and simple navigation in indoor. They have an analog or binary outputs with various operation ranges. The analog or binary signal is routed to universal CAN I/O module SC-DM04. This module has four analog or digital inputs and four digital outputs. The module can have additional function, for example outputs for RC servos or switch array decoder.

3) Sonar: Sonar is another sensor which can be connected via the universal I/O module SC-DM04. With special firmware the module behaves as a pulse decoder coming from sonar. The decoder accepts the SRF05 module from Devantech or compatible. There is also the option to connect the sonar with I2C interface via I2C to CAN gateway.

4) IR beacons: IR beacons were originally designed to facilitate precise robot navigation into the docking station but they can be used for wide variety of other applications. The transmitter consists of a circular IRED array. It transmits coded omnidirectional signal. The beacon has selectable code and signal intensity.

The receiver is also of circular shape with IR photodiodes attached on the perimeter. It can evaluate distance and angle for up to four beacons. The angle is calculated from the ratio of photodiodes currents and distance from intensity. Angular precision is 2-3 degrees and intensity corresponds to logarithm of distance in approximately 32 steps. The readings are reliable in most environments up to 3 meters.

The IR beacon system was tested both indoors and outdoors. It was presented at Eurobot 2009 as a sensor for absolute localisation on the playground and for opponent detection. Outdoor application was demonstrated at Robotour 2008. The set of one transmitter and two receivers facilitated reliable robot colony guidance. The following algorithm was actually simple enough so one of CAN modules was used for the control.

5) Bumpers: Robots often require various bumpers for object and collision detection. There are several options. The simplest one is set of micro-switches connected to digital I/O module. The status message is sent immediately after change (with limited frequency), and regularly once every second.

Another option is to use digital sharps. They are contactless sensors with detection range of 5 cm and 10 cm. The output is again digital and is handled the same way as with micro-switches.

6) Other modules: The set of available sensors and actuators is much wider and grows over the years. Among those not mentioned is a thermometer which is useful for gyro offset compensation and as a guard for battery charging. A light gate can be configured together with a servo module for automatic gripper action. Ultra bright LEDs can provide light for a camera in darkness[1].

G. User Interface

Eduro has a simple user interface, primarily used for Eurobot contest. There is a set of color LEDs, a selection switch, an easily accessible emergency stop button, recently an alphanumeric display and a beeper were also added.

[1]Used in robot Explorer in pipe investigation task

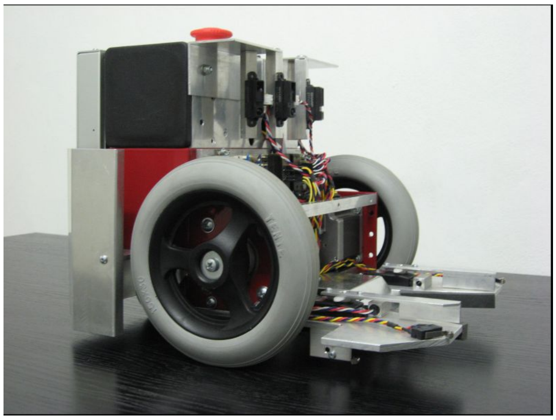


Fig. 2. Eduro prepared for Eurobot 2008 contest

The emergency stop feature deserves an extra note. Eurobot contest rules require that in case of an emergency pressing the emergency button disconnects all powered components - typically drive motors - from the power source. In reality, this simple solution would not stop the robot due to inertia. The implemented algorithm first sends stop commands to motors and shortly after that it disconnects the power. This solution at least slows down the robot.

III. Software

Application software can communicate with the base platform on several levels. The most commonly used include high-level standard Player interface and low-level direct access to CAN bus via RS232-CAN bridge. Another option is to leverage the set of Python library modules and functions which can be used for quick prototyping and was successfully used in most of this year's contests (see section with presentations).

A. Player

The Player/Stage (P/S) project [4], [5], [6] has been hosted on sourceforge since 2001 and it has become a de-facto standard interface for mobile robotic platforms. P/S is an open source project that originally targeted Active Media robots. However, the current set of supported platforms and devices is much larger, mostly thanks to the open source nature of its distribution which allows it to be easily extended to new machines.

The Eduro platform started supporting Player 2.1 in 2008 due to the interest from the development team members and collaborators who were familiar with this system from their work on other projects. Even though some of these contributors stopped using this system and moved to proprietary Python code due to problems with binary incompatibility between versions and bugs in even simple tools, we plan to support Player 3.0 on all Eduro platforms.

B. Pyromania

While it may seem unwise to build robotic control around a scripting language like Python, we found this approach to

Fig. 3. Eduro on Robot Challenge 2009 contest

Fig. 4. Eduro Maxi HD on Field Robot Event 2010

be quite appropriate and plan to keep leveraging it for even larger and more complex systems.

The time-critical control routines in Eduro are implemented through dedicated CAN modules. Computationally intensive tasks such like image processing can run in separate threads using Python's binding to OpenCV [7] or, if necessary, in separate programs written in more efficient languages (e.g., the C language). Even in these scenarios, Python remains present in its role of the integration language.

One of the major features, which Player lacked[2], was simple portability between Windows and Linux operating systems. We developed code for both platforms since limiting ourselves to only one would limit its appeal to potential users.

C. Direct control

The lowest level of robot control can be realized via direct access to CAN bus through serial line and RS232-CAN bridge. Programming on this level requires basic knowledge of CAN and CANopen protocols respectively as well as familiarity with detailed specification of incoming and outgoing messages for all modules.

IV. Configuration Examples

A. Eurobot

Eurobot [8], [9] is an annual international indoor competition for autonomous robots. Robots compete in solving a specific task that differs year to year but generally involves reaching certain goals within an operating space of about $2m \times 3m$ and within 90 seconds time limit.

The Eduro platform participated in three Eurobot events using the same base but varying mechanical attachments designed for that year's specific tasks. In "Mission to Mars"-themed event in 2008, this attachment was an automatically-triggered gripper. This gripper was implemented using servos, a lightgate module connected to the CAN network using bumpers, digital Sharps distance sensors (boundary detection) and analog distance sensors (feeder and opponent detection).

In 2009, the task involved building "temples". That year the attachment was a simple passive plowshare while an IR beacon system was used for opponent detection. The same system was also used for global Monte Carlo Localisation via triangulation.

[2]Player 3.x was already fully ported to Windows OS.

In "Feed The World"-themed event in 2010, the Eduro platform was equipped with a ball collector in front of the robot. The previously used modules were enhanced with a beeper and an alphanumerical display. The beeper was used to generate acoustic warnings in case of inconsistency between localisation detected by the beacons and the color of the team. The alphanumerical display was used to show the selected strategy.

B. Robot Challenge/Puck Collect

Videos showing Eduro's participation in Robot Challenge 2009 and 2010 contest [10] in Vienna are available. This event's theme and rules stay the same every year. The Eduro platform fits best in the "Puck Collect" category. In this competition, the goal is to collect red and blue pucks scattered around a white playing field ($2.8m \times 2.8m$) and carry them to the "home base" (colored squares $0.7m \times 0.7m$ located in opposite corners).

Eduro was equipped with a U-shaped passive collector so pucks were collected when the robot moved forward or turned in place. Dropping the pucks was implemented through backup motion. IP security camera with wide fish-eye lens was used for color recognition. Finally, long range Sharps (1.8 m) were sensing the border of the playground and also facilitated localisation services.

C. Field Robot Event

The outdoor version of Eduro (Eduro Maxi HD) participated on several outdoor competitions. In Field Robot Event [11] held in 2010 in Brawnschweig, Germany the robot was expected to perform various farming-related tasks in a mature corn field. The robot was equipped with an IP camera, LMS100 laser scanner, a compass, a GPS unit and other modules previously used in indoor competitions (e.g., beeper, display, user panel). A sprayer was connected to Eduro Maxi with a 3pin connector. Two logic outputs independently

Fig. 5. Authors and Eduro Maxi HD on RoboOrienteering 2010

controlled the spraying operation on the left and right of the robot.

For freestyle part of the event, the robot was equipped with a VTU10 tracking unit on loan from MapFactor [12] which in addition to tracking also facilitates two-way communication via a GPRS modem. The unit was attached to the robot via an USB port through which it accepted remote commands (GPS waypoint where the robot should autonomously navigate).

D. RoboOrienteering

A week after the Field Robot Event the same robot participated in RoboOrienteering event [13] in Rychnov nad Kněžnou, Czech Republic. This contest is similar to better known Robo-Magellan [14]. In both cases, the robots receive GPS coordinates for the starting point, waypoints and the end point and are expected to autonomously travel through the terrain between these points.

For this event the Eduro platform was equipped with tractor tires. Sonar was added in order to achieve better obstacle detection of benches and low placed tree branches.

V. Summary

In this paper we introduced Eduro, a robotic platform designed for education and research. Its modular design was proven successful through high rankings in numerous international competitions - 1st place in Professional Task of Field Robot Event 2010, 2nd place in Puck Collect at Robot Challenge 2009, or 2nd place in Czech Eurobot National Cup 2010. The Eduro platform has attracted enough interest for us to start its serial manufacturing planned for the end of 2010.

References

[1] Cleaning contest, "1st International Contest For Autonomous Cleaning Robots," http://www.service-robots.org/applications/cleaning.htm, 2002.

[2] Daisy, "quarter-finalist of Eurobot 2003," http://robotika.cz/robots/daisy, 2003.

[3] robotika.cz, "Robotour 2006 - robotika.cz outdoor challenge," http://robotika.cz/competitions/robotour2006/, 2006.

[4] Player/Stage, "device interface and simulator," http://playerstage.sourceforge.net/, 2001.

[5] Brian P. Gerkey, Richard T. Vaughan, Andrew Howard, "The Player/Stage Project:Tools for Multi-Robot and Distributed Sensor Systems," *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, June 2003.

[6] Richard T. Vaughan, Brian P. Gerkey, "Really Reusable Robot Code and the Player/Stage Project," *Software Engineering for Experimental Robotics*, 2006.

[7] OpenCV, "image processing library," http://www.sourceforge.net/projects/opencvlibrary, 1999.

[8] Eurobot, "Contest for amateur robotics," http://eurobot.org/, 1997.

[9] David Obdržálek, "Eurobot 2010," *Robot Revue*, May 2010.

[10] Robot Challenge, "Indoor robotic competitions," http://robotchallenge.org, 2010.

[11] Field Robot Event, "Agricultural robotic competition," http://fieldrobotevent.de, 2010.

[12] MapFactor, "VTU10 tracking unit," http://mapfactor.com/, 2010.

[13] RoboOrienteering, "Outdoor robotic competition," http://www.vosrk.cz/roboorienteering, 2010.

[14] RoboMagellan, "Outdoor robotic competition," http://www.robothon.org/robothon/robo-magellan.php, 2000.

Arduino Etoys

A programming platform for Arduino on Physical Etoys

Lic. Gonzalo Zabala   Ricardo Morán   Sebastián Blanco

Centro de Altos Estudios en Tecnología Informática
Universidad Abierta Interamericana
Buenos Aires, Argentina

gonzalo.zabala@vaneduc.edu.ar   richi.moran@gmail.com   sebastiangabrielblanco@gmail.com

Abstract— **In the last fifteen years, the technological education has been essentially based on digital technology, leaving aside the use of concrete material. Still having excellent simulators of the physical world, working with concrete material allows the development of cognitive structures that digital doesn't offer. Moreover, these didactic resources allow highly participative group dynamics that have not been yet reached by the existing computers at schools.**

Unfortunately, in our view there are two major difficulties for the presence of these resources in the classroom. The physical technology is expensive and suffers from constant wear. On the other hand, teachers are not accustomed to working in a dynamic classroom with a methodology of work in a participative group, and have fears about the use of specific technological equipment.

Physical Etoys is a development that aims to overcome these difficulties. Physical Etoys facilitates the interaction between inexperienced users and concrete material such as open hardware devices or popular toys by providing a powerful and intuitive visual programming system in order to explore and learn science in an enjoyable way. The objective of this paper is to introduce a new module of Physical Etoys which aims to persuade kids to do different electronic projects with an Arduino Board.

Keywords— educational robotics; Etoys; technology education; Arduino

I. REASONS FOR THE DEVELOPMENT OF THE PROJECT

Next we present the reasons that we have to manage with for the development of the project.

A. Fluidity in the use of technology

First of all, in the last fifty years the technology has taken a relevance in our lives that makes it difficult to think life without the integral use of them. It is for this reason that different analysts of the current school, as David Perkins [8] among others, considers that the presence of technology in classrooms and the necessity of a change of perspective keeping in mind the student more than its environment in its educational process are fundamental. That is to say, the student is no longer only the student: it is him plus his technological resources. It no longer cares where the knowledge is but how you access it. The problem is that, in spite of the exponential decrease of the costs of these resources, we are still in front of a considerable digital divide among those included and those excluded of the system. Gap that is not given by the access but for the significant use of technology. The more disadvantaged social classes are away from the metaphors that propose the current technologies. It is for that reason that the use of concrete material for the learning of technology allows to leave this framework and open conceptual and learning new opportunities. In synthesis, the children of all the social classes in their first years of life play with concrete material, and this game has a very deep load of technological learning. If we maintain this profile in the formal learning of technology, we will be able to reach a bigger number of students.

B. Technology with concrete material

Besides from a social greater reach, the concrete material allows us not only to develop intellectual activities but also sensory, that diminishes the problems of the passage from the concrete thought to the abstract one. In the physical experimentation, the student takes the error like a factor of his learning. and allows him to operate and control a group of continuous variables that no computer simulator provides. It is the real same world the one that defines the results reached by the boy's experiences.

However, the solution of problems with this material, allows the development of the systemic, structured, logical thought, but not starting from premises or abstract situations but from the solution of concrete problems.

Linda Williams [10] suggest the realization of activities with concrete material that generates processes not only in the left hemisphere of the brain (highly developed by the daily activities of the school) but also of the right hemisphere, what will allow to integrate components in a whole, with a simultaneous and parallel process, space and visual space.

C. Cross-curricular thematic and without gender difference

As we comment previously, technology is present in all the activities of our life. It is not in particular a privilege of any science or discipline, neither of any workspace especially. Therefore, it is fundamental that our students integrate the use of technology in all their subjects, and not simply in those where it seems "more natural" its presence. For it, we should leave the traditional framework of the technology teaching, where we develop devices with an end in itself, like the robot that follows lines. We should carry out significant projects for each child, to model devices that the man uses in his daily life, and that serve as excuse as a starting point, analysis or pursuit of diverse topics of the curricula. It is habitual that the technological activities of this type attract more boys than girls, for cultural diverse reasons that escape to this article. If we are able to propose the design of daily-life devices (for example, a table to create ceramic vessels, a microwave, a washing-machine, the dancer of a music's box, a turnstile), we will open the game to the cultural diversity that we have inside our classrooms.

D. Motivation for the learning

On the other hand, diverse studies that demonstrate the motivational impact that generates the use of these materials in the students, habituated to a not very participatory activity in the classrooms, exist. The possibility to build significant devices of concrete utility and the growing cycle in the learning that offers the test and error, generates in the student a deep interest not only in the construction but also in the contents linked to the carried out activity. That is, the use of these materials allows giving to the curricular content, even in cases of being less related with technology, a more significant framework reference for the student.

E. Teamwork

Work in classroom with these physical tools becomes impossible individually. Teamwork is necessary, beyond the economic limits in the purchase of equipment. It is important to order this teamwork with differentiated roles so that each participant has a specific and concrete work in the activity. Each of these roles enables the student to develop a skill set. Therefore, these activities allow us to introduce learning about teamwork and roles, conflict resolution, respect for differences and the need to listen to all members of the team. Each participant has its own point of view that enriches the work of the team. The proposed roles are related with the organization of working materials, the construction process, the communication with the teacher and the others teams, the development of written reports, and other activities.

F. Use of free or low cost hardware

The main cost of these projects is not the software, but the hardware platform used. For this reason, we decided to make the platform to program open or low- cost hardware, or robotics kits with presence in the schools of the world. Then, schools still have no equipment can purchase low-cost material as Arduino board. Schools that have some robotic kit or robotic toy can enrich their use, programming them with Physical Etoys.

II. TECHNOLOGICAL CHARACTERISTICS OF THE PROJECT

A. Cross-platform

One of the goals we define in the development of this project is the possibility that works both on Linux and Windows. In addition, the works developed in it should be cross-platform too. During the development, we were also requested that the software worked on Sugar, the operating system of the XO, the computers of the OLPC project. Nowadays it runs on 90% in the three systems.

B. Extensible

The experience we have lived in the education technology community suggested us that the development would not only be open but also easily extensible. The hardware proposals for the teaching of technology emerge every day, and we want to provide the possibility that each technology developer can build their tools on our platform. This is the reason we developed an easily extensible framework with basic knowledge of Etoys.

C. Why we use Etoys?

Etoys, the new educational version of Squeak, is an education tool to create multimedia and interactive projects. It has a long tradition of open development, because it was made by the Smalltalk team: Alan Kay, Dan Ingalls and other researchers. Furthermore, their educational criteria have been defined by great educational thinkers, such as Jerome Bruner and Seymour Papert [7]. Etoys is a highly effective tool for teaching math, science and arts, in a context of play and experimentation. Moreover, it’s cross-platform and has become the most important software on the OLPC netbooks, since it comes integrated with Sugar, from the outset. A large academic community is present behind its development, as MIT, Viewpoints Research Institute, University of Illinois, etc.

III. PHYSICAL ETOYS: OVERVIEW

Physical Etoys is a visual programming tool that connects the virtual world of computers with the real world in which we live in. With Physical Etoys it is possible to program real world objects (such as robots) to perform interesting tasks, or sense the world and use that information to control virtual objects (such as drawings on the screen).

It does not require any programming skills, and its consistency across the entire system makes it easy to accomplish some reasonably complex tasks that would be almost impossible in a different one.

Physical Etoys is an “extension” to Etoys. This is a wonderful software that helps children explore their own creativity in fun and educational ways, but it lacks communication with the outside world. Physical Etoys aims to overcome this necessity.

In outline, Physical Etoys is divided into a set of independent modules. Each module is responsible for controlling one robotic kit. Even though these modules can work independently from each other, the connection between them produces the most interesting results.

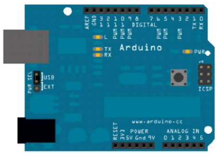
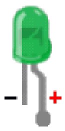
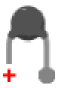
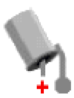
IV. WHAT IS ARDUINO?

Arduino is an open hardware platform based on a simple microcontroller board with digital and analog I/O pins. Due to its open philosophy, every teacher can access to different designs and build his own board (it is also possible to buy a prebuilt board). In addition, there is a great variety of examples of Arduino and a very collaborative community that is fond of helping people.

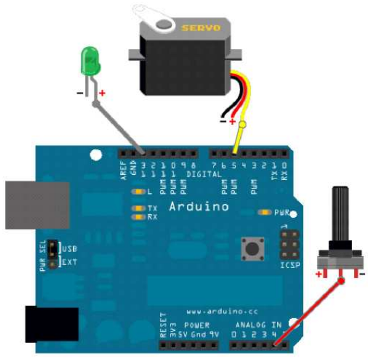
These characteristics are suitable for people who want to start using physical technology. Although the Arduino's official software is intuitive, it is still a low-level language like C and it looks cryptic for the average user.

V. Using Arduino with Physical Etoys

All Physical Etoys modules are composed by a few objects that try to resemble the real objects of their respective kit. The Arduino module is not an exception. You can see in the table below some of the Physical Etoys' objects and their correspondence in reality.

Name	Virtual object	Real object
Arduino board		
Buzzer		
Led/Pwm Led		
Photoresistor		
Potentiometer		
Pushbutton		
Servo		
Switch		
Thermistor		
Tilt switch		

The "Arduino board" is the main object of the Arduino kit. It contains pins on which other electronic devices can be attached using wires. All these interactions between real objects have been represented in Physical Etoys as you can see in the picture below.



1 – Components attached on a virtual Arduino

Every object has its own set of properties and commands that are accessible using the same interface. For instance, the "Led" object has an "is on" boolean property, the "Servo" has a "degrees" property, the "Photoresistor" has a "light value" property, the "Thermistor" a temperature value property, and so on.

This interface is also shared with all the graphical objects in Physical Etoys. Texts, sliders, pictures, buttons and every user interface widget that composes Physical Etoys is accessible and programmable in the exact same way (although they contain a different set of properties and commands). This extreme consistency across the entire system makes it really easy to use and explore.

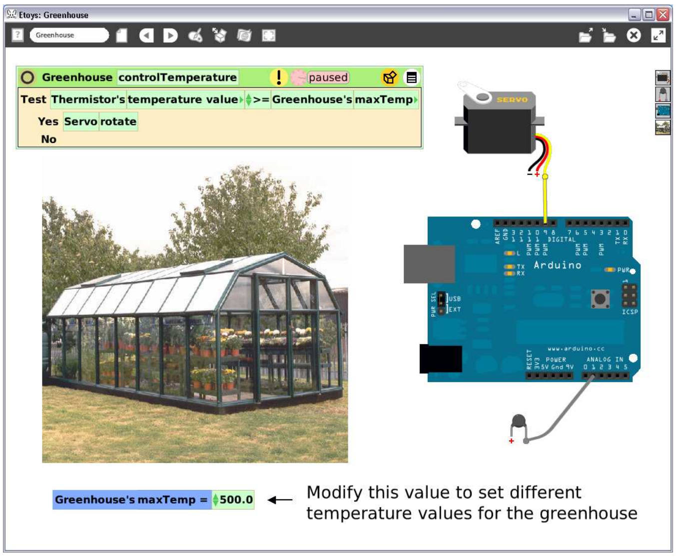
VI. Educational Examples

This section will describe a few exercises that can be implemented in a classroom.

A. Building a greenhouse

It is possible to build a miniature model of a greenhouse by using a servomotor and a thermistor. The motor will be used as a fan that keeps the greenhouse cool and the thermistor will sense the temperature of the air and it will activate the motor when its value exceeds a certain number.

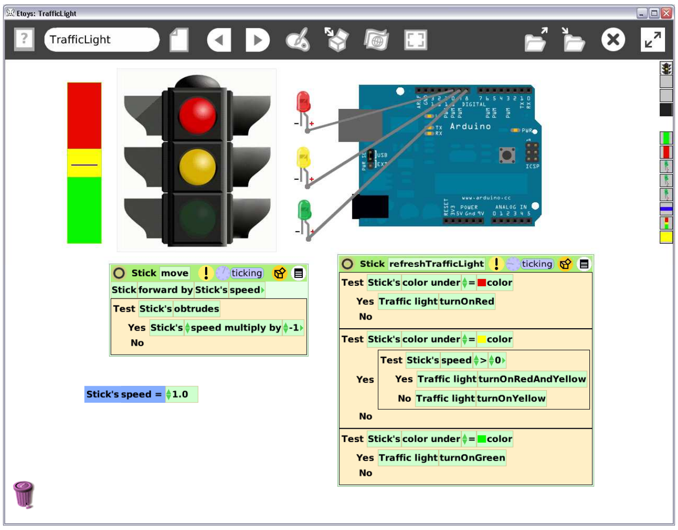
The picture below shows a simple implementation of the greenhouse project. The script "controlTemperature" at the top of the picture is the responsible of the behavior described above.



2. Greenhouse project implementation.

B. *Building a traffic light*

This exercise is a little more complicated. It uses three leds of different colors to represent a traffic light. Each led is turned on/off depending on the color behind a little "Stick" that moves across three different backgrounds: red, yellow and green.



3. Traffic light project implementation

These examples show two essential aspects of Etoys programming. On the one hand, it shows how abstract information such as the traffic light state and its behavior become concrete. On the other hand, it shows how the information of the world such as the temperature of the air can be conceptualized as numbers which can be used in any arithmetic or logical operation.

VII. Other Hardware Platforms Supported by Physical Etoys

The other modules composing Physical Etoys are listed below:

- *Nintendo Wiimote:*

  The famous Nintendo Wii's Joystick which detects the gesture of a hand, enabling the user to make scripts with a non-conventional way of communication with the computer.

- *Parallel port:*

  A type of interface for connecting various peripherals to the computer.

- *Lego Mindstorms Nxt:*

  A programmable robotics kit released by Lego. It allows the user to build almost anything without any knowledge of electronics. Considering that a lot of schools around the world already utilize the Lego Nxt to teach robotics, using Physical Etoys to program it is ideal for children that are just starting on the subject.

- *RoboSapien V2, Roboquad and I-Sobot:*

  These robots can be controlled by using an infrared transmitter. They are prefabricated and although their capabilities are limited, they are very attractive to the general public.

VIII. Physical Etoys in the World

Different educative communities have shown interest in using Physical Etoys on their own classes and workshops after the publication of its modules:

The SqueakNxt module, responsible of controlling Lego Mindstorms Nxt robots, has been used by an educative organization called Planète Science which took place in a workshop of Introduction to Robotics given at the Japan Expo Paris in France 2009. This non-profit organization intends to spread the science on the youth by organizing multiple activities including workshops at festivals and national contests such as the Final Eurobot, the French Robotics Cup and the First Lego League of France among others. During the Japan Expo Paris 2009 they used the SqueakNxt module to do different projects including:

- A drawing robot (similar to Logo).
- A robot that reacts to the environmental noise (its arms moved when somebody shouted).
- A robot that navigates through the exposition avoiding people.
- A robot that navigates through the exposition in order to lift plastic glasses using its clamps.

Planète Sciences has also shown interest in the Arduino Project, which has also been included in a software pack called SqueakBot, similar to Physical Etoys. Educational robotics has become mandatory in the French official curricula so there was a special class oriented to teachers in the region of Toulouse about the basic concepts of electronics and programming with Physical Etoys.

In Colombia a company called HYPER Neurotek which develops and integrates new technologies with education (preferably open-source projects) has shown interest in using Arduino to teach children how to use microcontrollers for building robots with an OLPC laptop.

In Spain, Citilab, an institute for the formation and the spreading of the ICT in Barcelona, decided to use SqueakNxt and Arduino for its Introduction to Robotics talks.

Finally, in Brazil, a consulting company called O3 Tecnologia that works in the area of educational technology used the Parallel Port project with Physical Etoys in robotic classes in high school.

IX. Conclusion and Future Work

The recognition that Physical Etoys has received in this short time is fills us with pride. This invites us to new challenges. The first one is to fully support the use of all hardware platforms on the three operating systems. We also have requested to add MacOs to them. The next challenge is to incorporate the microphone and camera of the netbooks as sensors to our project. In the case of video, we must think how to provide students with an easy programming mode, removing the complexity that the image processing has. And finally, we will propose a simple physical structure with motors and sensors, allowing to locate the netbook on it for using as an autonomous robot. Physical Etoys has a long way to go. We invite you to do this together.

References

[1] D. J Ahlgren and I. Verner, “An international view of robotics as an educational medium”, International Conference on Engineering Education (ICEE'2002).

[2] D. Alimisis, M. Moro, J. Arlegui, A. Pina, S. Frangou, and K. Papanikolaou, “Robotics & Constructivism in Education: the TERECoP project”, EuroLogo, 40:19–24, 2007.

[3] M. Bers, I. Ponte, K. Juelich, A. Viera, and J. Schenker, “Teachers as designers: Integrating robotics in early childhood education”, Information Technology in childhood education 123: 145, 2002.

[4] A. Druin. “Robots for kids : exploring new technologies for learning”, San Francisco: Morgan Kaufmann, 2000.

[5] Y. B. Kafai and M. Resnick, “Constructionism in practice: Designing, thinking, and learning in a digital world”, Lawrence Erlbaum, 1996.

[6] A. H. Odorico, “La robótica desde una perspectiva pedagógica”, Revista de Informática Educativa y Medios Audiovisuales 2, no. 5: 33–48, 2005.

[7] S. Papert, “Mindstorms : children, computers, and powerful ideas”, 2º ed, New York: Basic Books, 1993.

[8] D. Perkins, “La escuela inteligente. Del adiestramiento de la memoria a la educacion de la mente”, Gedisa Editorial S A, 2003.

[9] E. Sánchez, “La robótica pedagógica”. Educación, universidad y sociedad: el vínculo crítico: 117, 2004.

[10] L. Williams, “Aprender con todo el cerebro”, Barcelona: Martínez Roca, 1986.

[11] G. Zabala, “Desarrollo en un entorno educativo de objetos para el control de una interfaz de domótica”, Anales de WICC, 2007.

Educational Robotic Platform based on Arduino

Richard Balogh
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
richard.balogh@stuba.sk

Abstract — **The design of a new controller board for a mobile robot based on the Parallax Boe-Bot chassis is described. Disadvantages of the original Basic Stamp processor disappeared, more complicated tasks can be solved. As the board is compatible with the Arduino platform, also the open source development environment can be used. The requirements, design process and technical parameters are described. Also some illustration examples are shown.**

Keywords – controller, mobile robot, Arduino, robotic platform

I. Introduction

In our university we have been using for many years the commercially available mobile robots Boe-Bot[1] by Parallax, Inc. for education [1]. They were used in some laboratory exercises for students of the Mobile robotics lectures, some additional lectures for students of Embedded systems, or Automotive control systems. We were using this platform also for summer courses, student projects and for public presentations.

Our main problem with the Boe-Bot robot was with its controller unit. Although the Basic Stamp II with its programming capabilities is very reliable and useful for start up, our advanced students at the University were critical to use BASIC as a programming language for robots. They lack function definitions, program hierarchy, interrupts, parallel tasks, and direct access to peripherals like timers, counters etc. Our experiences with 8-bit RISC AVR processors by Atmel and growing popularity of the Arduino platform lead us to the design of a completely new controller board for robots. The main goal was to achieve as much compatibility as possible. Not only dimensions (which are essential to replace the board for the original one), but also the overall concept, connectors placement etc. were sustained. Now, we can use the robot with almost all original extensions of the Boe-Bot robot.

II. Components of the System

The Boe-Bot mobile robot [2] is a commercially available robotic kit by Parallax, Inc. It consists of two geared motors mounted on an aluminium chassis, batteries and control electronics. On the motors, there are mounted two plastic wheels. The rear wheel is made of a drilled polyethylene ball. Mounting holes and slots may be used to add custom robotic equipment.

The robot is controlled by the Parallax's popular microcontroller Basic Stamp II and the Board of Education. It is a simple board containing a processor, power supply circuits, interfaces, connectors and a small experimental solderless breadboard. The Basic Stamp II processor can be programmed with the PBASIC language - simple, but powerfull clone of BASIC language with a support of many specific peripheral devices [2]. Pros and cons of this platform were evaluated in details in [3].

Arduino is an open-source electronics prototyping platform based on a flexible, easy-to-use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments [4]. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing) [5]. The hardware reference designs (CAD files) are available under an open-source license, anyone is free to adapt them to its own needs.

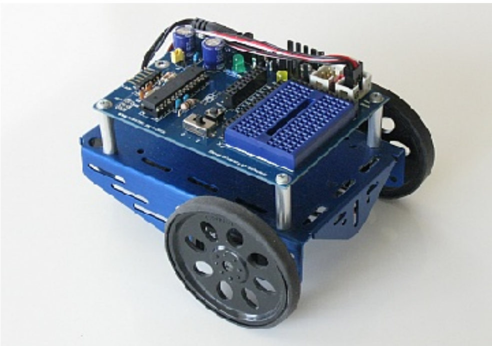


Figure 1: The Boe-Bot mobile robot with the new controller.

Figure 2: The number of searches for the term 'Arduino', relative to the total number of searches done on Google (source: Google Insights for Search).

1 http://www.parallax.com/Store/Robots/AllRobots/tabid/128/ProductID/302/List/1/Default.aspx

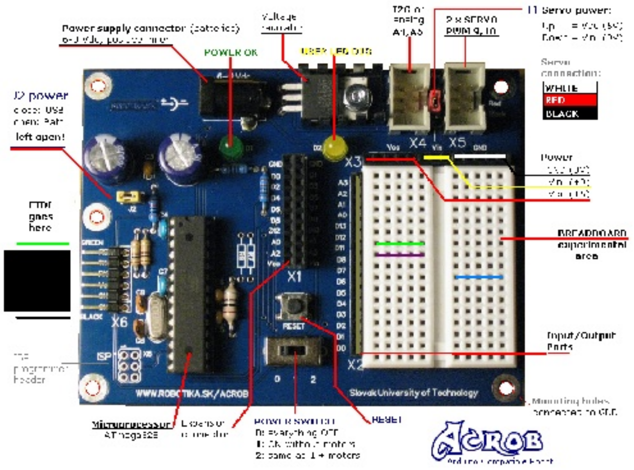


Figure 3: Description of the board.

This is also our case, we designed a completely new board retaining the compatibility with the platform. Arduino growing popularity (see Fig. 2) provides us with great supporting community with constant development of the software, many examples, tutorials and projects available.

III. Hardware

The new board is as much compatible as possible with the original Board of Education by Parallax Inc. [6]. The main differences are two: a different processor and a TTL serial interface without the converters.

As a power supply we can use the battery box (with four primary or rechargeable 1,5V AA size batteries) or a wall adapter. The power switch has three positions. Except the standard on/off positions there is a special “development” position when motors are disconnected so the robot is not moving on the desktop during the debugging.

An on-board voltage stabilisator provides 5V for the microcontroller and its peripherals. As the main processor, Atmel Atmega328P with a pre-burned bootloader is used. It provides the user with 32kB of the program memory, 2kB of data RAM space and 1kB of the EEPROM space. The main area of the board is occupied with a solderless experimental breadboard which enables to connect different additional components. On its left side most of the I/O pins are available, on its top there is a power supply connector. The board also contains connectors for servomotors and two additional sensors with digital or analogue outputs. A dual line connector in the center of the board enables to connect standard Parallax's extension boards like the compass or the LCD modules. See also the description of the comprehensive set of connectors for peripherals contained on the board in the Tab.1.

Programming and communication capabilities were increased comparing to the original Boe-Bot robot. We decided to have only the serial communication interface with TTL levels without any other converters on the board, so different converters can be used. We can use the standard

TABLE I. Connector Description Table

ID	Purpose	Type
X1	Expansion connector compatible with the Parallax AppMods.	2x10
X2	Access to the I/O pins	1x16
X3	Power supply for breadboard (Vcc, Vin, GND)	1x13
X4	Sensors (I2C bus and/or Digital/Analog Input)	2x3
X5	Servomotors (2xPWM)	2x3
X6	Serial / Programming interface (bootload)	1x6
X7	Power supply jack	3,5mm
X8	In System Programming interface – ISP (MISO, MOSI, RESET,...)	2x3

FTDI Chips[2] USB cable or the SparkFun's FTDI Basic[3] module for programming using the internal bootloader. Also we developed RS-232 level converter module to enable operation also with a standard serial interface (see the Fig. 4).

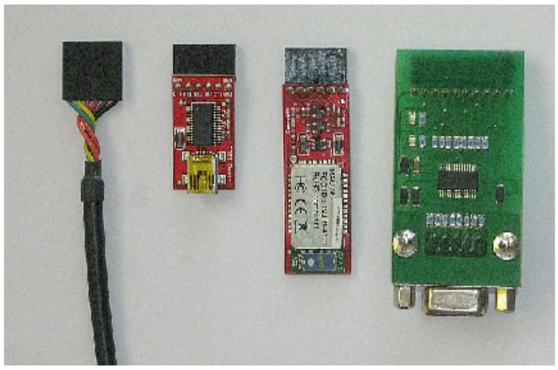


Figure 4: Programming and communication using (from left to right) a) FTDI USB Cable, b) SparkFun FTDI Basic module c) SparkFun Blue Mate Bluetooth module d) custom made RS-232 module.

After the program loading, the interface is free for any user serial communication operations. This enables to connect e.g. SparkFun's BlueMate[4] communication module to communicate with the computer or between robots using the Bluetooth interface. On the board there is also the connector for ISP programmer, so one can use any standard Atmel ISP programmer to burn the program into the processor. Together with the AVRStudio one can even debug, step and watch programs written in Assembler or avr-gcc languages.

The board can be used also standalone with the only connection using the FTDI USB cable. In such case user can power the board from the USB interface so no additional equipment is necessary. Such configuration can be used for introduction to the embedded systems programming, explaining basics of digital and analogue inputs, outputs or built-in peripherals like timers, counters, PWM and A/D converters. The schematics and the printed circuit board were

2 http://www.ftdichip.com/Products/Cables/USBTTLSerial.htm
3 http://www.sparkfun.com/commerce/product_info.php?products_id=9115
4 http://www.sparkfun.com/commerce/product_info.php?products_id=9358

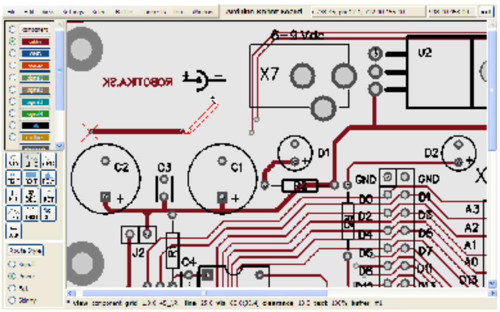


Figure 5: Design in the pcb program from the gEDA suite.

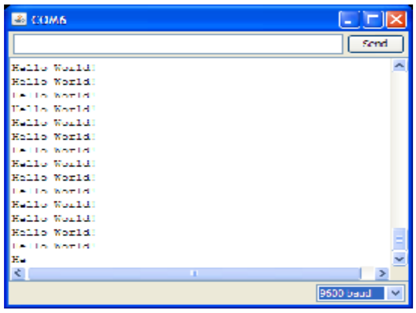


Figure 6. Hello World program in terminal window.

designed using the open source gEDA suite[5]. We used thru-hole components to enable students to build their own boards from the kits.

IV. Software

For programming, the standard Arduino IDE can be used. Other methods include the Assembler or avr-gcc languages integrated within the Atmel AVR Studio or using a set of command line utilities. We tested the environment on MS Windows XP operating system, but the Arduino IDE should work also on Linux and MAC OS systems. There is only one problematic point we found - during the installation process one need administrative rights to install USB drivers properly. This problem diminished in Windows 7 where drivers seemed to be already contained.

We prepared a set of basic programs to show an access to peripherals. We start with a basic digital I/O (LED and switch), then move to the analogue world – basic robot movements and analog sensor measurements. As the first analog sensor we find very useful Sharp distance sensors which are easy to connect and offer reliable results. Also their non-linear characteristics is challenging.

As the very first program we used the standard “Hello, World!” problem.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
 Serial.println("Hello, World!");
}
```

After the compilation and burning the program into the processor using the bootloader, a user can see the result using the internal built-in terminal window (see Fig. 6). Sometimes communication speeds didn't correspond to the real ones and characters were displayed incorrectly until it was changed.

From the listing above it is clear that programming is very straightforward and at the beginning no processor specific knowledge is required. Only important thing is to split the program into two basic parts – **setup** (which is performed only once) and **loop** (which is then performed infinitely – or better said, until it isnot switched off or reprogrammed). As the Arduino language is built over the standard avr-gcc compiler, we can still use all its features and combine also the standard approach e.g. direct access to all processor registers:

```
TCCR0A |= _BV(WGM01) | _BV(WGM00);
OCR0A = 127;
OCR0B = 255;
```

Of course, libraries can hide the internals from the user so no special knowledge is required. An example of a library for servos to show basic robot movements follows:

```
#include <Servo.h> // this program uses the Servo library

Servo LeftServo; // create servo object to control both servos
Servo RightServo; // a maximum of eight servos can be created

#define FAST  50 // try to change these values during the test
#define SLOW   5

void setup()
{
 LeftServo.attach(9);  // attach servo on pin 9 to the servo object
 RightServo.attach(10); // attach servo on pin 10 to servo object
}

void loop()
{                                // FAST FORWARD
 LeftServo.write(90 + FAST);     // value 90 is in middle, i.e. stop
 RightServo.write(90 - FAST);    // mirrored position
 delay(1500);                   // go fast forward for  1,5 s

 LeftServo.write(90 - FAST);     // ROTATE (PIVOT) LEFT
 RightServo.write(90 - FAST);
 delay(1500);

 LeftServo.write(90 - FAST);     // FAST BACKWARD
 RightServo.write(90 + FAST);
 delay(1500);

 LeftServo.write(90);           // STOP both motors
 RightServo.write(90);

 for(;;)  ;                     // stop the program operation here

} /* End of Loop */
```

For a comparison – a similar program written in the original Basic Stamp II language may look like this:

```
' {$STAMP BS2}
' {$PBASIC 2.5}

counter    VAR Word
pulseLeft  VAR Word
pulseRight VAR Word
pulseCount VAR Byte


' Forward
pulseLeft = 850: pulseRight = 650: pulseCount = 64: GOSUB Navigate

' Left turn
pulseLeft = 650: pulseRight = 650: pulseCount = 24: GOSUB Navigate

' Backward
pulseLeft = 650: pulseRight = 850: pulseCount = 64: GOSUB Navigate

END
```

5 http://www.gpleda.org

```
Navigate:

 FOR counter = 1 TO pulseCount
  PULSOUT 13, pulseLeft
  PULSOUT 12, pulseRight
  PAUSE 20
 NEXT
 PAUSE 200

 RETURN
```

V. Support

We created the supporting page at our robotics server[6] with all necessary documentation. Also we have started with creation of a comprehensive manual with example programs and connection diagrams. One of the big advantages of the open source system is a large community of users adding their experiences to the whole system. As the board is Arduino compatible, we can immediately start to use an existing repository of examples, documentation etc. If You want, for instance to connect an ultrasonic detector SRF-08 to a robot, you find very soon not only few examples but also a connection diagram[7] and even the whole library[8] for this sensor. Just type keywords 'SRF08' and 'Arduino' to your favourite search engine.

VI. Evaluation

A new robot, called Acrob (**A**rduino **C**ompatible **Rob**ot) was tested and evaluated at some different events. We prepared the robotic introductory lecture for students of the Automotive branch of study. The main goal was to give them an idea of mobile robots and its programming. During two lectures students were able to program basic movements and reactive behaviour of the robots. For students of the Mobile robotics course we prepared similar lecture and the platform was also used as an example of a differential driven platform. Also the infrared sensor distance detection was explained. Then we use the robots in a joined Austrian-Slovak lecture for secondary school students. During the lecture they were able to program movements, connect and evaluate line sensors and distance sensor so they finished with a simple line-following robot with an obstacle avoidance. In the time of writing this paper the robots weere intensively tested in the Summer School of Robotics and they were succesfull. The overall concept was succesfully tested also at the contest Robotchallenge Wien 2010 where the robot succesfully (though very slowly) passed the linefollowing category (see Fig. 7).

VII. conclusions

Presented robotic platform offers many capabilities. The main problem of the previously used Parallax's BoeBot platform –

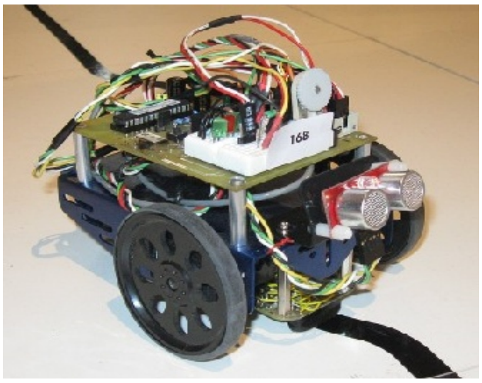


Figure 7: Prototype of the robot with linefollowing sensor and ultrasonic obstacle detector at the Robotchallenge contest.

programming in BASIC was successfully solved and programming is now possible both in assembly and C++ languages. Moreover we can use a large repository of examples and tutorials for the Arduino, libraries and components for this platform and also large community shared resources. The concept was proven in some robotic lessons for university and secondary school students and also at the summer school and robotic contests.

6 http://www.robotika.sk/acrob

7 http://www.robot-electronics.co.uk/htm/arduino_examples.htm#SRF08%20Ultrasonic%20Ranger

8 http://www.arduino.cc/playground/Main/SonarSrf08

Acknowledgment

This project is a part of the project CENTROBOT funded by the EFRE program Creating the future (CBC SK-AT 2007-13). Author also thanks to Parallax, Inc. for help and support.

References

[1] Balogh, R.: *Basic Activities with the Boe-Bot Mobile Robot.* In DidInfo 2008. 14th International Conference. FPV UMB, Banská Bystrica, Slovakia.

[2] Lindsay, A.: Robotics with the Boe-Bot. Student guide. Version 2.2., Parallax, Inc. Rocklin, California, 2004. ISBN 1-928982-03-4. Available on-line: http://www.parallax.com

[3] Hofmann, A. et al.: Robotic Education Platform. Literature Research and Evaluation. Technical Report, part WP2 of the Centrobot project. Wien, 2009. Available on-line: <http://www.centrobot.eu/en/aktivitaeten/robotic-education-platform/>

[4] Arduino Home Page. http://www.arduino.cc/

[5] Banzi, M.: Getting Started with Arduino. O'Reilly, 2009.

[6] Board Of Education. Development / Educational Platform for the BASIC Stamp and Javelin Stamp Microcontrollers. Parallax, Inc. 2007. Available on-line: <http://www.parallax.com/Portals/0/Downloads/docs/prod/boards/28150-BOE-Serial-v2.0.pdf>

Mobile robot Khepera III. Programming for MATLAB/Simulink environment

Maciej Garbacz
Department of Automatics
AGH- University of Science and Technology
Kraków, Poland
mgarbacz@agh.edu.pl

Mieczysław Zaczyk
Department of Automatics
AGH- University of Science and Technology
Kraków, Poland
mza@ia.agh.edu.pl

Abstract. **This paper presents mobile robot Khepera III and its programming environment. The laboratory stand and some results of made experiments was presented. The application is based on MATLAB/Simulink system. The proximity sensors and ultrasonic sensors are used to detect obstacles in robot's workspace.**

***Keywords.** mobile robot; path planning; obstacle avoidance;proximity sensors;ultasonic sensors*

I. Introduction

This paper describes the new laboratory stand, built in Laboratory of Automatics, Robotics and Fotovoltaics Systems. This stand is based on mobile robot Khepera III- product of swiss company k-Team. Khepera III is new, improved version of Khepera II [1]. The Khepera III is supplied by swapable battery pack composed of two Li-Ion Polymer elements. It is possible to work with robot continuously because there are extra battery packs and charger in laboratory. The communication between PC and Khepera is based on Bluetooth technology. In connection with the above there is no troublesome cable connection. Khepera III is equiped with proximity sensors and ultrasonic sensors to gather information about the workspace.

II. Khepera III Robot

The Khepera III has got a modular construcion. The robot has got a round shape to minimize result of collision with another robot. Robot is driven by two symmetrically placed wheels. Each wheel is moved by a DC motor coupled with the wheel through a reduction. The possible speed are between 14 mm/s to 298 mm/s. Each DC motor is equipped with incremental encoder. It gives information about the current position of robot. Some extra information about workspace give proximity and ultrasonic sensors. Robot is presented on fig. 1. The sensors visible at the top of the robot are ultrasonic sensors. Robot is equipped with 5 ultrasonic sensors type 400ST100/400SR100 of Midas company. These sensors consist of transmitter 400ST100 and receiver 400SR100. The carried out experiments shows, that ultrasonic sensors work better for greater distances. For smaller distances the proximity sensors are better. Khepera III has got 9 proximity sensors around robot and two extra under the robot to detect edge of table. In Khepera there are proximity sensors type TCRT5000 of Vishay Telefunken company. The sensors are placed around the robot as shown on fig. 2. Examples of characteristics of sensors are presented in next chapters. It is important that the more distance is, the smaller value of measure is. The value of measurements depends on condition of illumination, on material used to obstacles.

Figure 1. Mobile robot Khepera III

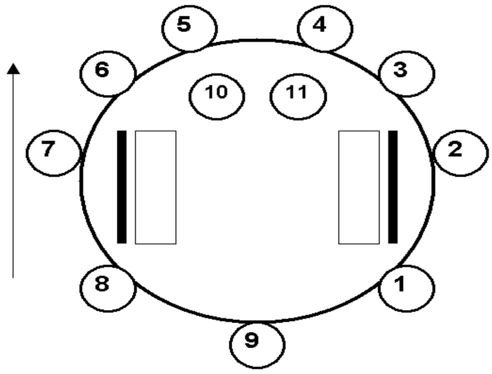


Figure 2. Position of the IR sensors

III. Programming for Matlab/Simulink

To make communication with Khepera easy there is constructed special library including ready-use command in MATLAB environment. Communication is based on serial line and it is possible in various environments. But the MATLAB is the most popular in our Department. In our application the RS232 protocol is used. PC is a master and Khepera is a slave.

The PC always initializes connection. Communication is realised by sending ASCII string. Each single connections consists of two parts:

- Command sending by PC: this command starts from capital letter, than there are numerical parameters (if necessary) separated by comma
- Answer send from robot to PC. The answer starts from small letter (the same like in command), afterwords there are numerical parameters separated by comma (for example measurements of sensors)

The commands can be classified into two groups:

- command concerned with configuration of robot (set parameters of serial protocol, set parameters of regulator, set sensor parameters)
- command concerned with control of robot (set position, set speed, read measurements of sensors)

Below the set of function for simply programming in MATLAB environment is mentioned:

- kopen ('COMx')- where 'x' is number of port; opens serial port COMx for communitacion with a robot and sets protocol parameters: 115299bps, 8 Data bits, 1 stop bit, no parity, no hardware control. Using of "*ref* = kopen ('COM5')" causes open communication by port COM5 and assign this communication to variable *ref.*
- kclose(*ref*)- function, which closes communication

Functions used to set the configuration of the robot:

- kConfSensor (*ref,n_sens*)- set number of active sonars (default set is only a central one)
- kConfSensEcho(*ref,n_echo*)- set number of sensor echos
- kInitMotors(*ref*)- initialize and reset of DC motors
- kBatteryState(*ref,index*)- get battery state; index- choose the voltage, the current, the absolute remaining capacity, the temperature or relative remaining capacity

Functions used to control robot:

- kAmbient(*ref*)- function returns 11-element vector of measurements from light sensors
- kProximity(*ref*)- function returns 11-element vector of measurements from proximity sensors
- kGetMeasure(*ref,us_numb*)- read distance in [cm] from chosen ultrasonic sensor
- kReadPos(*ref*)- read position from incremental encoders from left and right wheel
- kReadSpeed(*ref*)- read speed from both wheels
- kSetPos(*ref,left,right*)- set counters of position encoders
- kSetSpeed(*ref,left,righ*)- set speed for left and right wheel (control set value by PID controller)
- kSetSpeedProfile(*ref,max_speed,acceleration*)- set the speed and the acceleration for the trapezoidal speed shape of the position controller
- kSetTargetPos(*ref,left,right*)- set the position counter of the two motors. The position is in the pulse, each one corresponds to 0,047 mm
- kSetTargerProfile(*ref,left,right*)- set a positon to be reached. The move will be performed with three phase, a acceleration to reach the maximum speed and a deceleration phase before the finish position
- kSetPWM(*ref,left,right*)-set speed for motors, without speed controller
- kStop(*ref*)- stop robot
- kSetPosPID(*ref,kp,ki,kd*)- set kp,ki,kd parameters for position controller
- kSetSpeedPID(*ref,kp,ki,kd*)- set kp,ki,kd parameters for speed controller

Exemplary commands: kopen and kSetTargetPos in MATLAB is shown below:

```
function [ref]=kopen(p)
if p==1
ref=serial('COM5','BaudRate',115200,'DataBits',8,'Sto
pBits',1,'FlowControl','none');
else p==2
ref=serial('COM6','BaudRate',115200,'DataBits',8,'Sto
pBits',1,'FlowControl','none');
end
fopen(ref);

function r=kSetTargetPos(ref,left,right)
cmd=strcat('P',',','l',num2str(left),',','l',num2str(
right));
fprintf(ref,cmd);
v = fscanf(ref);
value = sscanf(v,' %s');
if value == 'p'
     r = 0;
else
     r = -1;
end
```

The crux of communication between PC and robot is sending property ASCII string. Written command in MATLAB make this communication easy for beginning students. Our library enable easy implementing and testing various path planning algorithm for mobile robots. It is possible generate any workspace using movable obstacles in our laboratory. So as, students can easily test own path planning algorithms. Written software was also used to test repeatability measurements from proximity and ultrasonic sensors. Fig. 3 describes measurement from 5 sonars for non-moving robot. The obstacles are in 25 cm distance from robot. Fig. 4 describes the same situation, but distance between robot and obstacles is 50cm. The experiments show good repeatability measurements. Fig. 5 describes measurements of distance (by central sonar) during motion of robot with constant speed. On the diagram measurement of the cental sonar was shown. Initial distance between robot and obstacle was 130cm, the stop condition was 20cm. Fig. 6 shows relation between measurement and distance for proximity sensors. These sensors return measurement as a 12-bit value. The greatest value is near the obstacle.

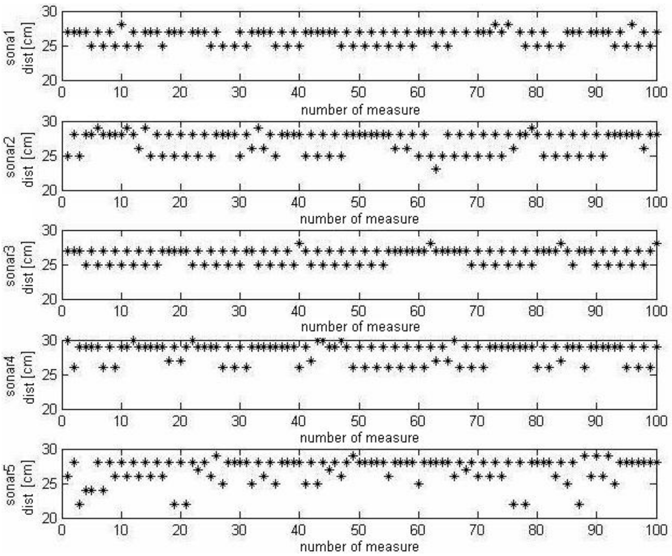


Figure 3. Value of measurements for each sonar; obstacle in distance 25cm

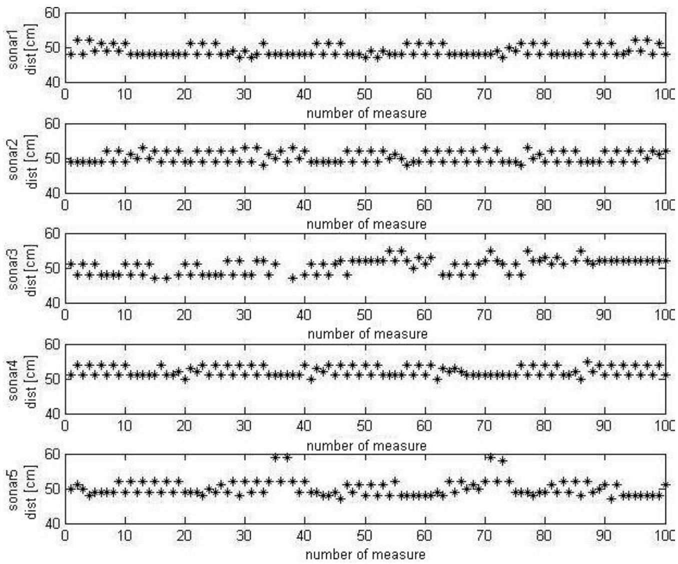


Figure 4. Value of measurements for each sonar; obstacle in distance 50cm

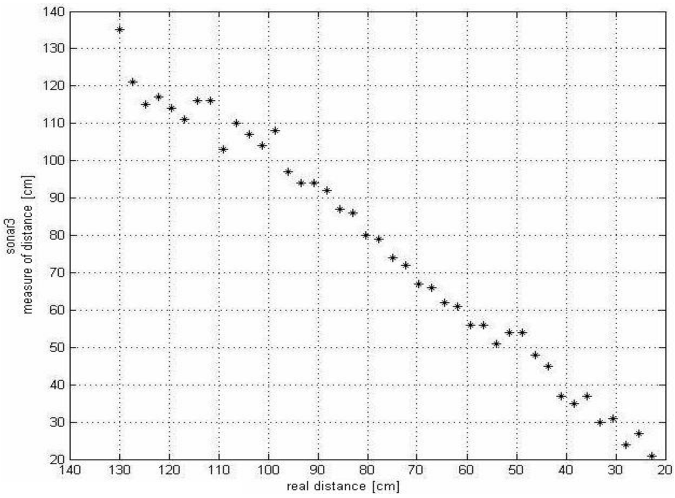


Figure 5. Value of measurement for central sonar during motion with constant speed

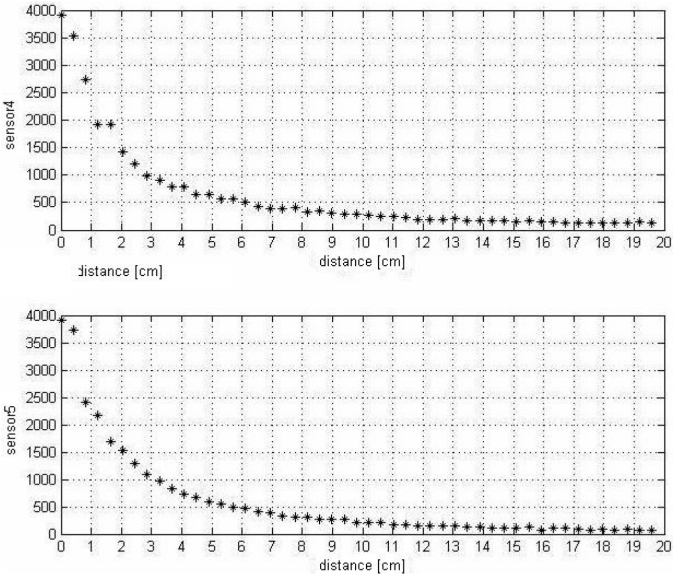


Figure 6. Value of measurement of proximity sensors for smaller distance between robot and obstacle

IV. SUMMARY

In this paper special library of function to communication with Khepera III robot was presented. This library enable easy writing m-file for Khepera. Commands for communication and control robot are very intuitive. Because of good knowledge of MATLAB environment among our students this library is very helpful. Students can concentrate on path planning algorithms and do not waste time on difficult problem with communication by serial port with robot.

REFERENCES

[1] Garbacz M., Laboratoryjny robot mobilny Khepera II. Automatyki (polrocznik AGH) t.9, z.3, 2005, 393-400.

[2] Midas Components Ltd.- Air Ultrasonic Ceramic Transducers 400ST/R100

[3] Vishay- Reflective Optical Sensor with Transistor Output TCRT5000(L)

[4] Khepera III, User Manual, K-Team S.A, 2007

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Ackerman Steering Chassis with Independently Driven Back Wheels

Tomáš Ripel
Brno University of Technology
Faculty of Mechanical Engineering
Brno, Czech Republic
tripel@centrum.cz

Jan Hrbáček
Brno University of Technology
Faculty of Mechanical Engineering
Brno, Czech Republic
jan@hrbacek.info

Jiří Krejsa
Brno University of Technology
Faculty of Mechanical Engineering
Brno, Czech Republic
krejsa@fme.vutbr.cz

Abstract— **When building the robot for outdoor competition the designer can choose from a variety of chassis. The differential drive and Ackerman steering drive are among the most popular. The paper describes our experiences with the custom built chassis that uses Ackerman steering principle together with independently driven back wheels. The chassis represent the base of Bender II mobile robot, used in Robotour 2009 outdoor competition. The goal of such solution is to obtain better traction on rough outdoor surfaces while keeping the mechanical design simple.**

Mobile robot, chassis concept, software differential

I. INTRODUCTION

Mobile robots can be seen more and more often these days. And not only the sophisticated scientific instruments, such as the Spirit and Opportunity rovers, that are moving on the surface of Mars since 2004. It is only a matter of time when autonomous robots become an ordinary part of our lives. Robotic contests play important role in speeding up the development of reliable robots both regarding the mechanical/electrical components and sophisticated control and navigation algorithms. Robotour competition is one such contest that enables smaller robots to compete, thus bringing into the design process the student teams.

The design of autonomous mobile vehicle is a sophisticated task to solve and construction of mechanical parts belongs among the most important parts the design stage. The selection of the type of the chassis used in the robot is the essential in the whole construction concept. This paper describes the approach used in Bender II mobile robot, used in Robotour 2007 and 2009 competitions. Bender II was designed mostly by the bachelor students.

Several chassis concepts are suitable for mobile robots [1] and many aspects have to be considered during its selection. Among the most important issues we can count robot utilization, energy convenience and environment, where the robot operates. Differential drive system [3] is the most common for its good maneuverability and it´s also very suitable for terrain irregularities. It´s construction is simple. However in case of more than two driven wheels configuration appears troubles caused by slippage wheels on the surface during wheeling, therefore the main disadvantage is its low efficiency. That leads to necessity to overdesign the actuators. The omnidirectional chassis has more efficient motion, but there are still huge losses from friction caused by fact, that not all wheels rotate in the direction of movement. It also requires flat and hard surface with no obstacles for its optimal utilization. The biggest advantage of omnidirectional type of chassis is ability to move holonomically which means that it can instantaneously change direction. This is utilized by wheels construction which has many varieties such as roller [4,5], Mecanum [6,7] or spherical wheels [8,9] are. Hybrid chassis with an independent rotation of all wheels behaves very well in uneven terrain, but because all wheels have to be powered and driven separately, it´s very inconvenient from the energetic point of view and also complicated to control.

Ackerman chassis is the most common type used especially in automotive industry, where it represents major share of applied chassis. This concept provides very efficient motion which is for mobile robots essential. The remaining issue is the possible loss of traction when mechanical differential is used. This paper describes our modification of classical Ackerman chassis in order to keep its advantages and resolve the drawbacks.

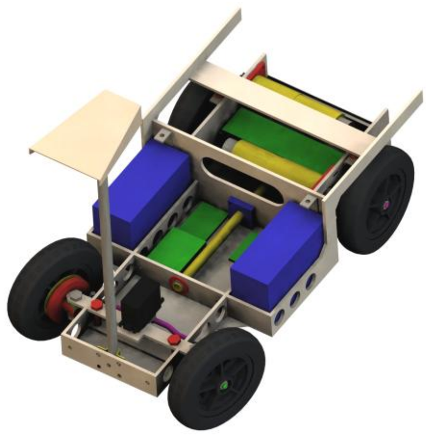


Figure 1. Bender II mobile robot 3D model

II. The Chassis Concept

Bender II is a four-wheeled mobile robot of medium size. It was designed as a testing platform for fusion of sensors and for robotic outdoor competitions at the Institute of Solid Mechanics, Mechatronics and Biomechanics, Faculty of Mechanical Engineering, Brno University of Technology. Platform total weight is 25 kg, and payload is 7 kg. It´s 600 mm long and 300 mm wide with 50 mm road clearance. Wheels are inflatable of 160 mm diameter. The robot was first modeled in SolidWorks, see the 3D CAD model in Figure 1.

In Bender II chassis the Ackermann steering is combined with independent rear drives. The concept of Ackerman steering was chosen as an effort to design mechanical platform comparable with real vehicle, while the independent rear drives were proposed to improve the traction in uneven surfaces the robot was aimed to operate in. Although the whole construction is simplified by the absence of suspension, robot´s behavior is similar to classical automobile motion. Because rear wheels are driven separately, there is no necessity to use mechanical differential. The function of mechanical differential is substituted by driving algorithm that controls individual motors in accordance with the steering angle of front wheels.

A. The swinging rear axle

The most significant mechanical change contrary to classical automotive chassis is the application of swinging axle. This solution partly compensates suspension and partly solves the problems with required loading capacity which would be problematic in irregular terrain using rigid chassis. Swinging axle is usually applied in trucks for its good mechanical characteristics. Due to this simple, practical and efficient mechanical concept the traction of rear wheels is ensured.

For its realization it is necessary to divide rear axle from the rest of frame. In this particular solution the whole frame is divided into two parts. The rear part contains the drive units and chain drives transmitting torsional moment to rear wheels. Front part represents the main construction, where all others components are placed. Those two parts are connected by torsional shaft which allows the parts to swing around each other. The swinging is mechanically limited so the maximal angle is about 10°. This way the constant contact between rear wheel and surface is ensured even in cases when the platform has to deal with heavy loads (laser scanner and other sensory equipment). Figure 2. shows how the swinging axle works.

Rear units which independently drive wheels are controlled by master control system. This system sets different velocities for each wheel during steering in dependence on actual angle a speed of robot. This way is absence of mechanical differential solved.

The drive units consists of Maxon RE40 DC motor with a 3 staged 43:1 planetary gearhead GP 42C. Chain transmission between drives and wheels has the ratio of 1.5:1. The motors are controlled by speed controllers, which were specially designed for the Bender II requirements. It communicates with the master computer via the shared RS-485 bus. The drive units together provide the power of 300W and maximal torque of 30Nm. The power supplies are represented by two lead-acid accumulators 12V/7 Ah. Their theoretical capacity is 168Wh.

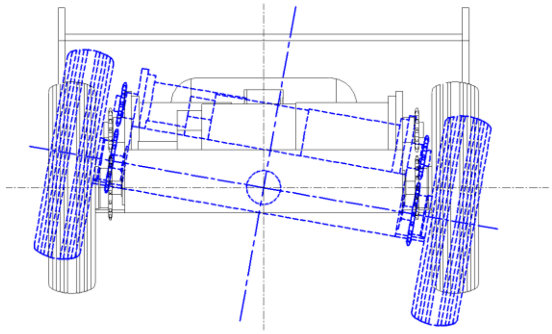


Figure 2. Swinging rear axle of the Bender II mobile robot

B. Ackermann streering

Ackermann steering (which is also known as kingpin steering) ensures proper angle of the front wheels during the robot wheeling. Each wheel has to be turned in a different angle, because each follows different radius. The inner wheel is tilted more, than the outer wheel. This condition is ensured by the geometry of the mechanism. This principle is useful especially at high speeds, because it reduces tire slippage. Although the Bender II is designed for slow speeds (up to 5km/h), the slippage effect (even infinitesimal) is undesirable because of the front incremental sensors. The steering mechanism is realized by double pivoting system. The pivots are at precise angles, so the imaginary axis passes the kingpin center, end of pivots and the center of rear axle (as shown on Fig. 3). The construction solution has to assure possibility to detent the static toe-in. This is realized by threaded rod.

These diagrams show how the Ackermann steering and swinging axle works. The pivot on Fig. 3 is placed in the center of the front wheel which doesn´t correspond with reality. From construction point of view it is almost impossible to achieve this pivot´s placement. The Bender II front wheels center of rotation is placed 50 mm off the wheel center which influences required torque of drive unit operating the steering mechanism.

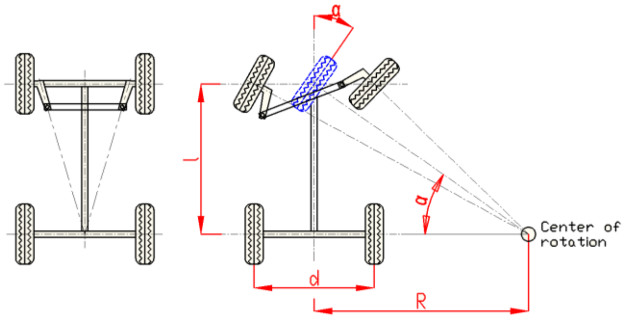


Figure 3. Ackermann steering principle

III. Mechanical Design

The construction of robot chassis is a complex issue, which requires knowledge about its functions and purpose. Mechanical design of the robot Bender II is simple, robust and efficient. This chapter describes a solid construction of the main parts as frame, steering mechanism and rear swinging axle are.

A. Frame design

Optimal frame for mobile robot has to be rigid and lightweight at the same time, which can be assured by proper material selection and mechanical design. Bender II has welded aluminum frame divided into two parts. Both parts are assembled from aluminum bars and welded by TIG method. Front frame represents the main body of the robot. Its shape provides optimal placement of accumulators and other heavy parts (laser scanner) between axles. This concept ensures balanced loading to each axle, which is suitable for driving properties. However, little overload of the rear axle is desirable, because of better traction. Shape of the front frame is shown on Figure 4.

Figure 4. Front frame of the Bender II mobile robot

In the rear frame the drive units with controllers are attached. Canals in sidewalls ensure movable bearing of drive units to allow its shift in case of necessity to tense the chain. Axle driving shafts are embedded in the middle of rear frame in ball bearings.

Figure 5. Rear frame of the Bender II mobile robot

The frame is a slightly overdesigned. The bars it consists of have size 5x50 mm, which provides surface big enough to attach additional device in the case of need. To reduce the weight of mechanical construction the holes are drilled all over the frame.

B. Steering design

The mechanism of steering is controlled by actuator usually used in RC models. It has torque of 1.5Nm. Front axle has two parts. The lower part is closely connected with the frame and the upper part is connected with it by bolts. The body of half-axis is embedded between two axial ball bearings so it can rotate around steering knuckle. The rigidity of the steering mechanism can be controlled by the nut which is bonding upper front axis. As was already discussed in previous chapter the static toe-in can be controlled by threaded rod which connects draw rod with the body of half-axis. All parts of the steering mechanism are made from the steel for its durability.

C. Swinging axle design

The rear axle is connected to the front part by the shaft which allows swinging those parts relatively to each other. The shaft is actually a tube on which the radial ball bearings are pressed and it is fixed in the rear axle. Bearing housing is bolted in the front frame. The swinging shaft is embedded in the front frame and connected with it by retained ring. The main disadvantage of this mechanical solution is the fact that it longitudinally divides construction which makes difficulties with drive units bedding and space in the middle of robot. Swinging axle mechanism is easily demountable. The design of the swinging axle becomes clear on Fig. 6.

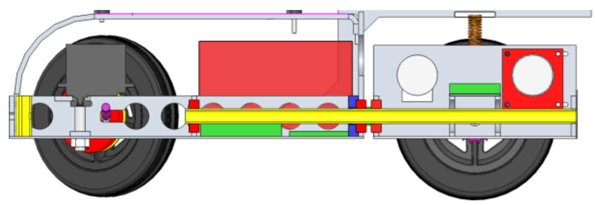


Figure 6. longitudinal cut through swinging axle

IV. SOFTWARE DIFFERENTIAL

Utilization of the software differential was mentioned in the previous chapter. Its algorithm controls independently the angular speeds of the driving motors and thus the rear wheels. The individual angular speeds of the wheels must be related with steering angle α. In order to determine the velocities, first the angular speed of virtual motor must be introduced, representing the movement of the robot:

$$\omega = \frac{v}{r} \cdot i \tag{1}$$

where v denotes forward speed, r is wheel radius and i is the total gear ratio between the motor and the wheel. The relation between the steering angle α and the curve radius R is illustrated on Fig. 3:

$$R = \frac{l}{\tan \alpha} \tag{2}$$

where l is the wheel base of the chassis. During the circular motion the dependence between tangential speed of a point and the distance to the center of the circle is linear. Such a distance, denoted R, is in the middle of rear axle, $R + d/2$ for the left wheel and $R - d/2$ for the right wheel (Figure 3). From this knowledge it is possible to express the relation between the tangential speed of left wheel and center of rear axle:

$$\frac{v_L}{v} = \frac{R + d/2}{R} \tag{3}$$

and similarly for the right wheel:

$$\frac{v_R}{v} = \frac{R - d/2}{R} \tag{4}$$

Tangential velocity is useful for good picture about robot´s speed, but for circular movement of wheel the angular speed is necessary. This is simply solved by multiplying the equations by v and i and dividing them by r. Final equations describe dependency between wheel spacing d and the current curve radius R:

$$\omega_L = \omega \cdot \left(1 + \frac{d}{2 \cdot R}\right) \quad (5)$$

$$\omega_R = \omega \cdot \left(1 - \frac{d}{2 \cdot R}\right) \quad (6)$$

where ω_L (ω_R) is the left (right) motor angular speed.

Quantities used in this paper follows a simple convention – turning to the right and forward movement implicates positive sign of the value, turning to the left and reversing is represented by negative sign.

A. Implementation of the software differential

The software architecture of the Bender II mobile robot follows a hierarchical scheme. The lowest level is formed by individual hardware devices, mostly interconnected by a shared RS-485 bus. The communication on the bus is controlled by a single master (the main computer) and makes use of a custom protocol. The exchanged data are encoded into packets of variable-length. The protocol features reliable delivery by using ACK/NACK response messages containing CRC consistency check result and automatic packet resending in case of ACK message timeout or NACK message reception.

The functionality of each hardware device is wrapped by appropriate low-level software module providing thread-safe access.

As a part of the middle-level software there is a module *Motion* that lies on the top of the low-level modules (individual hardware device interfaces). This module is an entry point for the higher software layers to control the motion of the robot. It encapsulates the software differential and provides two public methods – method `Go(speed, direction)` used to drive the robot at the desired speed to the desired direction and method `Stop()` equivalent to call `Go(0, anything)` to halt the robot.

The internal structure of the SW differential code follows the equations presented previously. The algorithm firstly converts the desired robot speed to the angular speed of a virtual centered motor. Then the desired curve radius according to (2) for a non-zero direction angle is computed (zero has the meaning of a straight movement and matches an infinite curve radius).

The next step is already to calculate the individual motor angular speeds according to (5) and (6). This can be done only when a non-infinite value of the curve radius is provided. Otherwise the differential algorithm is skipped and both wheels are driven at equal angular speeds.

The *Motion* module has now all the information to order the steering servomotor and the drive units to set the currently computed values. To conserve the shared communication bus bandwidth, *Motion* sends the command to each hardware unit only in case that the newly computed value differs from the previous one. The procedure described above repeats every time the upper software layers decide to change desired speed or direction of the movement.

V. HIGH LEVEL CONTROL

While the main aim of the paper is to describe the chassis, short description of the higher level control mechanism illustrate what structures were used during the competition. The overall scheme of high level control is shown on figure 7. Basically the action of the robot actuators must be selected properly based on all the sensory information, internal robot state, knowledge regarding the environment (no matter whether the knowledge is gained during the travel or inserted into the system prior to its mission) and goal definition. The actions are either in the form of general velocities – rotational and translational, that are further trasnferred depending on the type of the chasis; or actions are in the form directly linked to the chassis, in our case the steering angle and the translational velocity. Actions are further processed in the controllers, in Bender II case in the *Motion* module described in previous chapter.

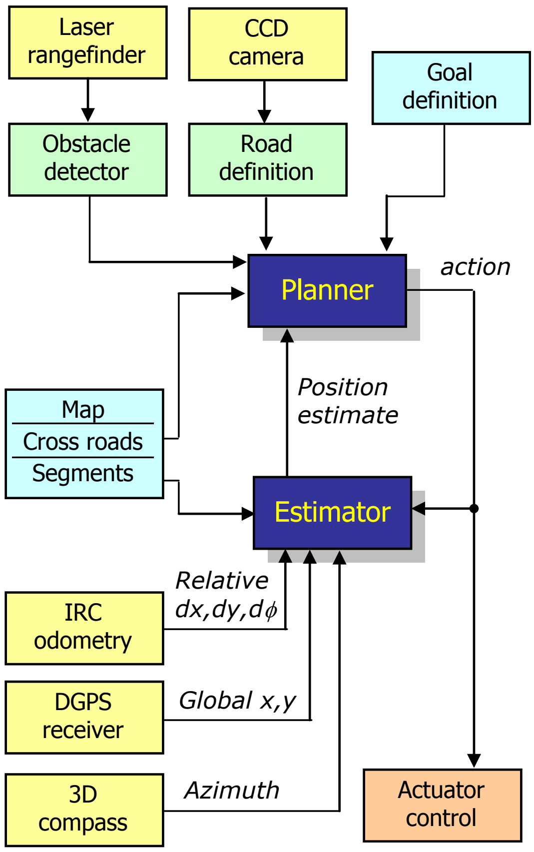


Figure 7. High level control scheme and signal flow

The action is produced by the planner based primarily on the sensory inputs and the estimate of robot position. The estimate is calculated by nonlinear version of Kalman filter. Some of the sensory inputs are fed into the estimator, processed and robots pose is extracted, while other sensory information are fed directly to the planner as the information from those sensors can not be used to determine robots pose, but are useful for planning. Lets first look at the estimator.

The estimator keeps the estimate of robots pose as an internal state. The state is changed by applying the action produced by the planner. Such a change is produced by the robot motion model. The information from IRC sensors on front wheels is used as an input to the motion model (even if strictly speaking such information is a measurement). The predicted state is further corrected using the measurements from sensor capable of giving the global position information.

In our case the xy coordinates are taken from the differential GPS receiver and heading angle is taken from 3D compass with the compensation of the mount plane inclination towards the ground plane. The particular sensor units used on Bender II were: as GPS receiver the custom built device based on the Lassen IQ module, digital compass module based on Honeywell HMC6343.

Estimator result is just one of the inputs to the planner. The main sensory input that keeps the robot on the path is the road description obtained from the processing of the images acquired by the camera mounted on the robot. Using preprocessing followed by the image segmentation and road description extraction the information about the local road is obtained and fed into the planner.

Image processing is not used for obstacle detection, the data from laser rangefinder by SICK is used instead. Such data are pre-processed by obstacle detector giving the planner information about avoidability of detected obstacle. Currently the USB2 Wide Angle Webcam Live WB-6200p is used, however the Pixelink family cameras are being tested as it provides high quality images mainly due to the high end optics.

The environment related information is used both in the estimator and the planner, the currently run segment of the map is extracted from the estimate while crossroad information are used by the planner. The goal of the whole mission is taken into account when globally planning the sequence of road segments taken from the environment map.

VI. EXPERIMENTAL EVALUATION

Robot Bender II has proved good behavior in both outdoor and indoor environment. Outdoor tests were performed on various surfaces like park footpaths, cobblestones and asphalt or sandy paths. During indoor tests robot has proved good mobility in spite of limited maneuverability given by the type of chassis selected.

A. Indoor test results

Very important test for mobile robot is its *power drain*. Bender II has a lot of electronic equipment on board which affects duration of its autonomous activity. This test was performed on flat surface with good traction properties. The TABLE I. contains the results of the test in different operational modes of robot. The term “on-board electronics” means the minimal configuration needed to drive the robot (drive controllers, steering control, wheel encoders, bus master unit and the main computer), not counting the power consumption of payload electronics.

TABLE I. ROBOT CONSUMPTION UNDER VARIOUS CONDITIONS

Regime of operation	***Average consumption [W]***
On-board electronics powered, no motion	33
Uniform motion at speed of 0.1 ms^{-1}	55
Uniform motion at speed of 0.2 ms^{-1}	60
Uniform motion at speed of 0.3 ms^{-1}	80
Uniform motion at speed of 0.3 ms^{-1}, 4° grade	120

On-board electronics of robot include beside basic electronics (approximately 33 W) also another major consumer, which is the SICK laser measurement scanner (LMS291). This device drains in operation roughly 30 W. With other equipment on-board like WiFi access point and the GPRS modem are, the total static consumption rises up to 70 W in average.

Another test performed on the robot was checking of stability. Good stability is necessary for its optimal behavior in terrain. The robot was tested with all possible equipment onboard because of its influence to the center of gravity. The test results confirmed this expectation; fully loaded chassis has very good stability for desired utilization.

More important than the static tests was dynamic stability trials. As was mentioned previously, the chassis has no dumping except the one provided by inflatable tires. Irregular terrain was simulated by obstacle (rectangular 4x8 cm cross-section) placed on flat floor. The height of the used obstacle determines theoretical maximal static tilt of the robot (i.e. vertical axis angular deviation when one front wheel is on the top of the obstacle) to value of 5.3°. The robot was overcoming the obstacle from different angles at various speeds and its response was observed. The result is that even in the case of the worst obstacle shape and position and relatively high speed, the fully loaded chassis embodies reasonable stability margin for reliable operation in target environments.

B. Outdoor tests results

This subsection describes robots behavior in outdoor environment for which it has been designed primarily.

Generally the one of the most important parameters for mobile robots is their *operating range*. The accumulators provide energy for all electronics onboard the robot including motors. Operating range is affected by consumption of each device. While robots movement is in non-traction mode, its energy demands are independent on running speed, from which we know, that the bigger speed the robot is moving, the bigger part of battery energy can be saved for traction. This fact was observed during the drive algorithms development. When at the beginning the speed was set to very low values (because of the safety reasons), the robot has reached the distance from start (without replacing batteries) of about 900meters. After speed increase to approximately 0.8 ms^{-1} the robot reached more than 1.6 km distance from the start. This value was found sufficient.

The traction was observed on a variety of outdoor surfaces during enormous number of tests, mostly in Lužánky city park, where Robotour 2009 competition took place. During the tests we did not encounter a single problem with the traction and independent rear wheel drives proved its advantages mainly on sandy surfaces.

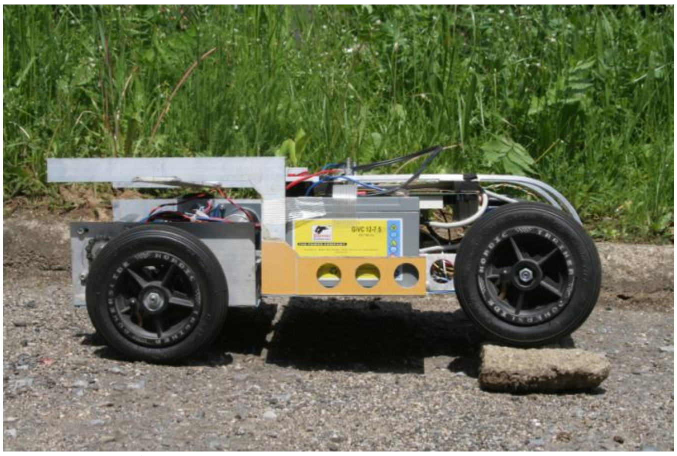


Figure 8. Swinging rear axle in action (side view)

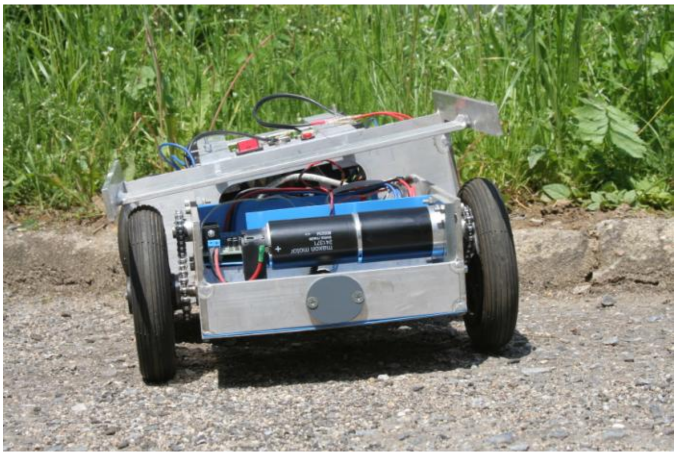


Figure 9. Swinging rear axle in action (rear view)

C. Overall experiences

It has been found that the chassis of our robot behaves well under various conditions. The independent rear wheel drives system does not suffer from the complete loss of traction in case of one wheel slippage, as both drive units hold their preset speeds independently. The swinging rear axle is capable of compensating terrain unevenness. By combination of these two major construction units, the robot is able to drive through surprisingly hard terrains and still behave well and economically on the road. Figures 8 and 9 show robots behavior in terrain.

While the ratio of the rear wheel angular speeds is dependent only on the steering angle and not on the unstable adhesion of individual wheels, the robot is not prone to under- or oversteering – the Ackermann steering is supported by the rear wheel speeds ratio.

The robot achieves forward speed of approximately 1 ms^{-1} (limited mainly by maximal input angular speed of the gearhead). Compared to an average light truck (that is likely to be robotized and autonomously operated in relatively near future e.g. in military supply service), our robot is scaled-down by factor of approximately 1:10. In this context, the speed of 1 ms^{-1} matches 10 ms^{-1} or 36 kmh^{-1} of a full-size vehicle. It may not seem to be very high, but the speed of an autonomous vehicle in an unknown and unpredictable terrain cannot be much higher in order to maintain both vehicle and environment safety. The only disadvantage of the presented independent drive units architecture (apart from obvious problems arising from using two motors, two gears and two controllers instead of one) is that the chassis becomes less controllable in case of misbehavior of one of the drive units. This happened at the beginning of the testing and was caused by communication problems, that be avoided by using a single controller driving both traction motors, when a possible communication failure would not cause discrepancy in wheel angular speeds.

VII. CONCLUCIONS

Autonomous mobile robot Bender II competed on Robotour 2007 in Prague and Robotour 2009 in Brno. The design of the robot, described in this paper in detail did not encounter any problem during the competition. The combination of Ackerman steering with independently driven back wheels and swinging rear axle exhibited outstanding behavior on a number of surfaces while keeping the energy consumption on acceptable levels. Therefore we can recommend this type of chassis to be used in robots designed for similar purposes.

The robot has been partially designed in the frame of several student projects. The students actively participated on the whole process beyond standard student´s work, resulting in higher motivation in further study process. The robot serves as educational tool for various tasks of dynamics, kinematics, electronics and data fusion.

ACKNOWLEDGMENT

Published results were acquired acquired with the support of the Ministry of Education, Youth and Sports of the Czech Republic, research plan MSM 0021630518 “Simulation modeling of mechatronic systems”.

REFERENCES

[1] R. Siegwart, I. Nourbakhsh, “Introduction to Autonomous Mobile Robots”, *MIT Press*, 2004

[2] H. Choset, et al., “Principles of Robot Motion", MIT Press, 2005.

[3] G. Campion, G. Bastin, B. d’Andrea-Novel, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots”, *IEEE Trans. Robot. Autom. 12*, pp.47-62, 1996.

[4] Fujisawa, S., Ohkubo, K., Yoshida, T., Satonaka, N., Shidama, Y., and Yamaura, H., “Improved Moving Properties of an Omnidirectional Vehicle Using Stepping Motor”, *Proc of the 36th Conf on Decision & Control. San Diego, California*, pp.3654-7, 1997.

[5] Williams, R., Carter, B., Gallina, P., and Rosati, G., “Wheeled Omni-directional Robot Dynamics Including Slip,” *Proc of 2002 ASME Design Engineering Technical Conferences*, Sep 2002.

[6] J. Salih, M. Rizon, S. Yaacob, A. Adom, M. Mamat, “Designing Omni-Directional Mobile Robot with Mecanum Wheel", *American Journal of Applied Sciences 3 (5)*, pp.1831-1835, 2006.

[7] Bradley A., Miller, S., Creary, G., Miller, N., Begley, M., Misch, N., “Mobius, an Omnidirectional Robot Utilizing Mecanum Wheels and Fuzzy Logic Control,” *Proc of the 28th Annual AAS Rocky Mountain Guidance and Control Conferences*, pp. 251-266, 2005.

[8] Ferriere L., Raucent B., “ROLLMOBS, a New Universal Wheel Concept,” *Proc of 1998 IEEE ICRA*, pp. 1877-82, May 1998.

[9] West, A.M., and Asada, H., “Design of Ball Wheel Mechanisms for Omnidirectional Vehicles with Full Mobility and Invariant Kinematics,” *ASME Journal of Mechanical Design*, 117, 1995.

Design of low-cost robotic arm for education

Filip Tóth
Faculty of El. Engineering and Information Technology
Slovak University of Technology
Bratislava, Slovak Republic
ftxrobot@hotmail.com

Boris Rohaľ-Ilkiv, Pavol Krasňanský
Faculty of Mechanical Engineering
Slovak University of Technology
Bratislava, Slovak Republic
boris.rohal-ilkiv@stuba.sk

Abstract – **The goal of this project was to design and implement the electronics for a robotic device – namely a robotic manipulator. The manipulator has a total of six degrees of freedom, three of which are on the manipulator itself, and the rest are in the wrist. A gripper is used as the end stage of the device. It was necessary to design the electronics so that they could be interfaced with standard hardware and software as that which is available in the automation labs of STU. A similar robotic manipulator will be used as a part of a mobile robot in the future.**

Keywords: robotic manipulator; degrees of freedom; electronics

I. Introduction

Pavol Krasnansky, a student of the Faculty of Mechanical Engineering of STU is the author of the original idea. I was given an opportunity to create the electronics of this device, which I took. The development of the electronics for the device has been going on for a total of three years so far. The goal of the project is to be able to control and solve direct and inverse kinematic problems via MATLAB.

II. The mechanics

The whole arm (fig. 1.) is connected onto a base, in which the drive and the mechanics for the first degree of freedom is mounted in. The second, third and fourth degrees are implemented inside the arm itself. The fourth degree of freedom is the gripper's wrist. The fifth degree is the rotation of both the gripper and the held instrument. As for the gripper itself, it is made up of two opposing "fingers", which move against each other symmetrically – they are the sixth degree of freedom.

A classical serial kinematic structure requires a fair amount of power from the motors – this is why the joints of the manipulator are driven through a worm drive gear arrangement. This provides great strength with relatively small motors, a zero backlash and, thanks to the self locking properties of this arrangement, zero power consumption while not moving. The disadvantage of this solution is the relatively low speed of the system.

Each joint (fig. 2.) has its own servomotor, an absolute rotary encoder, two limit switches for the maximal and minimal angles and the worm drive arrangement.

The highest requirements for torque are on the second degree – it is a joint that has to move the greatest weight (effectively the whole weight of the manipulator as well as any objects held by the hand). Also, the arm acts like a lever force which increases the torque requirements in this joint. Therefore a motor with a higher torque than anywhere else was used here.

The gripper itself has two fingers which create a parallel kinematic structure – thanks to this; the fingers are always parallel to each other. There are pressure sensors in the area in which the gripper makes contact with the gripped object. The total length of the manipulator (when set to a straight alignment) is 580mm. It has a total weight of 5075g. The actuators used are two types of servomotors (see table 1). In the second degree, an industrial TONEGAVA SEIKO servomotor with steel gears is used. In the rest of the degrees, Hitec HS-5955TG servomotors (professional RC modeling servomotor with titanium gears and shaft) are used.

MAB25 absolute rotary encoders are used for the sensing of the angle between joints. These are connected directly onto the shafts of the joint gears.

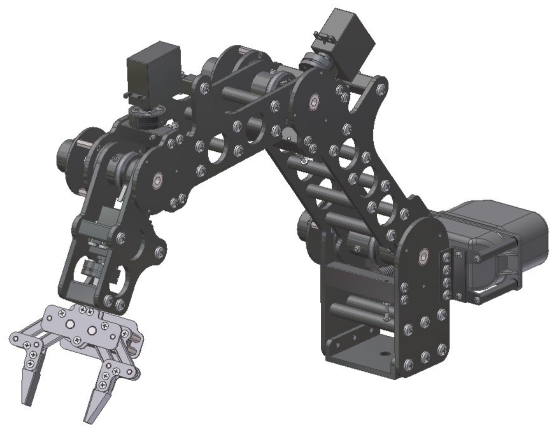


Figure 1. The whole arm

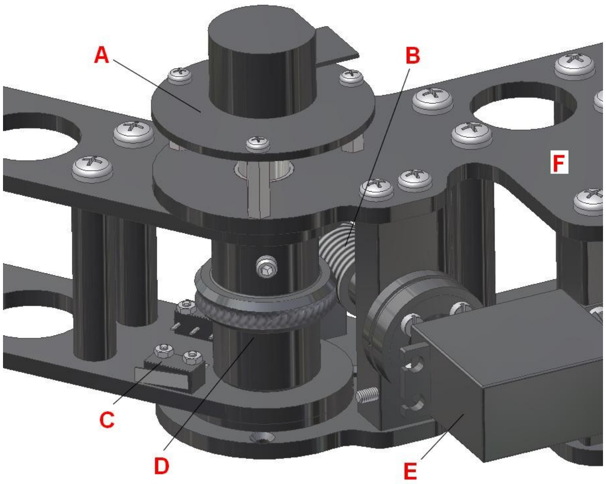


Figure 2. Joint assembly

A – absolute position sensor MAB-25, **B** – screw gear, **C** – safety switch, **D** – screw gear, **E** – servo – drive HS-5955TG, **F** – duraluminium

TABLE I. PARAMETERS OF THE ACTUATORS

	HS-5955TG	PS-050
Gears	titanium	steel
Torgue [kg.cm^{-1}]	31.2	91,5
speed[s / 60°]	0,15	0,29
Dimensions [mm]	40 x 20 x 37	100 x 44 x 79
Voltage [V]	4,8-6	5-12
Weight [g]	62	283,4

III. The electronics

Having electronics as a hobby I spent a great deal of time on its design, and tried to perfect it. I started designing the electronics (fig. 3.) once the mechanical part was almost ready, so there wouldn't be any changes in the electromechanical components of the manipulator (actuators or sensors).

Before the design phase it was necessary to analyze the requirements for this system, clearly define goals and priorities. These goals were set: Modularity, programmability, minimal cabling, robustness, software access to all parameters available, miniaturization, maximal power usage efficiency, connectable with a standard system, digital control of analog inputs/outputs, self-diagnostics capability, and safety.

I would like to point out that the device is completely original. Every part of it is a prototype.

A. The Hitec joint module

The Hitec joint module is a printed circuit board with control electronics, onto which the HS-5955TG servomotor, two limit switches, the absolute rotary encoder, the communication bus and the power for both the logic and power part connects. The board is fixed onto the segment between two degrees of freedom and is connected with the motor of that joint (relatively to the board, the motor is stationary). Since one of the goals of the design was modularity, each degree of freedom has its own module. These modules are connected onto a common bus and have a common power supply. The advantage of such a solution is that only six wires are used to connect all of the modules and enable the function of the whole system. These six signals are: Transmit data, Receive data, System ground, +5V, +7.4V, Power ground. In comparison, a centralized system would require a total of at least 64 cables going from the manipulator onto a central electronics board.

The other modules, such as the Tonegava joint module and the Joint module with grip force measurement are very similar to the basic module, with some minor differences (such as the extra electronics for force sensing).

The heart of the module is an ATMega8 [1] processor – an Atmel RISC processor. In this application I am using the ADC (analog to digital converter), UART interface (standard serial line), ISP (In system programming) interface, the PWM (Pulse width modulation) generator and the GPIO pins (general purpose input output).

A 14.745600MHz crystal is used as the clock source for the device, which is a frequency that allows both maximal computational power as well as precise, whole number settings of timers and the serial line.

To inform the user about the state of the device, three LED diodes are used (red, green and blue). Two of these (the red and green ones [2]) are connected directly to the GPIO pins of the processor, the third one is controlled by an N channel, IRF7341 MOSFET transistor [3].

The servomotor is connected in the classical manner of RC motors – 3 wires, one signal, the rest are for the power supply. The control signal is connected in series with a 1kΩ resistor onto a pin of the processor. The servomotor has been modified to allow continuous rotation of the shaft instead of the normal +/- 90° operation. The motor's speed and direction is controlled via a PWM signal – depending on the width of the signal, the motor will either rotate CW, CCW or halt.

To reduce the effect of the noise created by the power parts, the logic section and power section have their own power, which is not connected to the other.

To enable the measurement of the current the motor draws, a 0.1Ω shunt resistor is used to monitor the current on the motor. By measuring the voltage on this resistor through the ADC, the current being used by the motor can be accurately measured. Also, the ADC is used to monitor the voltage for the control part and the power part of the system. When there is no need for the motor to be turned on, the microprocessor disconnects power from the whole power section to conserve energy. To do this, it uses a P channel MOSFET transistor, namely a Si4435BDY [4]. Because the logic voltage is lower than the power voltage, a P channel transistor cannot operate directly from a GPIO pin of the processor – these transistors are controlled by the voltage between their Gate and their Source pins. The lower the voltage, the higher the resistance. In this case, +5V on the Gate would still result in an open transistor. Therefore an extra transistor is used to control the Gate voltage – an N channel transistor (namely IRF7341 [3]), with a pull-up resistor from the +7.4V. This one is controlled by a GPIO of the processor. This setup controls the main transistor. Using this setup, I have achieved a R_{DS} value of less than 0.02 Ω, which, in the case of a maximum current of 5.2Amps creates a 0.5408W of waste heat, which is within the normal operational parameters of the Si4435. To improve heat dissipation, a heat sink is formed on the PCB, to enable the transfer of heat via the pins of the SO-8 package. When all six degrees are at their maximal power consumption, the total consumption can be as high as 31Amps. Since all the modules are connected into a series via a bus, 31Amps would flow through the first module. This is the reason why there are T connectors used in the latest version – they have a really big contact surface, and are designed for a continuous current of 40A. The other connectors are standard, low current, low voltage, Wire to board connectors.

The absolute position encoder is connected to the module via a 6 signal interface. It is a standard SPI interface, through which the 10 bit information about the current position is transmitted [5]. The encoder is powered by +5V. It is a electromagnetic rotational encoder, type MAB25, made by the MEGATRON company.

The communication interface can be configured to work either as a multiplexed serial line, or it can be used as an I^2C bus (or similar). In the normal configuration the modules communicate with an external device that acts as a master on the bus and gives commands to the modules. A special bus driver for the Transmit signal is used, so the bus acts as an open collector system – this allows a module to transmit and receive without first notifying the other modules. It also allows for a non destructive collision. Each module has its own pull up resistor. The Receive signal is common for all modules – they all receive the commands from the master. The idea of this bus is that the modules are slaves and behave on a "speak when spoken to" basis. The bus driver used is a 74HC125 [6]. There is also the option of using I^2C as the main bus, where SDA (bus data) and SCL (bus clock) take the place of the Receive and Transmit. The advantage of such a configuration would be that the modules might communicate between each other (the I^2C standard allows for multi master communication [7]), or simpler connection to an existing system.

The module also has a RESET button, a jumper usable as a user control (for long term mode changes), a connector for the two limit switches (used as a backup in case of a problem with the absolute encoder or as a simple testing/calibration tool). On each module there's an ISP connector, using which it is possible to program the microcontroller with the appropriate firmware. For programming a proper programmer must be used – various open implementations exist on the internet, as well as commercial products.

This is the third version of the module – since the start of the project, the total board area has decreased by an impressive 57%.

There are mounting holes for M3 screws on the PCB. They are used to fix the module onto the manipulator. Also spacers are used to fix a transparent protective shield above the board.

I designed the PCB in OrCAD. It is a two layer PCB with a soldering mask on both sides.

B. *The Tonegava joint module*

The Tonegeva joint module is almost identical to the Hitec module. The difference lies in the type of servomotor used – which is a PS-050 TONEGAVA SEIKO servomotor. This servo has a separate control part and a separate power part – the control part has its own power as does the power part. The control part connects to the board like a standard servomotor – three pins – +5V, PWM signal and GND. The power section only connects via its power pins - +Vs and GND. Any voltage, ranging from +6V to +12V can be used to power it. All of the other parameters of this module are the same as those of the Hitec module.

C. *The gripper module*

The gripper module is similar to the Hitec module, the main difference being that there is no absolute position encoder but force sensors. These sensors are mounted directly onto the gripper's fingers. The sensors used are FSR-150AS from the German company FSR-Sensoren. The sensor is a resistor, whose resistance changes when force is applied to it. Its small size (12x12x0.5mm) makes it ideal for mounting onto the grippers fingers. Depending on the strength applied to it (from 0.1 to 100N) the resistance changes (from several MΩ to less than 1kΩ). This wide range of output values makes it somewhat problematic to work with. Fortunately, the maximal force of the fingers is 30N, and I've limited the minimal force to 1N. With this I've gained a sensor with an input 1 to 30N range and a 10k to 1kΩ output range. The maximal current through the sensor is 1mA, its temperature drift is 100ppm/°C. The sensor is connected into a simple voltage divider with another resistor. The output from this setup is fed into the ADC of the processor.

D. *The ATMega128 module*

The ATMega128 module is a breakout module for the processor (an ATMega128 [8]) and a few discrete supporting components required for its operation.

The reason for this board is that the processor used is an SMD component and it would be difficult to replace in case of it being damaged. Also, different modules can hold different programs, so software development can be helped by this. All of the pins of the processor are connected to pins. The board can be connected into an appropriate socket.

Aside from the processor and the pins, the only components on the board are a LED and a RESET button. The module is the heart of the communication and HMI module.

E. The communication and HMI module

The communication module (fig. 3.) is a PCB that is not mechanically connected to the manipulator. Its purpose is to provide a “middle man” between the manipulators electronics and the measurement card on the PC. Another task of this board is providing analog inputs outputs for various degrees of freedom using a standard voltage signal (0-10V). The card has 8 such inputs and 8 outputs. Aside from this it has two RS232 serial ports and a special D-Sub connector for connecting the manipulator itself.

There is also a HMI (human machine interface) on this device. The user can control and monitor the robotic arm using buttons, LEDs, a piezoelectric transducer, potentiometers, two encoders and a 4x20 character LCD display.

The main processor used here is the ATMega128, which is mounted on the above mentioned module. All of its pins are used.

The board provides a power supply for both the power part of the manipulator and the logic part. The power supply is heavily filtrated using relatively large capacitors. The power supply is also protected by diodes and fuses. Only the logic power supply is monitored, the power part is monitored and switched by the modules themselves.

The angles of the joints of the manipulator can be set via analog the analog inputs. There are 8 of these inputs. The internal ADC of the ATMega128 processor is used for their measurement. A precise voltage divider, manually adjustable by a precise multi turn trimmer, is used to convert the standard signal (0-10V) to the range of the ADC (0-2.56V). The ADC is a classic 10bit successive approximation ADC with a maximal sample rate of 15kSps.

Unfortunately, the processor has no DAC (digital to analog converter), therefore it cannot create the analog voltage outputs by itself. Therefore, and external DAC was used. The requirements for this DAC were: voltage output, internal reference, at least 10bit resolution, a simple power supply (not symmetrical, preferably from +5V) and a parallel communication bus. In the end I chose the MAX530 DAC – a low power 12 bit DAC, with multiple power and output options. It also has an output buffer and an internal configurable reference - +2.048V, +4.096V, +/-2.048V, or an external reference can be used. It’s quite fast – a settling time of only 25µs. It also has several options for connecting it to a data bus and controlling it – either the 4 bit interface, when the three parts of the 12bit word are written in sequence, or, the 8 bit interface, where 8 and for bits are written in sequence. Aside from the main control bits, there are also control bits, which take care of the conversion and the settings of the DAC. A total of 8 pieces of these DACs are used here. If each one was connected individually to the processor, a total of 112 pins would have to be used. Instead, all of the DACs share a common data bus. Some of the pins are connected onto a 74HCT154 multiplexer, which is addressed by the processor and picks the appropriate DAC to work with. Thanks to this, only 16 GPIO pins of the processor are required to operate all of the 8 DACs.

Since the required output of the device is of the range 0-10V, amplification of the signal was required – all of the DACs outputs had to be put through an operational amplifier. The advantage of an opamp is that its output signal will be stronger than that of the DAC alone, also it will have a higher amplitude. It is simple to set the required voltage gain. The disadvantage is, that it is another block in the analog path of the signal – it can add nonlinear transfers, noise, offsets and such to the signal path and degrade the signal. To avoid additional problems with this I didn’t choose normal opamps, but rather precision opamps, which enable me to set various parameters and fine tune the amplifier stage. Also, it must not need symmetrical power, and have a maximal output voltage of at least 10V. In the end I chose the TLC271CP opamp – a low noise, precision opamp, with a typical noise of 25nV. It also allows the users to choose from three Bias modes – these define key parameters of the opamp and enabled me to choose the best mode for my application. The modes are HIGH, MEDIUM and LOW. Each of them is a compromise between power consumption and suppression of the negative parameters. I chose the HIGH bias mode, which almost nullifies the offset. The tradeoff is a higher power consumption. The power supply for the opamp is +12V. The output of the opamp is properly loaded using a resistor to the ground, and through a protection resistor it is connected onto the 0-10V output of the module.

There are holes for various spacers on the board – these spacers hold both hold the Perspex casing of the board and hold the board above ground.

The PCB is a four layer PCB, with a solder mask, and a description layer (with part description and helpful texts) as well, designed in OrCAD.

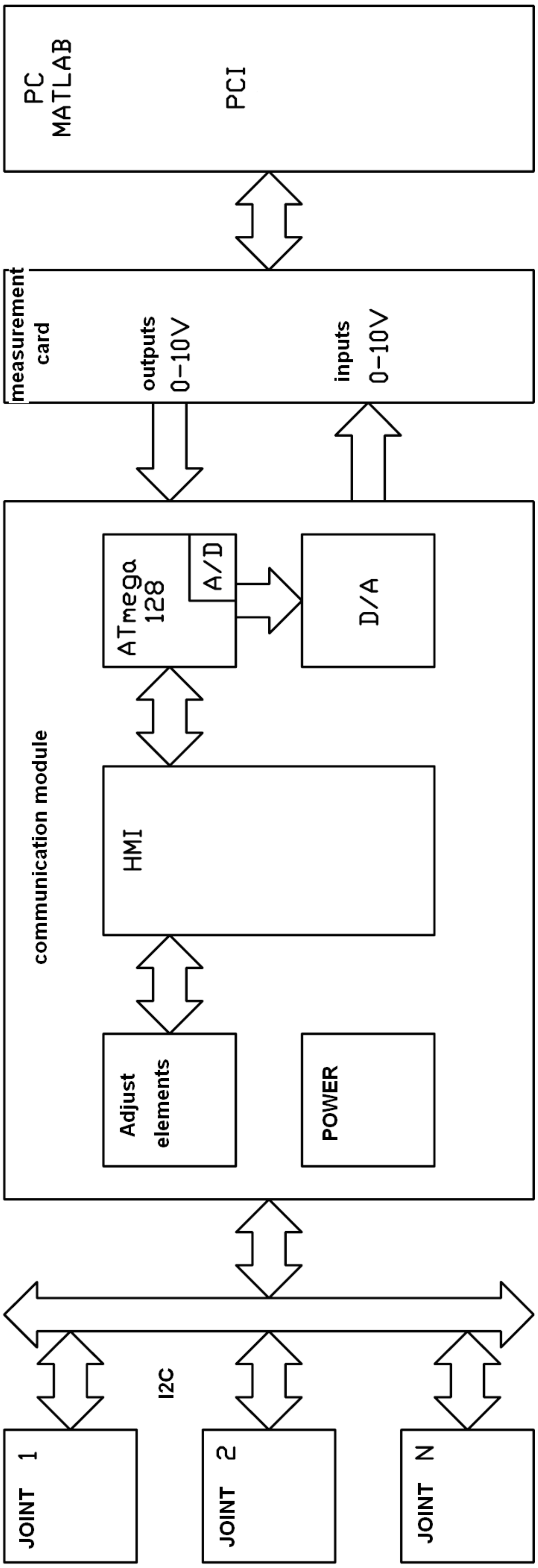


Figure 3. Block diagram of the electronics

IV. Module interconnection

All joint modules are connected parallel – they have a common communication bus and power bus. The connectors used are “Wire to board” for the data bus. It is a 4 pin connector, with the signals Transmit Data, Receive Data, System Ground and +5V. A common problem in robotic systems is how and where to attach the wires to avoid too much mechanical stress. It was necessary to choose the proper type of cable for this application. The best wires turned out to be flexible 0.25mm^2 silicon cables for measurement devices – their core is made out of 128 0.05mm threads.

T type connectors were used for the main power distribution. When dealing with high current, all additional resistance can prove itself to be a problem. The last module in the series will have to deal with a voltage that is down by all of power dissipated on the previous stages. Standard connectors used have a relatively low maximal current rating and relatively high resistance. Professional connectors which are meant for this kind of current have a too massive construction. In the end I chose RC hobbyist connectors – they have the best size-to-resistance ratio. Their resistance is within a few mΩ. Similarly, appropriate cables must be used. For the main power distribution I chose highly flexible PVC cables with an area of 2.5mm^2. Their core is composed of 651 fine wires with a diameter of 0.07mm. The most stressed cables are those connecting the force sensors. The gripper head can be rotated by up to 360°. The sensors are mounted at the end of the gripper head. Since the current won’t rise above 1mA, it was possible to minimize their diameter. I used a highly flexible “LIFY” cable with an area of 0.05mm2. Its core is composed of 26 threads with a diameter of 0.05mm.

A special hybrid D-Sub connector is used to connect the manipulator itself and the communications module. This connector has 17 signal pins and 7 high current pins. The high current pins are designed to withstand a continuous current of 40A. They are gold plated with a 0.8 micron layer. The size of the connector is identical to a classical 50pin D-Sub connector.

On the communications module there are sockets for classical, 4mm banana plugs. This type of connection is common in labs and is very practical.

V. Design, manufacture and testing

I did this design based on the experience I got from other designing other robotic systems and on the application suggestions from the manufacturers. I tested out most of design on a bread board first. When the design did not meet my requirements I looked for a better solution.

The main design tool I used was Cadence OrCAD – a software package for the design of printed circuit boards and electronic schematics.

Most of the PCBs used have two layers. The communications module was made with 4 layers. All of the parts were soldered

manually. After assembly, each module was tested individually and loaded with 100% of the designed load capacity for a few hours.
The Perspex protection covers were designed in AutoCAD – a popular CAD system for mechanical drawing. The output was processed into a program for a CNC machine which cut out all of the components.

VI. Education

The robotic manipulator is currently being used by students to solve basic robotic kinematic problems, direct and inverse kinematic problems as well as problems involving the calibration of the kinematic structure of the manipulator. Students solve these problems in the form of individual assignments, using AVRStudio, MATLAB/Simulink and utilize the connection of the manipulator to a computer. After completing these basic tasks, groups of students are faced with more challenging tasks – such having the end point of the manipulator follow a predetermined path.

Assessment

Currently, the mechanical part of the manipulator is undergoing reconstruction. So far, all of the components have been only tested by themselves. When the mechanics are completed, the electronics will be mounted onto it, which will animate the so far static manipulator. After this, it will be necessary to fine-tune all of the components together. After that, the programming of the controlling microprocessors may commence and the creation of the base algorithms for controlling the manipulator. Then, finally, it will be possible to use the manipulator to solve and simulate kinematic problems and try and solve other similar control problems. MATLAB doesn't need to be the tool used, since many other software packages can use the DAQ card.
Thanks to the experience gained on this project, it is probable that a new version of the manipulator will be developed, with better dynamics, lower weight. This manipulator will be mounted onto a mobile platform (with special mecanum wheels and a powerful computer), which can be used to autonomously explore an area or do similar tasks.

References

[1] Atmel Corporation: 8-bit AVR with 8K Bites In-System Programmable Flash , [online] Publication in May 2008, http://www.atmel.com/dyn /resources/prod_documents/2486s.pdf

[2] Kingbright Corporation: Surface mount LED lamps, [online] Publication in 2008, http://www.sos.sk/pdf/king/king-smdled.pdf

[3] IR WORLD HEADQUARTERS: IRF7341 HEXFET Power MOSFET, [online] Publication April .2005, http://www.irf.com/product-info/datasheet/data/irf7341.pdf

[4] Vishay Corporation: P-Channel 30-V (D-S) MOSFET, [online] Publication 03.03.2003, http://www.datasheetcatalog.org/datasheet/vis-hay/72123.pdf

[5] Megatron Elektronik AG & Co.: Halleffekt absolutwertgeber serie MAB25 SER, [online] Publication 28.02.2008, http://megatron.de /Impuls1/DB_MAB25.pdf

[6] SGS-THOMSON Microelectronics: QUARD BUS BUFFERS (3-STALE), [online] Publication in 1994, http://www.datasheetcata-log.org/datasheet/SGSThomsonMicroelectronics/mXrytxt.pdf

[7] TwiNgMan: Zbernica I2C, [online] Publication 17.04.2007, http://new.twingy.sk/rozne/zbernica-i2c

[8] Atmel Corporation: 8-bit AVR Microcontroller with 128K Bites In-System Programmable Flash , [online] Publication December .2003, http://www.datasheetcatalog.org/datasheet/atmel/2467S.pdf

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Mono Axial Vehicle platform for education purposes

Jozef Rodina
Institute of Control and Industrial Informatics
FEI STU
Bratislava, Slovak Republic
jozef.rodina@stuba.sk

Peter Hubinsky
Institute of Control and Industrial Informatics
FEI STU
Bratislava, Slovak Republic
peter.hubinsky@stuba.sk

***Abstract*—This paper presents a concept of the Mono Axial Vehicle (MAV) platform for education purposes. The chassis of the MAV can rotate around the wheel axis, hence the system behavior is changing with rotation of the chassis. This allows development of the different control methods on the same vehicle platform. Chassis of the MAV is made of composite materials and the wheels are made of aluminum. The control board is based on 16-bit DSP microcontroller and MEMS sensors. It is able to control up to six servos. Additional sensors may also be added. In order to speed up development, the C libraries and MATLAB based dynamic model are available.**

Keywords-mono axial vehicle, inverted pendulum, oscillation damping, ZVD shaper, LQ regulator, DSP, MEMS

I. Introduction

Mono axial vehicle (Fig.1) is a very interesting platform for design and evaluation of control methods for movement systems [1]. A system behaviour can be described like an inverted pendulum (centre of gravity of the chassis is located above the wheels axis) and a classical pendulum/two-mass system (centre of gravity of the chassis is located under the wheels axis). Therefore different control methods have to be used.

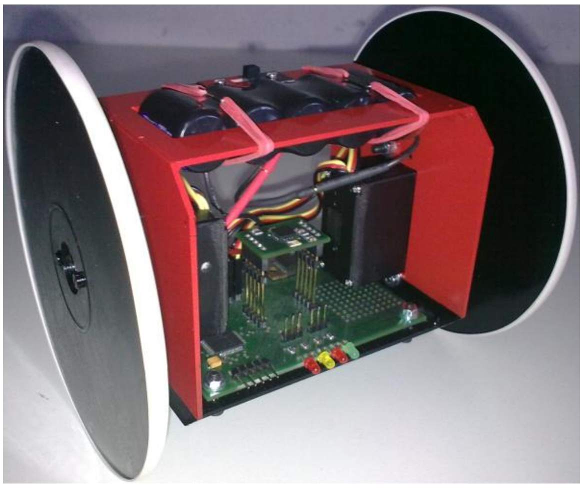


Figure 1. Mono Axial vehicle

The system in upper position is unstable therefore we have to use stabilization control. The LQR controller is used for this case. The system has tendency to oscillate in lower position. We are using two methods for oscillation damping: input signal shaper (feed-forward methods) and derivative feedback. A block diagram of the motion control is shown in Fig.2. All these control methods have been tested in MATLAB SIMULINK.

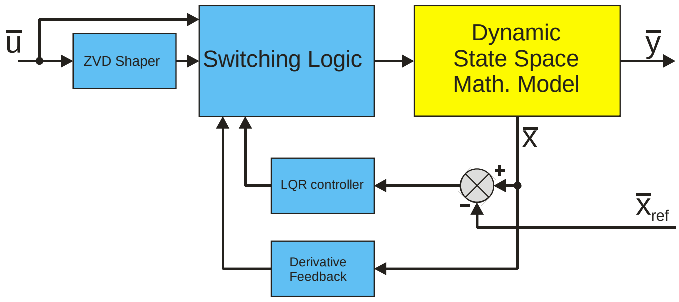


Figure 2. Principal scheme of the motion control of the Mono axial vehicle

II. Mathematical Model

The MAV used two motors for its motion and this cause that not only straightforward motion is possible, but we consider only straightforward motion in mathematical model. In order to develop the control system, we need to create a mathematical model of system that will be corresponding with behaviour of the real system. This system behaviour is similar to pendulum on wheels. The body, wheel and DC motor dynamic are analyzed separately at the beginning, but in the end we will get two nonlinear equations of motion that completely describes dynamic of the vehicle. These equations are nonlinear, but in our case of the hybrid control system we have to linearise them.

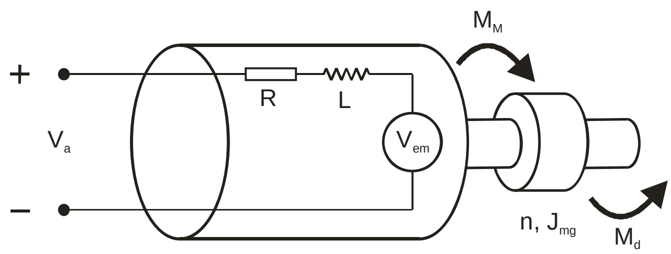


Figure 3. Diagram of DC motor

According to the Fig. 3 the following equations for DC motor can be defined:

$$M_M = Cu.i.n \tag{1}$$

$$\dot{\omega} = \frac{Cu.n}{J_{mg}.R}V_a - \frac{Cu^2.n}{J_{mg}.R}\omega - \frac{M_d}{J_{mg}} \tag{2}$$

where:

M_M	torque produced by DC motor
Cu	torque constant
i	current generated in the motor armature
n	gearbox moment gain
ω	output shaft angular velocity
$\dot{\omega}$	output shaft angular acceleration
J_{mg}	moment of inertia of the motor and gearbox
M_d	disturbance torque on output shaft
R	rotor winding resistance
V_a	voltage applied on a motor terminals

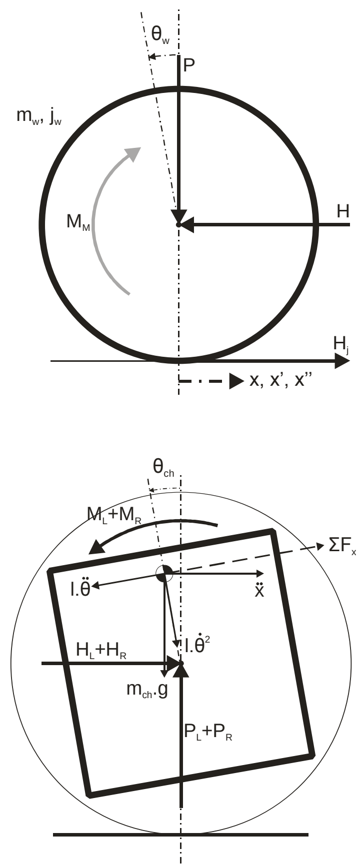


Figure 4. Free body diagram of the wheel and the chassis

For the wheel and the chassis (Fig.4)

$$(m_w + \frac{J_w}{r^2})\ddot{x} = \frac{Cu.n}{R.r}V_a - \frac{Cu^2.n}{R.r^2}\dot{x} - H \tag{3}$$

$$H_L + H_R = m_{ch}.l.\ddot{\theta}\cos\theta - m_{ch}.l.\dot{\theta}^2 + m_{ch}.\ddot{x} \tag{4}$$

$$\begin{aligned}&(H_L + H_R).l.\cos\theta + (P_L + P_R).l.\sin\theta - \\ &- m_{ch}.g.\sin\theta - m_{ch}.l.\ddot{\theta} = m_{ch}.\ddot{x}.\cos\theta\end{aligned} \tag{5}$$

$$\begin{aligned}&-(H_L + H_R).l.\cos\theta - (P_L + P_R).l.\sin\theta - \\ &-(M_L + M_R) = J_{ch}\ddot{\theta}\end{aligned} \tag{6}$$

where:

m_w	wheel mass
r	wheel radius
J_w	moment of inertia of the wheel
$\dot{x}, \ddot{x}$	wheel velocity, wheel acceleration
m_{ch}	chassis mass
l	distance from center of the chassis to the center of gravity of the chassis
J_{ch}	moment of inertia of the chassis
H	force
P	force
$\theta_{ch}, \dot{\theta}_{ch}, \ddot{\theta}_{ch}$	angle, angular velocity and angular acceleration of chassis
$\theta_w, \dot{\theta}_w, \ddot{\theta}_w$	angle, angular velocity and angular acceleration of wheel

By rearranging and combining equations 1 - 6 we get the non-linear equations of motion of the mono axial vehicle:

$$\begin{aligned}&\ddot{\theta}(J_{ch} + m_{ch}l^2) = \frac{2Cu^2n}{Rr}\dot{x} - m_{ch}.l.\cos\theta\,\ddot{x} - \\ &- m_{ch}.l.g.\sin\theta - \frac{Cu.n}{Rr}(V_{AL} + V_{AR})\end{aligned} \tag{7}$$

$$\ddot{x}(2m_{ch} + \frac{2J_w}{r^2} + m_{ch}) = \frac{2Cu^2 n}{Rr^2}\dot{x} - m_{ch}.l.\cos\theta\ddot{\theta} - \\ - m_{ch}.l.\sin\theta\ddot{\theta}^2 + \frac{Cu.n}{Rr}(V_{AL} + V_{AR}) \tag{8}$$

These two equations above can be linearised in a surround of upward position. Therefore:

$$\cos\theta = -1 \qquad \sin\theta = -\theta \qquad \frac{d\theta}{dt} = 0$$

We get:

$$\ddot{\theta}(J_{ch} + m_{ch}l^2) = m_{ch}.l.\ddot{x} + \frac{2Cu^2 n}{Rr}\dot{x} + \\ + m_{ch}.l.g.\theta - \frac{Cu.n}{Rr}(V_{AL} + V_{AR}) \tag{9}$$

$$\ddot{x}(2m_{ch} + \frac{2J_w}{r^2} + m_{ch}) = \frac{2Cu^2 n}{Rr^2}\dot{x} + m_{ch}.l.\ddot{\theta} + \\ + \frac{Cu.n}{Rr}(V_{AL} + V_{AR}) \tag{10}$$

$$\ddot{\theta} = \frac{m_{ch}.l}{(J_{ch} + m_{ch}l^2)}.\ddot{x} + \frac{2Cu^2 n}{(J_{ch} + m_{ch}l^2)Rr}\dot{x} + \\ + \frac{m_{ch}.l.g.\theta}{(J_{ch} + m_{ch}l^2)} - \frac{Cu.n}{(J_{ch} + m_{ch}l^2)Rr}(V_{AL} + V_{AR}) \tag{11}$$

$$\ddot{x} = \frac{2Cu^2 n}{(2m_{ch} + \frac{2J_w}{r^2} + m_{ch})Rr^2}\dot{x} + \frac{m_{ch}.l}{(2m_{ch} + \frac{2J_w}{r^2} + m_{ch})}\ddot{\theta} + \\ + \frac{Cu.n}{(2m_{ch} + \frac{2J_w}{r^2} + m_{ch})Rr}(V_{AL} + V_{AR}) \tag{12}$$

By substituting equations 11 into equation 10 and equation 12 into equation 9 we are able to obtain state space equations of the system. Coefficients A_{22}, A_{23}, A_{42}, A_{43}, B_1 and B_2 are obtained from equations 11 and 12 [2].

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_{42} & A_{43} & 0 \end{bmatrix} . \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \\ 0 \\ B_4 \end{bmatrix} . \begin{bmatrix} V_{AL} & 0 & V_{AR} \end{bmatrix} \tag{13}$$

A. *Mathematical Model Simulations*

We simulate the dynamic model by applying the step of the control signal on the inputs of model. This excites oscillations of the chassis. Also disturbances affected on the chassis (applying torque on the chassis is affecting $\ddot{\theta}_{ch}$) excite oscillation. This disturbance is used in all simulations. This can be seen in Fig.5 and Fig.6. The mathematical model of the MAV is in stable position (at 180 deg.) in these simulations.

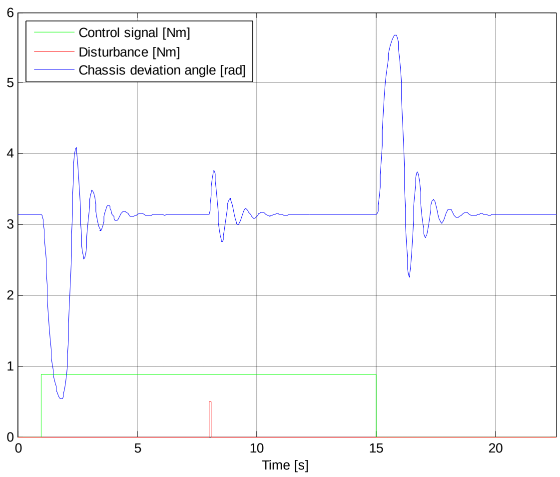


Figure 5. Oscillation of the chassis of the MAV

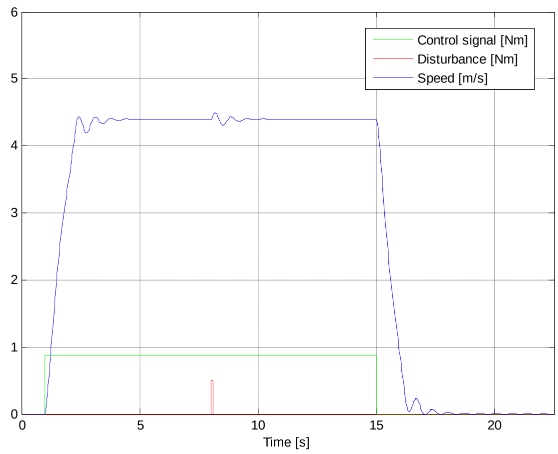


Figure 6. Oscillation of the speed of the MAV

III. OSCILLATION DAMPING

A. *Control Signal Shapers*

Changes of control signal (wheels speed is changing) are exciting oscillation (around lower position) in the MAV. This oscillation or this state of the MAV is called residual oscillation. To avoid this oscillation input signal shapers are usually used. As can be seen in Fig.7 the shaped signal is obtained by convolving desired input with the series of Dirac impulses.

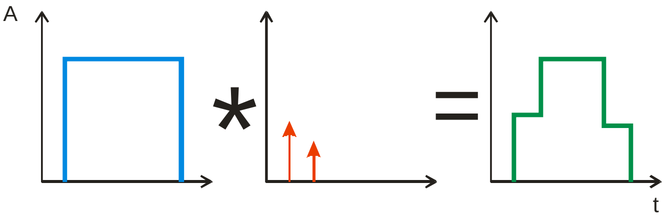


Figure 7. Basic principles of signal shaping methods

Basic type of shapers is Zero Vibration Shaper (ZV shaper). It uses only 2 impulses. The ZV shaper is sensitive to the changes of the system own natural frequency, therefore in our case we are using Zero Vibration Derivative Shaper (ZVD shaper). The ZVD shaper is less sensitive to changes of the system natural frequency and its uses 3 impulses. ZVD shaper for system with a natural damping *b* is de-scribed by following equation:

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{(1+k)^2} & \frac{2k}{(1+k)^2} & \frac{k^2}{(1+k)^2} \\ 0 & \frac{T_D}{2} & T_D \end{bmatrix}; k = e^{-\frac{b.\pi}{\sqrt{1-b^2}}} \quad (14)$$

where:

A_i	Amplitude of i-th Dirac impulse
t_i	Delay of i-th Dirac impulse
T_D	Period of the system natural oscillation

Simulation scheme of the ZVD shaper is shown in Fig.8. [3].

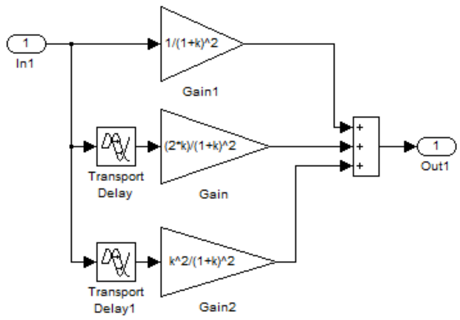


Figure 8. Simulation scheme of the ZVD Shaper

B. Evaluation of the ZVD Shaper

For evaluation we created a simulation model of damping control using ZVD shaper (Fig.9).

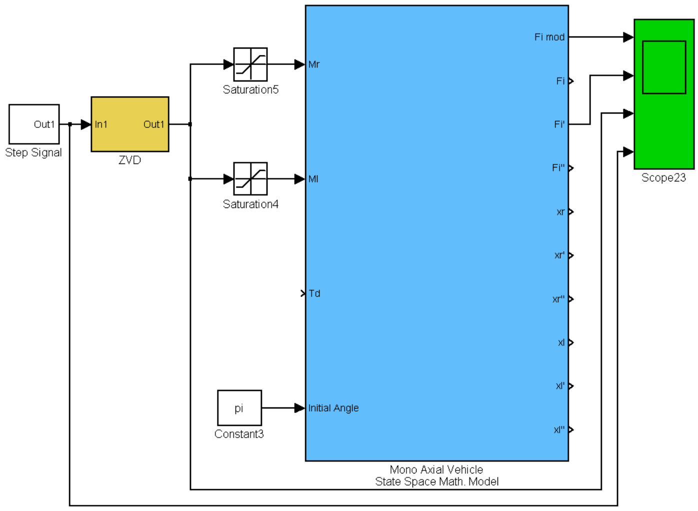


Figure 9. Simulation model of damping control using ZVD shaper

First we need to identify oscillation parameters of the MAV. Resonant frequency and period of the MAV was calculated as follows:

$$\omega_0 = \sqrt{\frac{m_{ch}.g.L}{J_{ch}}}; T_0 = \frac{2\pi}{\omega_0} \quad (15)$$

The damping ratio b was identified experimentally from the oscillation shown in Fig.5. ZVD shaper was set by using these parameters. The ZVD shaper was tested with the same input signal as was in the simulations above (Fig.5, Fig.6). Next two graphs (Fig.10, Fig.11) shows result of these simulations.

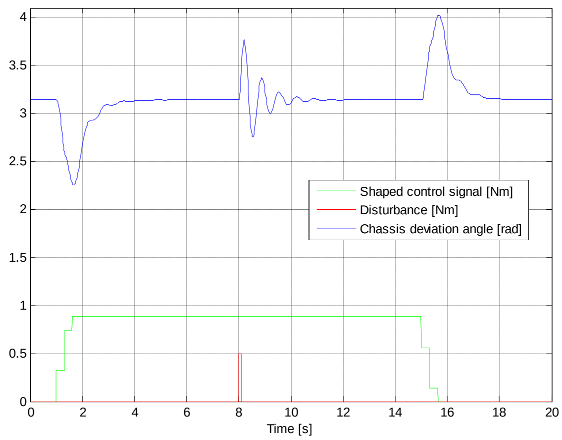


Figure 10. Graph of the chassis angle of the MAV when is used the ZVD shaper

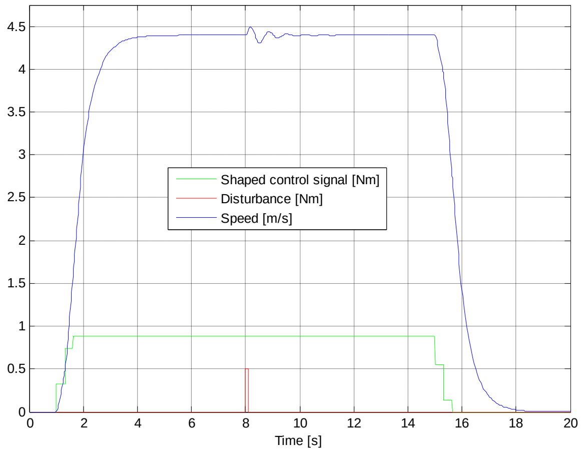


Figure 11. Graph of the velocity of the MAV when is used the ZVD shaper

C. Derivative Feedback

Next method is the derivative feedback. This method increases the naturally low damping ratio of systems to appropriate value. By using the derivative feedback we can move oscillating complex poles of the system into the position near to the real axis in the complex plane, so the damping ratio of the system will increase. Scheme of typical derivative feedback is illustrated in Fig.12. We want to damp oscillation of the chassis angle. We will use the signal of angular velocity of the chassis for feedback. So derivation of the angle output of model is not needed in this case, because it is available from the Mono Axial Vehicle simulation block. Derivative feedback gain was set experimentally. If the system contains yet another low damped pole pair, it can be destabilized easily. This might decrease effectiveness of this method.

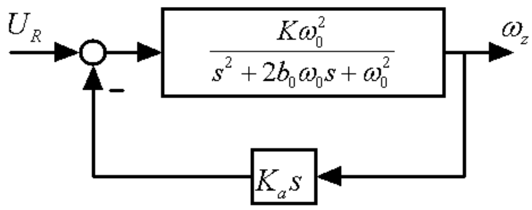


Figure 12. Typical structure of the control system with derivative feedback

From Fig. 12 we can define these parameters of system :

ω_0	natural frequency of a system
b_0	natural damping of a system
K	system gain
K_a	gain of the derivative feedback

We can calculate the gain of the derivative feedback by following equation:

$$K_a = \frac{2(b-b_0)}{K} \qquad (16)$$

where:

b	desired value of system damping (optional)

D. Derivative Feedback Evaluation

As mentioned before we set derivative feedback experimentally. The value which gain feedback was finally set is equal to 0.23. Results of the simulations are visible in Fig.13 and Fig.14. Derivative feedback in comparison with the ZVD shaper is able to avoid even oscillation caused by the disturbances. Also by using this method the damping coefficient of the system was increased.

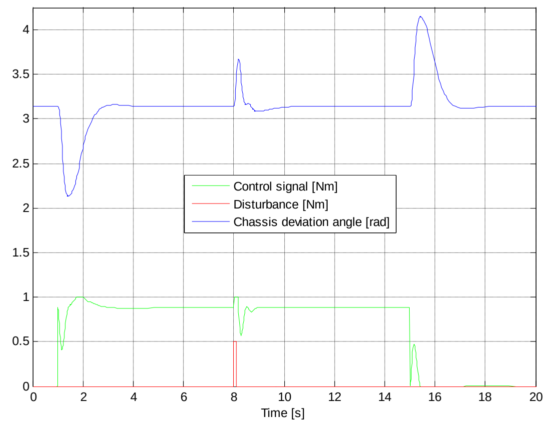


Figure 13. Graph of the chassis angle of the MAV when is used the derivative feedback

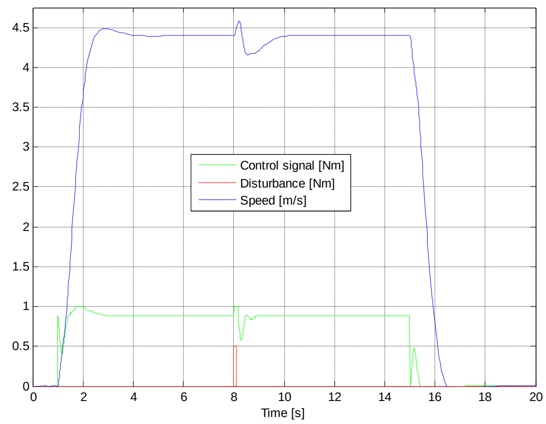


Figure 14. Graph of the velocity of the MAV when is used the derivative feedback

IV. THE LQR CONTROLLER FOR STABILIZING THE MAV

Linear Quadratic Regulator Controller is optimal state-feedback controller with good robustness for robotic applications (Fig.15). In fact the LQR controller is optimal pole placement controller.

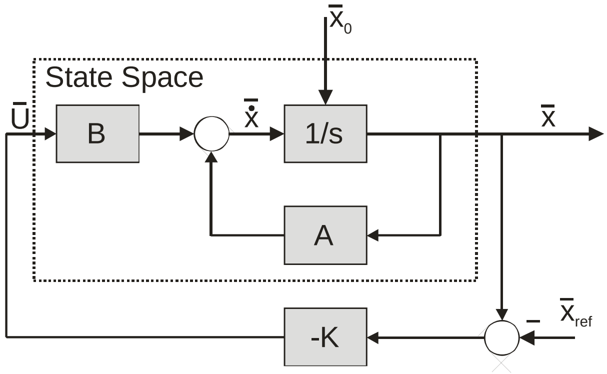


Figure 15. LQR controller

In order to design the LQR controller we need to get the linear state space model of the system as follows:

$$\dot{x} = Ax + Bu \qquad y = Cx \tag{17}$$

And the control law of LQR is defined by this equation:

$$u = -K(x - x_{ref}) \tag{18}$$

The control gain K (19) is obtained from the infinite horizon performance index J (20) and the solutions of the Riccati equation for infinite horizon is (21).

$$K = R^{-1}B^T P \tag{19}$$

$$J = \frac{1}{2}\int_{t_0}^{\infty}(x^T Qx + u^T Ru)dt \tag{20}$$

$$0 = A^T P + PA - \dot{P}BR^{-1}B^T P + Q \tag{21}$$

The matrices Q and R matrices are usually diagonal and first choice for these matrices can by given by using the Bryson's rule.

The LQR is a linear controller, therefore we have to linearise mathematical model of the MAV. Linearization of the dynamic model of the MAV can be done near an equilibrium of the MAV in upper position. In time of writing this paper we were still working on the LQR controller, therefore we didn't have any simulations available yet. [4]

V. Experimental Setup

Experimental MAV device is prepared for the efficiency measurement of proposed methods is based on the following hardware components:

Drive:
2 x MAXON RE35 90Watt (15.0V nominal voltage) + Planetary gearbox GP 42 C with ratio 12:1, IRC sensor HED–5540. Motor controller MAXON EPOS 24/5

Sensors:
3 axes accelerometer + 3 axes gyroscope, 2 axes magnetometer, laser scanner Hokuyo URG-04LX, Ultrasonic range sensors and GPS SiRFstar III receiver.

Accumulators:
4 – cells LiFePo A123 2200 mAh. Rated for 30C continuous discharge (66.6A)

Control board:
Based on dsPIC33FJ64GP306 (16 bit digital signal controller). This is used for low-level control (ZVD, LQG, Kalman Filtering etc.). Devkit 8000 based on OMAP3530 (ARM Cortex A8) for high-level control (navigation etc.).

Communication board:
Based on zigbee modules. This board will provide telemetry data and remote control for the MAV

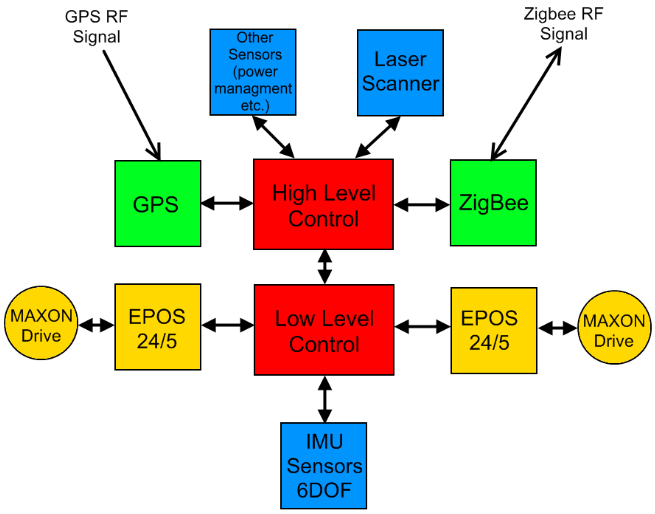


Figure 16. Block diagram of the MAV

Only part of the sensors is used for the damping of the oscillation and/or for stabilizing the platform, the rest is planned to be used for the navigation and obstacle avoidance or for other purposes followed from the application where the robot would is used. The block diagram of the experimental setup of the MAV is in Fig. 16. The second version of the MAV is being developed in these days. The Fig. 17 shows the part of the 3D virtual model of the MAV.

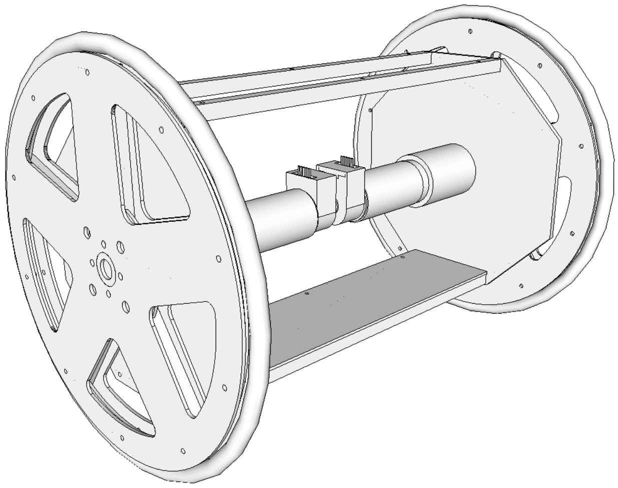


Figure 17. 3D virtual model of the MAV version 2

VI. CONCLUSION

This paper presents various control techniques to damp oscillation and optimal control of the MAV which can be used as platform for service mobile robots.

First we analyzed oscillation and designed ZVD shaper for damping oscillation. In next step we present derivation feedback, which is used for oscillation damping. ZVD shaper does not affect stability of the system, but it is also not able to damp oscillation caused by disturbances. The derivation feedback in comparison with the ZVD shaper is able to eliminate oscillation caused by disturbances, but it might destabilize the system. Next advantage of the ZVD shaper can be that it does not require any additional sensors. Also the ZVD shaper can be used if sensors fail. We can also combine these two methods for oscillation damping. For example we will use ZVD shaper during the phase of control signal change, but after this phase we will use derivation feedback.

We also showed basic principle of the LQR controller. Since we are still working on it during the time of writing this paper, the LQR control method is not described and evaluated here.

These control techniques describe just oscillation damping and stabilizing of the MAV. Next step will be to implement this hybrid control into real MAV and to implement methods for navigation in environment to the MAV [5]. Now we are also working on chassis of the MAV and on electronics for it. Chassis will be based on composites materials. The MAV will be driven by MAXON motors and planetary gear. Electronics is based on DSP microcontroller and MEMS sensors.

ACKNOWLEDGMENT

This work was supported by Grant Agency of Ministry of Education and Academy of Science of Slovak Republic VEGA under Grant No. 1/0690/09. The authors are pleased to the acknowledge this support.

[1] RODINA,J.-HUBINSKÝ,P.: Stability control design of segway like differential drive by using MEMS sensors, s.483-487, Metalurgija (Metallurgy) č.2, roč.49, 04-06/2010, Croatia, ISSN 0543-5846.

[2] GRASSER F., D'ARRIGO A., COLOMBI S., RUFER A.: JOE: A Mobile, Inverted Pendulum, Published in 2001 [online] http://leiwww.epfl.ch/publications/grasser_darrigo_colombi_rufer_mic_01.pdf

[3] SINGHOSE W., SINGER N., SEERING W.: Comparison of Command Shaping Methods for Reducing Residual Vibration, Published in 1995. [online] http://www.me.gatech.edu/inputshaping/Papers/ECC95.

[4] KOZÁKOVÁ A.: Design of discrete-time compensator for reference tracking with disturbance rejection. /Transactions of the VŠB - Technical University of Ostrava, Mechanical Series/, 2008, vol. LIV, No.2 (2008), 67-72. ISBN 978-80-248-1727-9

[5] BABINEC A., DUCHOŇ F.: Metódy reaktívnej navigácie mobilného robota v neznámom prostredí, Published in: Nové trendy v kybernetike, automatizácii a informatike : Odborný seminár. Gabčíkovo, Slovenská republika, 7.-9.9.2009. - Bratislava : STU v Bratislave FEI, 2009. - ISBN 978-80-227-3107-2. - CD-ROM

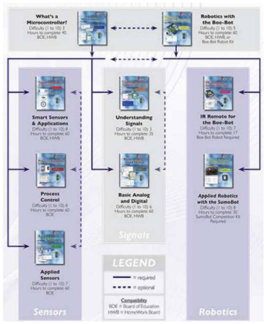
Boe-Bot Robotics Kits Used in Education

Jessica Uelmen
Parallax Inc.
Rocklin, CA, USA
juelmen@parallax.com

About Parallax

- Privately held company in Rocklin, CA
- Founded by Chip Gracey & Lance Walley in 1987
- 35+ employees in Research and Development, Sales, Manufacturing, Education, Marketing and Technical Support
- Designs and manufactures microcontrollers, robots, and sensors
- Strive to provide the electronics design industry with products that are technically innovative, unique, and economical

Stamps in Class Program



- Used in technology and pre-engineering programs from middle school to college
- Teaches electronics and programming in an integrated, hands-on fashion
- Introduces diverse subjects
  - Programming
  - Electronics
  - Process control
  - Physics
  - Sensors
  - Robotics
  - Digital signals

The Boe-Bot Robot

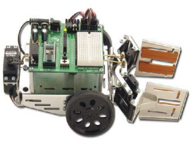
- Flexible, reprogrammable robotics platform
- Features the BASIC Stamp 2 microcontroller and easy-to-learn PBASIC programming language
- Robotics with the Boe-Bot text includes 40+ hands-on activities
- Expandable with sensors and hardware add-on kits
- Used in thousands of classrooms worldwide

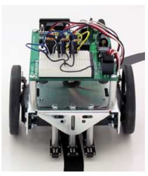
Classroom Applications

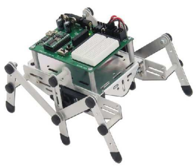
- Middle School (Ages 11-14)
  - Create handouts with simple concepts
  - Code is given to students
- High School (Ages 15-18)
  - Told to follow book instructions
  - Only then more complicated code is given
- University (Age 18+)
  - Follow book at faster pace, but ideas are expanded on
    Ohm's Law, Kirchhoff's Current Law, etc.
  - Used in many introductory microcontroller, engineering, and mechatronic courses

Why Teachers Choose the Boe-Bot

- Cost-effective
- Easy to get started, infinite possibilities
- Resources & support
  - Parallax Educator's Forum
  - Free PDF downloads for all books
  - Educator's Courses



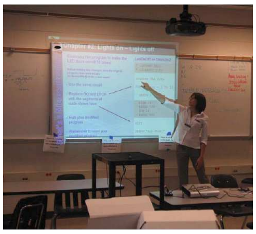




Programming Languages

- PBASIC is a great first programming language
  - Easy to understand and get started with
  - Creates a strong foundation in elemental programming concepts
- Higher-level courses (i.e. university) need more language options
  - C is an industry standard: prepares students for the workforce
  - Assembly may be eventually needed for high-performance applications
  - New language variants are being developed for specific industries and new microprocessor architectures
- The ability to learn programming languages is an important skill in itself

Teacher Support: Educator's Courses



- Two day teacher training events
  - *What's a Microcontroller?* (Day 1)
  - *Robotics with the Boe-Bot* (Day 2)
- Teachers get to keep all kits used in the course
  - Helps solidify concepts learned
  - Demonstrate to faculty
- Attendees range from middle school science teachers through university professors

Course Benefits

- Provides us with feedback on how well concepts are presented
  - This information is used to:
    - Revise educational texts
    - Create supplemental activities
    - Change the way we teach the courses
- Keeps us up-to-date with educational trends and teacher needs
- Allows teachers a chance to collaborate with each other and share ideas for their classrooms

Generating Excitement

- Contests & giveaways!
- A feeling of support
  - Course instructor's contact
  - Educator's Forum
  - Free PDF downloads of all our books
  - Additional projects
  - `"How to" videos on YouTube
- Breaks

Challenges

- Addressing attendees at different skill levels
  - How to best present concepts
  - Letting no one feel 'left behind'
  - Bringing a stock of additional sensors & documentation
- Being conscious of not 'overloading' attendees
- Making sure material gets covered
  - Outline 'key concepts' to be covered during the course

What Students Want

- Small, topic-based projects
  - Instant gratification
  - Relatable projects
  - Multimedia resources
- The ability to grow and create their own inventions

Teacher Requests

- Option to program in a C language
- Platform that allows students to delve more deeply into the topics that interest them
- Quick & easy to use software and materials
  - Allows teachers more time to concentrate on the content of their math, science, or engineering class
- Accompanying CAD package for software simulation
- Ability to explore and change the robot's circuits
- Reusable & low cost platform

Future Plans

- Adding a new educational program centered around the Propeller chip
- Developing a C language for use with the Propeller
  - Will be as easy to pick up as PBASIC
  - Available on multiple platforms (Windows, Mac OSX, etc.)
- Materials will be more project-based and application-specific

Teaching Robotics at the Postgraduate Level: Delivering for On Site and Distance Learning Students

Jenny Carter and Simon Coupland

Abstract— **The MSc Intelligent Systems (IS) and the MSc Intelligent Systems and Robotics (ISR) programmes at De Montfort University are Masters level courses that are delivered both on-site and by distance learning. The courses have been running successfully on-site for 6 years and are now in the third year with a distance learning mode. Delivering material at a distance, especially where there is technical and practical content, always presents a challenge but the need to deliver a robotics module increased the challenges we faced significantly. There are two robotics modules though the second one is only available to those on MSc ISR. We have chosen to make the first robotics module, Mobile Robots, the focus of this paper because it was the first that had to be delivered and it is delivered to students on both programmes. This paper describes the rationale, delivery and assessment of the Mobile Robots module to students on the MSc IS/ISR with a specific focus on those students that are studying in distance learning mode. We believe it serves as a model for others attempting to teach robotics at distance.**

I. Introduction

The MSc Intelligent Systems (IS) and MSc IS and Robotics (ISR) programmes are delivered both on-site and by distance learning. MSc ISR (previously MSc Computational Intelligence and Robotics) has been running successfully on-site for 6 years and the more recent variant of the course, MSc IS, for 3 years. Both courses are now in their third year of running with a distance learning (DL) mode. The two courses share 7/8 modules, the eighth being a second robotics module for MSc ISR and a data mining module for MSc IS (see Figure 1). All modules are assessed by coursework only so there are no examinations. We believe this enables us to set realistic assessment exercises that stretches the students and encourages the investigation of novel areas. A significant number of our Masters students have published papers resulting from their work either on the final project or in some cases resulting from their module assessments. The courses have evolved and developed over the years and attract significant numbers of students, especially to the distance version – so for example around 20 new students enrolled at the beginning of the last two academic years. The courses can be studied in full or part time modes by on-site students and in part time mode by DL students. Full time students take a complete calendar year to complete an MSc programme and part time students can complete it in a minimum of two or maximum of six years, though three years is more typical.

One of the biggest challenges has been the development and delivery of the robotics module at a distance and this is why we have chosen to make the Mobile Robots module the focus of this paper.

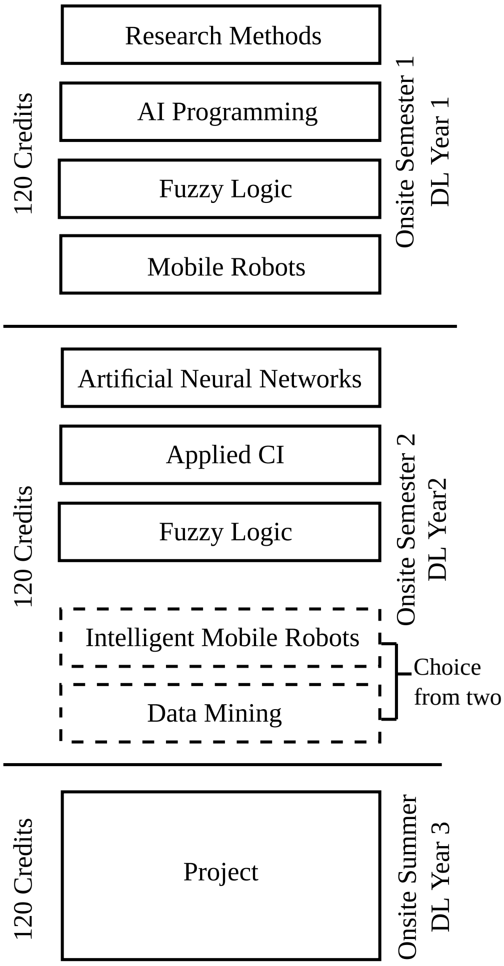


Fig. 1. Course Structure for MSc IS and ISR.

The remainder of this paper is structured as follows: Section II considers the background of the courses, their development and structure; Section III the approaches to learning that we have adopted for the course, Section IV gives a detailed account of the delivery of the Mobile Robots module and finally Section V draws conclusions from this work.

II. Background

The courses are delivered mainly by the members of the Centre for Computational Intelligence (CCI) at De Montfort

Jenny Carter and Simon Coupland are with the Centre for Computational Intelligence (CCI), De Montfort University, Leicester, LE4 1BH, United Kingdom, (emails: jennyc@dmu.ac.uk, simonc@dmu.ac.uk).

University. Their development enabled us to capitalise on the research taking place within the CCI and therefore on the strengths of the staff delivering the modules. It is generally preferred that staff members teach their special interests thus enabling research to support teaching and vice versa. There are significant benefits when a course is delivered by a team of people with a strong interest and research focus in the same areas. Initial decisions were necessary to determine the content of the courses. There are a large number of topics that could be considered but the areas of fuzzy logic, neural networks, evolutionary computing, knowledge based systems and logic programming provide an array of tools and paradigms that encompasses much of what is considered to be computational intelligence and thus form the basis of the content. The ability to get mobile robots to react intelligently to their environment is a highly developed research field and it is one that is being tackled within the CCI; it also provides an ideal application area for applying the previously mentioned techniques and therefore mobile robots modules were included. MSc ISR includes two mobile robots modules whilst MSc IS replaces one of these with a data mining module as an alternative application area for those less interested in pursuing mobile robotics work. In addition to the modules mentioned so far, a research methods module is delivered in semester 1 to ensure that students are equipped with the necessary skills to carry out literature searches, write project proposals and so on; and the Applied Computational Intelligence module enables students to pursue an appropriate area of their own interest in greater depth.

III. Approaches to learning

We aim to adopt an approach to our delivery of the courses that embraces modern technology in such a way that the students have appropriate learning experiences whether they are studying on-site or at a distance.

De Montfort already uses Blackboard[1] as a platform for providing e-learning materials for all students and this is used extensively though not exhaustively in all faculties. It was therefore an obvious choice as the main platform for the MSc. Decisions about the best way to use Blackboard and which other resources to employ alongside it were necessary and as both on-site and distance students study the modules concurrently the experiences need to be as similar as possible. Some practises have been adopted for all modules and this includes providing physical materials (e.g. textbooks, software, and other materials as necessary). We also record lectures and post them on De Montfort's streaming server (DMUtube). The students are able to view the lectures from within Blackboard and it has proved to be a popular method. Other methods adopted include sound over Power-point slides using tools such as Articulate Presenter; software demonstrations using screen and voice recorders. [3] define the term 'networked learning' to describe a particular kind of web-based or on-line learning. Their definition of networked learning is "'learning in

[1]http://www.blackboard.com/

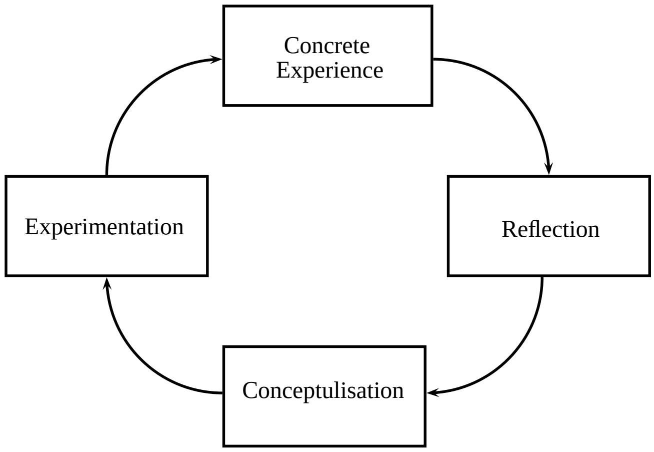


Fig. 2. Kolb's Learning Cycle [6].

which information and communications technology is used to promote connections: between one learner and other learners; between learners and tutors; between a learning community and its resources"'. In adopting this idea of networked learning, it is important for us to make sure that we are not simply providing materials in a variety of forms but that the learning is networked i.e. there is human to human communication taking place within each module. One way that we do this is to make use of an assessed discussion board on our virtual learning environment (VLE). It is assessed based on the number of contributions over the semester rather than the quality of the content. We have found this to be very successful and it is clear that it helps to create a virtual learning community amongst our students. Such communities are identified as being important for student engagement in e-learning by [2]. Our experience of using this mechanism has shown that it encourages students to become more of a cohort through communicating with each other whether on-site or at a distance and it helps the distance students in particular to feel less on their own. The discussion board component is worth 10% on every module and it is this that encourages students to use it initially. We find that as they get used to using it they become more involved and often answer each other's questions and so on. Other practises used, though to a lesser extent, are blogs which are used for keeping project journals and also as a way of putting current students in touch with past graduates from the course; a Facebook group; and more recently wikis for sharing subject related ideas and student presentations.

In order to deliver the course effectively it has been useful to consider approaches to learning and teaching in higher education more generally. Most of the modules include both theoretical and practical work and the assessments are usually open enough to allow the students to investigate appropriate topics in their own way thus there is an attempt to facilitate experiential learning as defined by [6]. Kolb suggests that learners acquire knowledge according to the learning cycle shown in Figure 2. A further example of this approach in practise can be seen in the design of the OU module, Artificial Intelligence for Technology in [5], [9]. Here they use a learning cycle that is derived from [6]. These can be considered to be constructivist approaches to learning where students construct their own knowledge through various experiences within (and without) of the course. We believe it to be very important for our students to draw on non-course experiences as many of them have work experience: for example, DL students are often in full time employment, there is a wide variety of first degree subjects amongst them and a significant number already have PhDs. Due to these factors, often our students are interested in applying the knowledge gained from the MSc within their working environment or to their previous subjects or research area.

On-line learning in higher education is also considered by [1], who describes four levels of interaction as part of an on-line curriculum interaction model. Level 1 includes materials presented as text, Powerpoint presentations, videos etc. and relies on the students' own motivation. Level 2 has increased interaction amongst the students such as the discussion board activities used extensively on our MScs. Level 3 includes synchronous activities such as chat rooms and the final level, 4, is where the highest level of the e-learning community is offered and is where the learners and instructors engage in a variety of synchronous activities. Level 4 is achieved to some extent on our courses when we hold meetings, presentations, demonstrations (usually using Skype) with tutors and students. We plan to increases this as the course evolves further.

Dabbagh [2] defines pedagogical models for e–learning namely: open learning, distributed learning, learning communities, communities of practices and knowledge building communities. We believe that our approach incorporates aspects of the distributed learning model and to some extent, the learning communities model. Distributed learning is where the learners are in many different locations and can choose to study at times that suit them. Communication with the faculty staff and other students is through a variety of electronic means. Learning communities are groups of people who support each other in their learning activities and can be broadly defined as including "any social network or infrastructure that brings people together to share and pursue knowledge" [2]. The next section focuses on how we approach the delivery of the Mobile Robots module on the MSc.

IV. The Mobile Robots module

To be successful the mobile robotics module must combine hands-on practical work with advanced theoretical concepts. The teaching and assessment strategies have to work face to face and at a distance. For many students this module is their first exposure to programming robots and the first time they have come across the inherent challenges such as hardware limitations, behavioural debugging and dealing with uncertainty. To best support our diverse student population we have developed a clear delivery strategy which we believe serves as a model when delivering a first semester postgraduate robotics module. Our strategy is depicted in Figure 3.

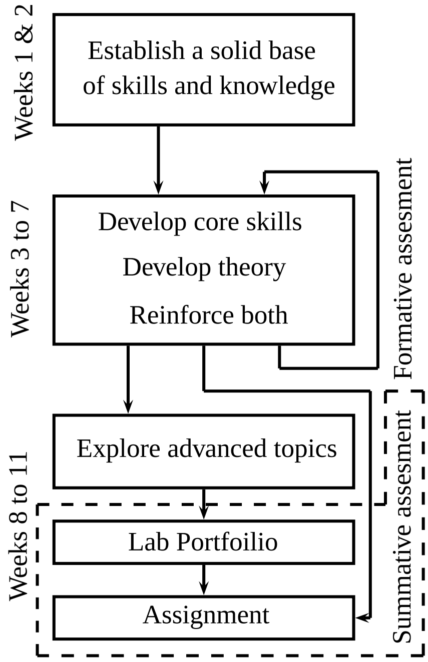


Fig. 3. Teaching and Assessment Strategy for Mobile Robots.

A. Establishing a Solid Knowledge and Skills Base

Arguably the most important and probably the most difficult to teach a at distance part of the module is the first two weeks. It is vital that students come out of these first two weeks with the core knowledge and skills to make progress on the module. The students come on the module from a diverse set of backgrounds, some may have good knowledge of the topics covered in these first two weeks, others may have limited or no experience. For on-site students it is relatively easy to judge a student's starting level through body language and informal questions. For distance learners a different approach must be taken. We supply a range of learning materials in these early weeks, the compulsory material covers topics at a fairly high and abstracted level but contains pointers to deeper material which gives more detailed explanations and worked examples. Students are strongly encouraged to dig down in the material until they are confident in their understanding and are able and motivated enough to do this. To establish the core competencies needed by the students a set of multimedia materials are provided to the students. These materials cover what might be termed housekeeping issues, but are essential to progress in the module, topics include:

- Building the robot model.
- Changing the batteries in the robot.
- Updating the robot's firmware.
- Basic operation of the robot.
- Installing the BricxCC IDE.
- Using the BricxCC IDE: writing your first program, compiling, uploading and executing.
- Installing GCC with OpenGL and OpenCV.

- Setting up compiler short cuts and makefiles.

The media covering these topics are videos of lectures, videos of staff using the robots, video tutorials of software, lecture slides, documentary notes and textbook sections. An example taken from this material is given in Figure 4 which is taken from a set of instructions detailing how to modify the default Lego model for use in the module.

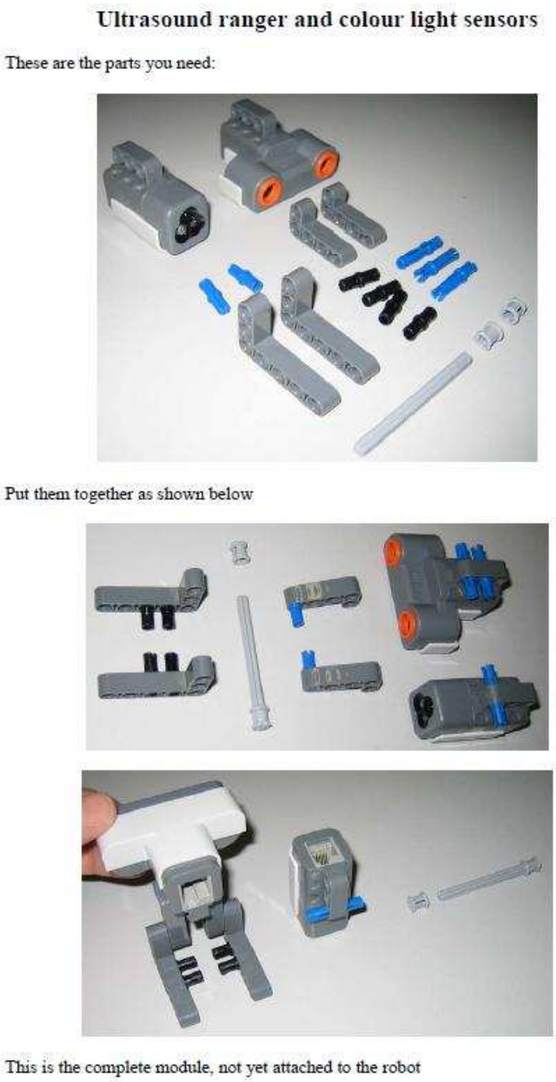


Fig. 4. Instructions for Robot Modifications.

B. Developing Core Skills and Theoretical Underpinning

Through weeks 3 to 7 students are presented with a range of theoretical topics:

- Sensors and actuators.
- Low-level control.
- Real-time programming.
- High-level control.
- Behavioural control architectures.

The content and delivery pattern of this phase of the module is cumulative by design: each topic depends on an understanding of the previous topic. From a technical standpoint this can be seen as a bottom up approach to teaching robotics. We start from a basic understanding of how a reading of an environmental phenomenon can be taken by a machine, through control strategies for simple actuators, programming issues associated with such devices to higher level, abstracted control strategies. We deliberately take this approach to avoid glossing over important issues and sources of uncertainties in mobile robots. We could take the opposite approach and begin with a high level view and then drill down to what is really happening. We have chosen the bottom up approach as it gives students an explanation for the idiosyncrasies of robot control from the outset. When the students come to write a high level control program, let's say obstacle avoidance using an ultrasonic range sensor, and the robot fails, crashing into an obstacle, the students already know what may have caused the fault. They are aware that different material reflect sound in different ways, that ultrasonics sensors have conical wavefronts not perfect straight lines and that perhaps the thread checking for obstacles is not run frequently enough.

One very important aspect of this phase of the module for distance learners is the high level of formative assessment and feedback given on a weekly basis. The students undertake a lab based piece of work every week which in some way gives a practical insight into the theoretical material. This lab work is assessed and marks and feedback are given to the students using the Blackboard virtual learning environment. The grades are purely formative and give the students a clear measure of the level they are working at and what they can do to improve. It is important that the deliverables for these formative labs are short and concise or the level of marking quickly becomes burdensome. Clear guidelines are given to the students in this regards and lengthy submissions penalised. Two of these lab pieces form part of the summative assessment going in to the lab portfolio as discussed later on.

At the end of week 7, students have covered all the core aspects of mobile robots in theory and practice. They are aware of the issues that arise when working with robotics and have a practical experience of working with robots.

C. Exploring Advanced Topics

Having established all the key knowledge and skills the students need to meet the module learning objectives, we then take a brief look at some of the more challenging topics in robotics namely:

- Robot/computer vision.
- Collaborative robots.
- Computational intelligence in robotics.

The second module, intelligent mobile robots (see Figure 1) covers in detail what most academics consider to be the advance topics in robotics: navigation, localisation and path planning. These topics are not covered in the mobile robot module, where we look at this different set of advanced issues. Each of the subject areas listed above is covered by one weeks worth of materials. The lab work on vision systems and collaboration is summatively assessed as part of the lab portfolio, giving the best students an opportunity to excel. A significantly advanced piece of software is provided to the students on each of these topics. Since only one week is given over to each of these topics, it is unreasonable to expect any of the students to begin work on any of these topics from a blank sheet of paper.

For the robot vision lab the students are given a working face recognition program which uses hue masks [7] and morphology operators [8] to identify a human face. The software makes extensive use of the OpenCV library. The students task is to choose an item for which they will modify the face recognition program so that it recognises this new object. The lab brief gives the best students the opportunity to showcase their technical skills and the knowledge of scientific method. Results from experiments showing classification rates and statistical analysis are not uncommon amongst the top 20%.

For the collaborative robot lab the students are given a simulated blackboard [4] server and four simulated clients who connect to the blackboard[2] via TCP/IP. All the networking is taken care of and the students' task is to decide what information should be transmitted and when that information should be transmitted. Students at the lower end of the spectrum tend to struggle with this work, although most get a pass mark. Students at the high end take the work much further implementing multi-threaded communication systems and advanced visualisation tools.

The final topic covered on the module, computational intelligence and robotics, is only covered at the theoretical level. The didactic material covers areas where computational intelligence has been shown to be useful in robotics with examples from the literature and from work in our own lab.

D. Assessment

Assessment of robotics work is generally challenging, these challenges are compounded when assessing work from distance learners.

Our assessment rationale is clear – we assess each student against clear set of learning outcomes. On completion of this module, the student should be able to:

- Demonstrate a comprehensive understanding of the principles and techniques used in building and controlling autonomous mobile robots by the design and implementation of adaptable controllers for autonomous mobile robots on a real robot system.
- Demonstrate a comprehensive understanding of the theoretical principles of the techniques used in building and controlling autonomous mobile robots and of the advances that are being made in this field.

The scale of assessment must clearly differentiate between pass and fail and between pass and distinction. To pass, a student must demonstrate that they have met the learning outcomes. To achieve a distinction students must meet the learning outcomes, show high levels of skill in the controller design and implementation and demonstrate a deep theoretical understanding of the issues covered on the module. This summative assessment against these criteria is done with two submissions, the lab portfolio and the assignment. The lab portfolio contains work from weeks 4, 6, 8 and 9. These labs allow the students to demonstrate a breadth of understanding: week 4 covers low level programming and sensors, week 6 covers behavioural control, week 8 covers vision systems and week 9 covers collaborative robotics. Weeks 4 and 6 labs are relatively straight forward and weeks 8 and 9 are more challenging. It is important that the portfolio covers a wide breadth of topics and that the assessment allows the students to demonstrate their theoretical understanding of the work covered. The assignment is submitted after the main teaching period. The students have to build a robot controller to perform line following using a behavioural architecture of their choice. This allows each student to demonstrate all the technical knowledge they have acquired throughout the module i.e.:

- They must choose an appropriate architecture and justify that choice.
- They must build a controller to behave in line with the specification.
- When the controller fails they should show an understanding of why the failure occurred.
- They should attempt to measure the performance of their controller, choosing appropriate metrics.

Some of these are easily assessed through documentary evidence, however controller performance needs to be demonstrated through the real-time operation of the robots. For onsite students this is done through formal demonstrations, for distance learning we offer them a live video demonstration (usually via skype) or allow them to submit a video of their controller on their robot with audio commentary on the robot's performance.

E. Student Performance

Table I gives the student numbers and pass rates on the mobile robots module over that past three years and table II

TABLE I

STUDENT PERFORMANCE 2007 – 2010.

	07/08		08/09		09/10	
Numbers	OS	DL	OS	DL	OS	DL
Enrolled	5	0	0	6	9	6
Failed	1	0	0	0	0	0
Pass	1	0	0	0	5	4
Distinction	2	0	0	5	4	2
Deferred	0	0	0	1	0	0

TABLE II

STUDENT PERFORMANCE RATES 2007 – 2010.

	OS	DL
Failed	8%	0%
Pass	46%	36%
Distinction	46%	64%

[2]Not to be confused with the Blackboard virtual learning environment discussed in this paper.

distils these numbers in to fail, pass and distinction rates for on-site (OS) and distance learners (DL). It seems that study at a distance presents no barrier to students achieving the highest standards on this module, in fact a slightly higher rate of distance students achieve a distinction on this module.

V. Conclusion

Delivering courses at a distance is a topical area. With the many available mechanisms for interacting with learners electronically there are a number of choices to be made regarding the approach to take. In this paper we have described some of the decisions we made when developing the MSc Intelligent Systems and the MSc Intelligent Systems and Robotics for on-site and distance delivery. We have provided a case study of how this applies to one of the most practical modules, namely, Mobile Robots. We have discussed our strategy for the delivery of this modules namely: firm basis of practical skills, build theory and practical with frequent feedback and give space to those most able to push their skills and knowledge as far as they wish to. The pass (more so the distinction) rates give over the past three years show how successful this final point has been, particularly for distance learning students. We believe that by following this model it is possible to teach a technical, practical subject through a distance learning model, and shown that a lack of contact is no obstacle for well motivated and determined students. The module and the course are successful and sustainable with a total of 55 students currently enrolled (11 on site, the rest as distance learners). The course continues to evolve as the available technologies improve; additionally we gather feedback from our students regularly, using the responses to inform future developments. We hope to continue in this way ensuring that our students benefit from a carefully crafted course that makes appropriate use of current e-learning research and associated technology.

References

[1] C. A. Chaves. On-line course curricula and interactional strategies: The foundations and extensions to adult e-learning communities. *European Journal of Open and Distance Learning*, 2006.

[2] N. Dabbagh. Pedagogical models for e-learning: A theory-based design framework. *International Journal of Technology in Teaching and Learning*, 1:25 – 44, 2005.

[3] P. Goodyear, S. Banks, V. Hodgson, and D. McConnell. *Research on Networked Learning*, pages 1 – 10. Springer, Norwell, Massachusetts, 2008.

[4] B. Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26:251–321, 1985.

[5] Hopgood and Hirst. Keeping a distance-education course current through e-learning and contextual assessment. *IEEE Transactions on Education*, 50:85 – 96, 2007.

[6] Kolb. *Experiential Learning: experience as the source of learning and development*. Prentice-Hall, New Jersey, 1984.

[7] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice-Hall, 2001.

[8] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. *The Photogrammetric Journal of Finland*, 21(1):97–109, 2008.

[9] Weller and Hopgood. Implementing a learning model for a practical subject in distance education. *European Journal of Engineering education*, 22:377–387, 1997.

An Open Platform for Teaching and Project Based Work at the Undergraduate and Postgraduate Level

Benjamin N. Passow, James Wheeler, Simon Coupland, and Mario A. Gongora
Centre for Computational Intelligence
Faculty of Technology
De Montfort University
LE4 1BH, Leicester, UK
Email: {benpassow,jw,simonc,mgongora}@dmu.ac.uk

***Abstract*—Robots are a great tool for engaging and enthusing students when studying a range of topics. De Montfort University offers a wide range of courses from University access courses to Doctoral training. We use robots as tools to teach technical concepts across this wide and diverse range of learners. We have had great success using the Lego RCX and now NXT on the less demanding courses, and conversely with the MobileRobots Pioneer range for postgraduate and research projects. Although there is a distinct area in between these two where both these platforms meet our needs, neither is suitable for every aspect of our work. For this reason we have developed our own hardware and software platform to fulfil all of our needs. This paper describes the hardware platform and accompanying software and looks at two applications which made use of this system.**

Our platform presents a low-cost system that enables students to learn about electronics, embedded systems, communication, bus systems, high and low level programming, robot architectures, and control algorithms, all in individual stages using the same familiar hardware and software.

I. Introduction and Background

Robots and control systems have become essential parts of modern industry and are increasingly used in education. Within many teaching curricula, pupils are often introduced to robots at the primary school stage, where they learn concepts such as direction, angles, measurement and sequencing. At this level, Roamers [1], Pixies [2], and BeeBots [3] are popular choices due to their simple programming interface and “friendly” appearance.

At a higher level, students may make use of their theoretical knowledge by applying these to a real world machine [4]. General computer science as well as robotics and artificial intelligence students begin to explore the mechanics of robot design, constructing their own robots and adding sensors and actuators to suit a particular challenge. In this format there is generally some form of processor unit or brain that contains the control instructions and connects to the sensors and actuators. The control software is often developed on a standard PC and then uploaded to the controller via a communications link. Common choices for this format are the Lego Mindstorms [5] RCX and NXT and the Robix Rascal [6]. This showed to be effective for motivating students in practical activities [7].

For teaching software processes relating to control systems, it is often desirable to employ a robot platform with standard actuators and sensors (e.g. having motion, vision, hearing, proximity detection etc) with an embedded PC as the central control processor. In this environment, students learn to write control software that uses the underlying operating system to communicate with the available sensors/actuators. Examples of such robot platforms include the MobileRobots Pioneer and Peoplebot [8].

At De Montfort University, whilst we have found the Lego Mindstorms kits and the MobileRobots equipment to offer extremely useful platforms for the various teaching courses offered, there are some concepts; such as electronic design and embedded programming, that neither platform allows us to teach in the way we would like. For this reason we have developed our own PCB with an onboard Microchip microcontroller and several I/O connections that easily interface to commonly used actuators and sensors. Since a student version of the Microchip Integrated Development Environment (including editor, compiler, debugger and programmer) is freely available, we may use this as the main environment within which students develop their embedded code. The Microchip In-Circuit Debugging tools are relatively cheap and provide a useful means for interfacing between a host PC and the robot control platform.

By using a modular approach to the design of the platform along with its accompanying electronic interfacing and software libraries, we are able to easily reconfigure the platform according to the nature of the concepts being taught. As an example, for first year students we can provide them with pre-built sensor circuitry and a software library of high-level C functions that enable them to design a simple embedded system whilst shielding them from the lower-level complexities of electronics and software. As the teaching programme progresses, the control platform can be reconfigured so that students are required to design their own electronic interfaces or write their own low-level software in order to accomplish the tasks assigned to them.

This paper describes the development of the platform and software libraries in more detail. We include two case studies highlighting how the platform has contributed to the teaching programme at both first year Bachelor course level and also at Masters and Doctoral training levels. Finally we offer a conclusion that summarises how this approach may be of benefit to other educational establishments with a robotics teaching programme.

II. The Platform

The hardware side of the platform consists of a printed circuit board with voltage regulation, a 16bit Microchip programmable interrupt controller, analogue and digital peripheral input and output pins, two RS232 serial ports, I^2C bus, as well as pulse width modulation and motor control outputs. The PIC is programmed using a commercially available USB In-Circuit Debugger. This section will introduce and discuss the platform in more detail.

A. Hardware and its Components

The design of the board is optimised for mechatronics and control projects. It is based around a Microchip microcontroller dsPIC30F4011, which can run at up to 30 million instructions per second (MIPS), has 48kB program memory, 2kB random access memory, 1kB non-volatile EEPROM memory and 31 I/O ports. The PIC is powered by 5 VDC for the digital power supply, which is regulated by a standard analogue voltage regulator LM7805. We used the TO-92 package to maximise the power dissipation capability so that a range of battery voltages can be used, up to an online-charging lead acid battery at 14.8VDC. Since this board is intended for robotics projects, it is assumed that it will be used with batteries only, not a mains power supply; as such it has no rectifier diodes or large ripple-filtering capacitors at the input. It includes only the compact capacitors required to filter the feedback and noise from the digital clock and circuitry and the power devices that might be connected (e.g. electric motors).

This particular PIC provides three PWM-specific outputs (balanced pairs of digital outputs); two of which are connected to a dual motor driver chip L298N. This provides two full H-bridge PWM direct motor power outputs from the PCB. The H-bridge driver chip provides an interface between the digital supply voltage (typically +5VDC) and the battery voltage (typically +12VDC), which supplies power directly to the motors through the H-bridge. We have tested powering motors from 7-12 volts from different types of batteries (e.g. 7.2V or 9.6V from an array of NiMH, 7.4V from an array of Li-Po and 12V from standard sealed Lead acid), and our system has shown to be quite effective for most applications. All standard protections are included in the PCB so that the students need only connect the motors directly; there is a set of flyback fast switching inverse diodes to ground and power VCC (battery) and capacitor in parallel with the motor. The third PWM set of outputs from the PIC is available for expansions in the projects via a connector in the PCB.

Four of the PIC's signals are dedicated for driving RC-hobbyist servos (pulse position controlled position-servo mechanisms). These position-servos draw the power from the 5VDC regulated power supply to avoid problems when using batteries above 9V, which would be outside the tolerance of such devices (typically designed to work between 4.8V - 9.0V). The outputs from the PIC are connected to four 3-pin headers arranged in the standard Ground-Power-Signal configurations used by most RC-hobbyist servos.

TABLE I
Platform Interfaces

Quantity	Interface name and description
Actuators:	
2	Full H-bridge motor drivers
1	Full-balanced PWM digital output
4	Direct connections to ppm postion-servos
16	Simple digital actuators via I/O ports
Sensors:	
<127	I^2C sensors Available to our students are: • Digital Compass • Ultrasonic ranger • Other boards
9	Analogue sensors (1Msps @ 10bit) Available to our students are: • Light dependent resistor • Inertial measurement unit
16	Digital sensors (various)
Communication:	
2	UART serial ports (RS232 via converter)
1	I^2C bus (master or slave mode)
Expansion:	
17	Additional programmable I/O pins

Finally, there are two more dedicated headers, both intended for communications. One uses one of the PIC's UART pins to connect to a standard RS232 serial port. The pins come directly to the headers so that the digital signals from the PIC are available directly, i.e. there is no RS232 level-converter driver on the PCB. This allows connecting directly to other digital serial ports. If a standard serial port is going to be used (e.g. to connect to a computer) then an external RS232 level converter (e.g. MAX232) is required. We have various mini-PCBs with a MAX232 already mounted for use in various projects. The other communications header provides digital signal connection to the I^2C port from the PIC. This is mainly used for connecting to peripherals such as ultrasonic rangers, electronic compasses or IMUs. The addressable structure of this serial bus allows multiple devices to be connected and it has proved to be very useful and versatile as there is a vast range of peripherals, sensors, etc. that are available commercially and at low cost using this protocol.

The remaining I/O pins of the PIC are connected to a general-purpose header, which the students can use to connect any other type of peripheral or device not covered by the other headers mentioned above.

This convenient and compact design provides the optimal configuration for robotic and control projects. Table I summarises the platform's available interfaces for the students to use.

B. Development Environment and Tools

The microcontroller is programmed and can be debugged using Microchip's in-circuit debugger ICD2. This device is connected via USB to the host machine running the integrated development environment called MPLab. The standard programming language that comes with this development environment is assembler. In order to program with a high level programming language, an additional cross-compiler is required. We use Microchip's C30 compiler which is freely available for research and students projects. The compiler is fully ANSI compliant and includes a set of libraries for easier device configuration and use.

III. Software Libraries

To enable students new to programming and robotics to work with the platform we have written a set of high level functions for them to use. This Section details some of the software libraries that provide simple software interfaces to functionality such as timers, sensors, communication, and motor control.

A. Timers

At the heart of any embedded controller is a timing system, our system is no different. Our API supplies four basic functions which can be combined to give all timing functions necessary:

```
// Initialise timer device
void          timePassed_init(void);
// Reset timer device
void          timePassed_reset(void);
// Get elapsed time (ms) as a uint
unsigned int timePassed_ms(unsigned char);
// Get elapsed time (s) as a uint
float         timePassed_fs(unsigned char);
```

The function `timePassed_init` sets up the timer by setting the relevant configuration bits on the PIC's timers. This function must be called before the other timing code will work. The function `timePassed_ms` returns the elapsed time in milliseconds as an integer whereas `timePassed_fs` returns the elapsed time in seconds as a floating point number. Elapsed time in both these functions is a measure of how much time (measured using processor clock cycles) has elapsed since the PIC timer was reset. The PIC timer is reset by four possible actions:

- Calling `timePassed_init()`.
- Calling `timePassed_reset()`.
- Calling `timePassed_ms(1)`.
- Calling `timePassed_fs(1)`.

Although the initialisation function must reset the timer, we also provide the explicit `timePassed_reset()` reset function. Additionally the timer may be reset when measuring the elapsed time by calling the relevant function with a parameter of 1. These functions provide a simple interface for measuring time in milliseconds and seconds.

B. Analogue to Digital Converter

The ADC provides access to readings from analogue sensors connected up to our embedded system. Our API provides four functions for controlling and accessing the sensor readings from the ADC:

```
// Initialise ADC
void myadc_init(void);
// Start the ADC reading timer
void myadc_startReadings(void);
// Stop the ADC reading timer
void myadc_stopReadings(void);
// Read data from the ADC
int  sensorReading(char sensorNumber);
```

The ADC needs to be initialised, this is done by calling `myadc_init(void)`. The initialisation routine sets up a timer driven interrupt system which reads data off the ADC according to a timer which can be controlled through the API. The timer is started and stopped using the `myadc_startReadings(void)` and `myadc_stopReadings(void)` functions. When the timer elapses it causes an interrupt routine to run with regular frequency. The interrupt reads data from the ADC to a predefined data structure via a mean of two filter. This data can be accessed through the `sensorReading(char sensorNumber)` function. This is in effect an interrupt-driven polling system – the ADC is polled with a regular frequency as designated by a timer. It is worth noting the the polling timer causes interrupts to by raised, meaning that although the ADC-API uses a polling system this could be modified to a pure interrupt driven system fairly easily.

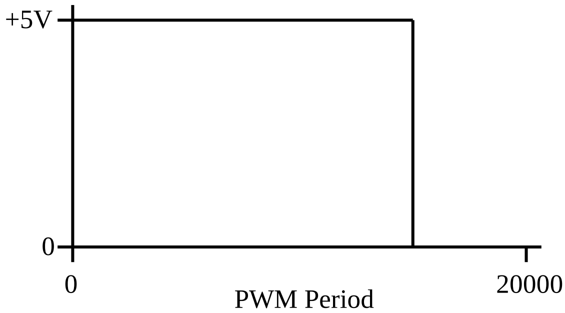
C. Motor Control

The motors are controlled using a standard pulse width modulation approach, taking into account that an H–bridge motor driver is used. Two duty cycle registers are utilised, one for each motor, with forward and reverse control. Figure 1 depicts the forward and reverse control of a single motor using PWM through an H–bridge motor driver.

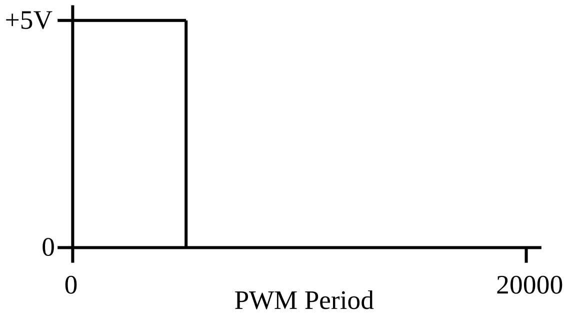
Our API provides three functions for controlling the motors:

```
// Initialise the motor control system
void MotorControlPWM_Init(void);
// Set the motor speed off both motors
void MotorSpeed(int motorLeft,
                int motorRight);
// Turn a choice of motors off
void MotorOff(int choice);
```

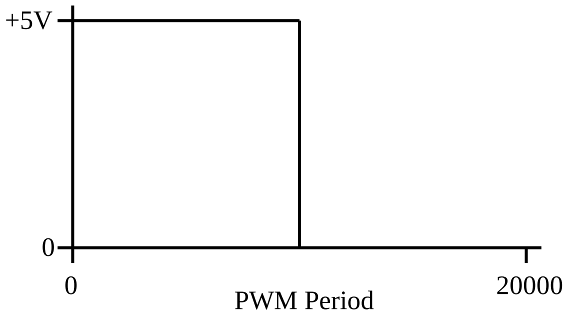
The `MotorControlPWM_Init()` function needs to be called before motor speeds can be controlled. This function sets up the two duty cycle registers and organises the relevant pins for PWM output. The `MotorSpeed(left, right)` function takes integers as percentage values i.e. calling `MotorSpeed(-25, 75)` causes the left motor to turn in reverse with 25% power (not speed – generally power to speed is a non-linear relationship) and the right motor to turn forward with 75% power. The `MotorOff(choice)` function turns off one or more motors when passed one of three constants: `MOTORLEFT`, `MOTORRIGHT` or `ALLSTOP`. If the function is called with `MOTORLEFT` or `MOTORRIGHT` then the respective motor is stopped with a powered stop (see Figure 1(c)), if called with `ALLSTOP` then



(a) 50% Forward Power.



(b) 50% Reverse Power.



(c) Powered Motor Stop.

Fig. 1. PWM Motor Control with an H–Bridge.


(a) KITTDASH9 On a Sumo Arena.

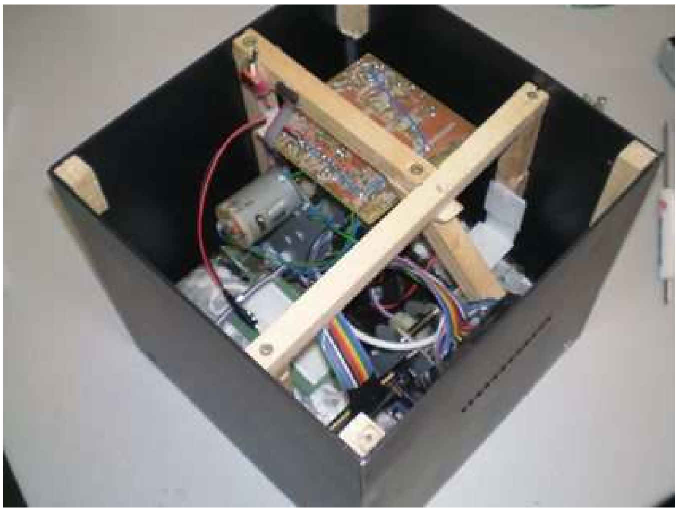

(b) The Interior of KITTDASH9.

Fig. 2. The KITTDASH9 Sumo Robot.

PWM is switched off (PWM timer base is disabled), switching off power to the motors and letting the motors drift.

IV. Application Case Studies

The platform introduced in this paper has been used in a variety of projects including an inverted pendulum robot, balancing weight robot, an autonomous Dr Who Dalek, a sumo fighting robot and an autonomous helicopter. We focus on the latter two for our application case studies of the hardware and software as they are on the opposite ends of the higher education spectrum.

The first case study looks at a robot built by first year students on our Artificial Intelligence and Robotics Bachelors degree. This robot took part in the standard sumo competition at the 2009 Robot Challenge in Vienna. The second case study investigates how a compact version of the same system was used to control an autonomous helicopter for a Masters dissertation and later on in a PhD project.

A. Sumo Robot – KITTDASH9

KITTDASH9 was built by a group of first year undergraduate students studying Artificial Intelligence and Robotics at De Montfort University. The students built the robot within the robot club which runs once a week and not during formal teaching time. The robot was designed and built to be entered in the standard class of the robot sumo competition at the Robot Challenge 2009. Figure 2 shows the KITTDASH9 including the mounted embedded system (notice it is mounted upside down) and drive train.

The robot has four custom built light intensity sensors, one on each corner and a modified serial ball mouse to provide a basic form of odometry. The robot has no range finding or bump sensors. Locomotion is provided by two independently driven tracks fitted with a high traction rubber surface. The robot is fitted with a lighting effect system consisting of an array of red LEDs controlled by a separate PIC which is

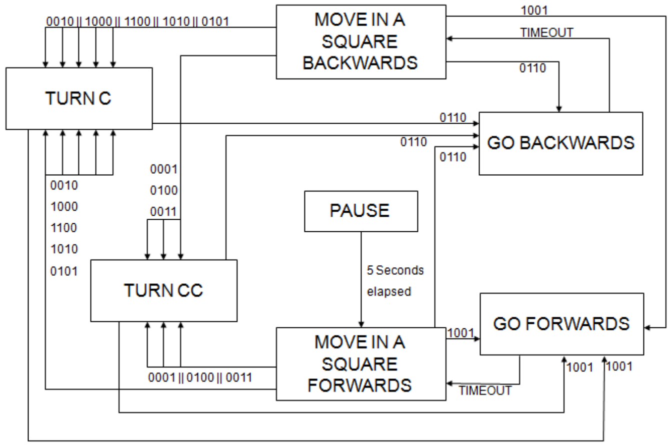


Fig. 3. The Finite State Machine Control Architecture as a State Transition Diagram.

Fig. 4. Autonomous Helicopter Flyper based on our Proposed Platform

TABLE II
MICROSOFT SERIAL MOUSE PROTOCOL [9].

	D6	D5	D4	D3	D2	D1	D0
1st word	1	LB	RB	Y7	Y6	X7	X6
2nd word	0	X5	X4	X3	X2	X1	X0
3rd word	0	Y5	Y4	Y3	Y2	Y1	Y0

connected to the main embedded system being discussed here.

The students implemented a finite state machine control architecture, as depicted in Figure 3. Each state has a clear control objective which is implemented through a combination of the timer and motor control functions from our API. Transitions between the states are enacted by a combination of states from the light intensity sensors, given on the state transition diagram as a binary string, for example 0101. Notice the light intensity sensors give binary readings. The students achieved this by taking readings from the light intensity sensors, using the ADC part of our API, and putting them through a hard limiter to decide whether the sensor is over a white surface or a black surface – the only two surfaces the robot will encounter during a sumo battle. Each sensor has an individual hard limiter threshold, allowing each sensor to be individually calibrated.

As mentioned earlier, KITTDASH9 is fitted with a modified serial mouse. Although the students did not manage to use this sensor in their control process, they did (with significant help) manage to get readings from the mouse unit. The mouse was connected directly to the second serial connection on the embedded system. As the mouse ball moves, events are generated and data giving the amount of motion in the x and y axis are sent on the serial bus. Each event consists of three 7 bit words (see Table II) and the motion reading must be decoded from these three words as given below:

δx = word1 & 0x03 << 6 + word2 & 0x3F

δy = word1 & 0x0C << 4 + word3 & 0x3F

Most of the code to read the serial port was written by the authors, however the students had to decode the readings from the mouse. This meant they got practical experience using bit masking and bit shifting; both of which are taught to students, but rarely covered in practice.

The robot was finished on time and the code written mainly by a group of first year undergraduate students. This would not have been possible without the pre-built embedded system and programming API ready to use. Unfortunately the robot only performed moderately well in competition, it appeared to be under powered compared to its rivals. The high traction rubber meant the robot defended well but it lacked the power to push opposing robots out of the arena.

B. Autonomous Helicopter – Flyper

Our proposed hardware and software platform has also been used to create an autonomous helicopter called Flyper. This robot, as shown in Figure 4, has been built by a post graduate for his Master of Science dissertation and later on used in his Doctoral training. The robot's embedded system and software architecture are like the platform design introduced in this paper but the circuitry has been miniaturised to save space and weight.

In general, helicopters have 3 rotational degrees of freedom (DOF), called pitch, roll and yaw, as well as 3 translational DOF called up/down, left/right and forwards/backwards. The helicopter used in this work is a Twister Bell 47 small indoor helicopter model. It is a coaxial rotor helicopter with twin counter rotating rotors with fixed collective pitch and 340 mm span. The rotors are driven by two high performance direct current motors and two servos control the rotor blades' plane angles. The weight of the helicopter in its original state is approximately 210 grams and it can lift up to 120 grams. Before modification, the helicopter was remote controlled by a pilot handling four controls simultaneously: the amount of lift, heading, pitch and roll.

Due to the limited payload the small helicopter is able to carry, the student reduced the platform's physical size by using a prototyping board rather than a PCB. This reduced the size from 80 x 80 mm to 52 x 33 mm and from 51 grams to 25 grams without heat sinks.

In order to keep the autonomous helicopter at a low cost, the student chose to use standard sensors that were already available to him: sonar distance sensors (SRF08) for measuring altitude and attitude and a digital compass (CMPS03) to determine the heading. The I^2C bus was used to connect and read the sensors using the PIC microcontroller. Figure 4 shows three sonar sensors mounted on the helicopter as well as the digital compass at the far end of the tail.

In order to avoid reflections received from one sonar but transmitted from another, the sensors have been installed at an angle of 10°away from the centre of the helicopter. With this configuration in place and given a flat ground, the attitude of the helicopter can be determined by analysing the difference in measured distances between the sensors. Although the accuracy of the calculated attitude is restricted to the accuracy and resolution of the sonar sensors, the system showed to work as intended.

The PWM outputs together with the L298N motor driver were set to power the two brushed DC motors driving the rotors over a two cogwheel transmission. A small alteration to the circuitry changed the use of the H-bridge as such to using it as a simple driver. This configuration provided the motors with the power required although the motor driver partially reached its peak output current of 4 ampere (e.g. during take off).

Within only three months, the student built an autonomous helicopter that achieved relatively stable flight [1]. Furthermore, during his Doctoral training he used this robot to study the use of evolutionary algorithms to tune and optimise conventional proportional integral derivative (PID) control algorithms directly on the robot [10], [11].

V. Conclusions

In this paper we introduced a low cost platform to be used extensively in the broad spectrum of higher education. The platform can be put together by first year students to learn about electronics, bus systems, and digital technologies. The same students can then program the system using a high level C API. Later on, individual students can build new robots using the existing platform and generate complex programs using Assembler and C. Post-graduate students can use the existing robots to study and compare robots, behaviours, and control architectures.

By using industry-standard components and a modular approach, we have developed a low-cost robot-control platform that may be easily reconfigured to suit some of the general computer science and all levels of the robotics teaching curricula: our platform enables students to learn about electronics, embedded systems, communication, bus systems, high and low level programming, robot architectures, and control algorithms, all in individual stages using the same familiar hardware and software.

Acknowledgment

The authors would like to thank Mr Dilip Chauhan for his help in producing the printed circuit boards.

[1]For test flight videos please visit http://youtube.com/thecci

References

[1] "Roamer official website," May 2010, `http://valiant-technology.com/uk/pages/roamer_home.php`.
[2] "Pixie official website," May 2010, `http://www.swallow.co.uk/pixie/pixie1.htm`.
[3] "Bee bot official website," May 2010, `http://www.beebot.org.uk`.
[4] V. Douglas, "Robots make computer science personal," *Communications of the ACM*, vol. 49, pp. 12–25, 2006.
[5] "Lego mindstorms on wikipedia," May 2010, `http://en.wikipedia.org/wiki/Lego_Mindstorms`.
[6] "Robix rascal official website," May 2010, `http://www.robix.com/`.
[7] J. Adams, S. Turner, S. Kaczmarczyk, P. Picton, and P. Demian, "Problem solving and creativity for undergraduate engineers: findings of an action research project involving robots," in *International Conference on Engineering Education ICEE*, 2008.
[8] "Mobilerobots research robots website," May 2010, `http://mobilerobots.com/ResearchRobots.aspx`.
[9] P. Bourke, "Decoding data from the microsoft serial mouse," April 2003, available at http:// local.wasp.uwa.edu.au/˜pbourke/dataformats/serialmouse/ or on CDROM from http:// local.wasp.uwa.edu.au/˜pbourke/.
[10] B. N. Passow and M. A. Gongora, "Optimising a flying robot: Controller optimisation using a genetic algorithm on a real-world robot," in *Proceedings of the International Conference on Informatics in Control, Automation and Robotics, ICINCO'08*. Madeira, Portugal: INSTICC Press, May 2008, pp. 151–156.
[11] B. N. Passow, M. A. Gongora, S. Coupland, and A. A. Hopgood, "Real-time evolution of an embedded controller for an autonomous helicopter," in *Proc. of the IEEE Intl. Congress on Evolutionary Computation (CEC'08)*, Hong Kong, June 2008, pp. 2538–2545.

ROBINI – Robotic Initiative Lower Saxony

Development of practice-oriented education modules in schools

Julia Kramer
Prospektiv Gesellschaft für betriebliche Zukunftsgestaltungen mbH
Dortmund, Germany

The project ROBINI – Robotic Initiative Lower Saxony develops practice-oriented robotic education modules, which are being tested and implemented within the core curricula of general education schools and vocational schools in the Hanover Region. The practical part is established through school – company cooperation for an applied approach. By embedding the future-proof subject of robotics the Hanover Region is to be strengthened in two ways: Technical skills of students are advanced and they are given a focused occupational orientation so that in the long run the next generation of technical specialized staff will be secured. And through the integration of especially small and medium sized enterprises in the project, they are given additional information on the possibilities of the employment of robotic technology, which could make the difference in competing in a globalized market.

Robotics; education modules; practice orientation; customized application; school – company cooperation; occupational orientation; technically skilled staff

I. NEED FOR ACTION

Against the background of economically relevant technical innovations and an increasing automation of technical processes in industry, the competitiveness in the globalized market especially of small and medium-sized enterprises (SME) depends notably on the availability of specialized staff. Numerous enterprises have already given notice of recruiting problems for technical specialized staff. Analyses of the labor market of the region of Hanover have verified that vacancies especially in the field of mechanical and electrical engineering as well as vacancies in technical production cannot be filled. This negative development is particularly crucial for the region of Hanover and Lower Saxony for its economy is still widely based on these branches.

Mainly three trends are decisive: Firstly, the demographic shift, which will even enforce the lack of specialized staff in the near future. Secondly, the situation that too little school graduates choose technical vocational trainings or courses of study. And thirdly, the present technical staff is often not properly trained according to the practical and forthcoming requirements.

The project ROBINI wants to show that the application of robots can find a remedy here. On the one hand many SMEs are not aware of the possibilities and advantages concerning the implementation of a robotic system: On the one hand the use of robotic systems helps to increase productivity and flexibility, and on the other hand – especially against the background of demographic shift – robots can take on tasks where manpower is missing or simply assist, so that wearing tasks can be carried out longer. Furthermore, the application of robots create diversified, challenging and fascinating work tasks for the technically skilled and specialized staff, which should not be underestimated.

In the face of the augmenting and further increasing relevance of robotics in the working environment, and especially in SMEs, the necessity of a further diffusion of this topic in relevant courses of education is arising, first and foremost the training of practical competences. It is imperative to make education future-proof.

II. THE PROJECT ROBINI IN BRIEF

ROBINI – Robotic Initiative Lower Saxony is a project of the Business and Employment Promotion Office of the Hanover Region in cooperation with Prospektiv GmbH, a research institute in the field of work science. The project ROBINI focuses on the implementation of practice-oriented robotic technology education both in general education schools and vocational schools in the Hanover Region, Lower Saxony.

Overall eight schools with classes up to 30 students take part in the testing over a runtime of three years (2009-2012). Additionally, the project involves a pool of enterprises and is supported by so called strategic partners like the regional Employment Agency, the Leibniz University of Hanover, the Association of German Engineers and the Robotation Academy GmbH.

In this context ROBINI pursues several aims simultaneously:

ROBINI – Robotic Initiative Lower Saxony is a project of the Hanover Region in cooperation with Prospektiv GmbH and is sponsored by the German Federal Land Lower Saxony and by the European Social Fund (ESF).

A. Aims

- *Inspiring young students for natural scientific and technical themes:*

  Despite a growing number of so called MINT (math, informatics, natural sciences, technology)-initiatives trying to increase the number of graduates in these courses, the lack of specialized workforce still persists. Since positive experiences in handling technology are the basic prerequisites for a lasting interest in and advanced insight also into the specialized field of robotics, early playful and hands-on socializing with MINT themes has to be part of the project's aims.

- *Accomplishing a deepened occupational orientation for students in the field of robotics*:

  As a second step, students need orientation! They are to know which skills and qualifications are needed when working with robots, which opportunities of training and further training exist and which are suitable for them.

- *Generating new recruitment opportunities*:

  The practical experience orientation can be called a meta-aim of the project ROBINI. The tight link-up between theory and practice and a close cooperation between different level schools and companies in carrying through the education modules, create the mutual opportunity for future apprentices and instructors to get to know each other outside and in advance of job interviews.

- *Sensibilizing and supporting regional companies in the implementation and utilization of robotic technology*:

  Through the involvement in the project plus additional information and activities especially for SMEs, they are to become aware of the opportunities that a use of robots could bring with regard to business economics, regional economy and employment market.

- *Establishing a regional network of education and further training in the field of robotics*:

  The project ROBINI aims at creating the basis for a regional network. A platform is to be established where theoretical, practical and of course educational experts can connect and exchange knowledge and opinions for mutual learning.

B. Approach

The above mentioned aims will be achieved by an intensive cooperation between schools, enterprises and strategic partners such as the Leibniz University of Hanover or the Robotation Academy. Within the framework of this cooperation education modules with a strong focus on company practice will be developed, sampled and prepared for the transfer to other schools. For this purpose contents, material and particularly practical experiences and advices concerning the realization and implementation will be compiled in guidelines. The emphasis is placed on a specific occupational orientation and on imparting technical and economical knowledge as well as developing skills concerning work organization in connection with robotic technology in companies.

III. Robotics in Education

It can be observed that robotics lessons have already found their way into general education schools. Besides the use of e.g. Lego systems, which are especially appropriate for quick learning progress with young students, the participation in competitions like the "RoboCup" is quite common. Thanks to committed teachers practical education outruns the formal development of curricular. However, these numerously co-existing activities are up to now primarily limited to working groups and there are no consistently regulated robotics lessons. Nevertheless, this trend is again evidence for the increasing relevance of robotic technology.

The ongoing project ROBINI contributes to furthering and above all perpetuating this trend of robotics in education. At the moment six general education schools and two vocational schools participate in developing, sampling and implementing its education modules, aiming at consolidating educational robotics and core curricular.

A. Education modules

The ROBINI education modules are constructed holistically, that means interdisciplinarily and cross-curricularly. They are designed to fit various courses and to be integrated into standard curricular. The modules can be adapted to different types of school, different student ages or different levels of knowledge and ability. In order to achieve this, but also to work out and highlight where diverse aspects should be followed, the topics and approaches of the education modules are regularly discussed by all participating project partners backed by experts of research and practice and evaluated by the project coordinating team.

In detail the education modules include several contents:

- Occupational orientation with regard to robotic technology,
- basic knowledge of robotics,
- practical knowledge concerning the application of robotic technology,

- case studies and feasibility studies with reference to a certain practical example,
- preparation of a check list and guideline for companies that are planning an implementation of robotic technology,
- presentation and discussion of results with representatives from involved companies.

Beside technical knowledge also social and methodical competences of the students are being trained by this interdisciplinary approach.

Furthermore, since practical experience plays a major part not only in designing and arranging the contents of the education modules, but also in the carrying through, involved companies can on the one hand share this practical experience and on the other hand benefit at the same time from compiled information concerning the implementation of robotics and from getting into early contact with qualified junior employees.

B. *Ways of implementation*

As was already mentioned, the modules for general education schools and vocational schools are being realized considering the different levels of knowledge and ability. At the current state of realization the first education modules are being sampled in the lessons in close collaboration with the responsible teachers. In this way the teachers are being capacitated to continue to carry through the modules themselves after the end of the project duration. Where suitable and needed external experts are invited to provide additional input in class, for example in the field of technical details of robots, ranges of application, or in matters of occupational orientation. While the first topics are an example of where schools and enterprises work together, either in school or in the company, the latter topic affords an opportunity for the cooperation between general education and vocational schools.

To give proof of flexibility of the education modules and a little more insight, some examples of implementation shall be described. These are only two ways being tested of many more possible of how to discover and teach "basic knowledge of robotics":

- *Robotics as part of the curriculum for an informatics course in senior class*:

  Other than maybe expected at first glance the subject informatics not only allows programming of robots according to the definition of their tasks, using different methods and different programming languages. Under the heading "hardware" students for instance also get to know the possible sensors of robots and construct them in order to simulate various operations.

- *The obligatory elective subject "Robotics"*:

  This way of implementation is tested in different schools at different class levels – from 5th to 10th grade. The advantage of a newly created subject is obvious: One is free in composing contents. In addition to the above mentioned constructing and programming, which is of course still main interest, topics like trends of robotics in society or advantages and disadvantages of robotics at work are being discussed. And this additional knowledge is precisely what is needed in practice.

In both cases LEGO Mindstorms NXT or fischertechnik ROBO (lines of programmable robotics/construction toys) have been used as primary learning aids.

C. *First experiences and achievements in brief*

At this state of the project ROBINI the basis for evaluation does not yet substantially exist. Nevertheless, we would like to share first experiences and achievements:

- The new issue of robotics is widely accepted at schools, whereas SMEs in the Hanover Region are still more depreciative.
- The approach of students to MINT-themes and robotics cannot start too early. And especially girls get easier conversant with robots, if they can experiment in sex segregates groups.
- The implementation of robotics in regular education, i.e. within the core curricular, is possible, but it requires the customized preparation of the topics for the needs of the different types of school.
- So far the implementation and tie to the following subjects has been successfully tested: informatics, politics and labor study.

IV. Further Need for Action

One of the important issues for the project ROBINI is that the first playful approach to robotic technology by means of LEGO Mindstorms NXT or fischertechnik ROBO has to be transferred to an entrepreneurial reality. To bring together the different actors and to inform especially further companies about the project and the chances of cooperating with schools and of implementing robotic technology, a series of workshops is being held in the project. In order to create a curricular for robotics in education, which in the long run can be applied by any type of school, an overall exchange of experiences is being initialized and shall be evaluated at the end of the project ROBINI.

Robotika.SK Approach to Educational Robotics from Elementary Schools to Universities

Pavel Petrovič[1], Richard Balogh[2], Andrej Lúčny[1]

[1]Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia, ppetrovic@acm.org, lucny@microstep-mis.com.

[2]Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 81219 Bratislava, Slovakia, balogh@elf.stuba.sk.

Abstract—**The Association Robotika.SK organizes and participates in a wide range of activities, projects, contests, events, workshops, summer schools, seminars, prepares educational materials, builds educational hardware and software platforms. This article presents our viewpoint on educational robotics, the challenges, tasks, goals, and means of achieving benefits for the learners and teachers. We summarize several years of experience we collected and provide perspectives on the future development, and some of our future plans.**

Keywords - educational robotics, robotics contests, robotics summer schools

I. Introduction

In a happy society, people get the chance to work on what they believe in. We believe in robotics, and we think that we can also improve the chances of others who share our common interests. We believe that robotics technology can help humans to avoid arduous, repetitive, dangerous, and unpleasant tasks, we believe that robotics technology does and will allow us to reach beyond our current horizons, both in microscopic and macroscopic worlds, but within our environments as well. Robots may help rescue people, animals, and other living creatures in critical situations. Robotics can be applied to make our environment cleaner. Robotics technology might bring easier, cheaper, more versatile and flexible solutions to various common tasks. Moreover, we believe that robotics technology can contribute to improvement of the educational process in schools, it can provide entertainment, and for young people, it can be the reason for spending of lot of their time in a valuable and useful way. Robotics can also attract many young people to the fields of science and technology. We also honestly believe that reasonable application of robots in production process will not take the work from people and generate unemployed. On the contrary, the resources saved by cheaper production can be used to give better and more interesting work to the people, in more comfortable working conditions. We think there are not enough robots around us and large efforts are needed to bring them here. We founded the association Robotika.SK, a non-profit, non-political and non-governmental organization, and we use it as a platform for organizing cooperation of institutions of higher education, preparing seminars, talks, summer schools, competitions, initiating, coordinating and realizing various projects, supporting schools, and individuals. All activities are centered around our information website robotika.sk that always brings up-to-date news from the activities organized by us and our partners, as well as robotics news from our region, and outside. In this article, we give an overview of our past and current activities. The following sections describe our viewpoint on educational robotics, the overall structure of our activities, cooperation, individual robotics projects, student work, seminars and talks, summer schools, contests, and public presentations.

II. Educational Robotics

The omnipresence of technology today is a fact. However, a traditional view prevails, namely that technology is still completely dependent on us. Mobile phones, portable computers, digital assistants, intelligent security systems, automatic vending and money transfer machines, advanced technology in production – everything remains fixed at a single place where it was installed, or wherever we take it with us. Soon, however, the technology will start to move in our environments on its own. Automatic delivery, monitoring and service, personal assistants, cleaning, guiding, shopping, and many other tasks will be performed by autonomous mobile devices working on our behalf. And even those that will still be fixed, they will be able to act more autonomously and take smart decisions in dynamic environments as contrasted to being pre-programmed to a fixed sequence of operations.

Many of the tasks named above are performed by robots already today and we must get prepared for this forthcoming age. In particular, we must:

- make sure people will be able to understand the mode of operation of these devices;
- make sure people will be able to control, and even program such devices to utilize their potential;
- prepare enough skilled engineers, who will be able to create them, and provide the necessary service;
- keep building a sufficiently large community of professionals in all related areas, which are important for the progress of development of robots – material science, energy science, physics, electronics, mechanical engineering, communication and human interaction, computer science and technology.

This is why we need educational robotics today, to foster the progress and development, and to avoid stagnation and crises. Every meaningful application of the robotics technology in any form of educational process is a valid contribution. The following ideas have been tried and implemented:

- organizing robotics summer schools and summer camps
- organizing competitions with robots
- building hobby-robotics clubs, labs, and free-time centers
- teaching programming with robots
- using robotics to explain and elaborate on mathematics
- using robots in teaching physics and science
- setting up interdisciplinary student projects utilizing robots
- developing special courses with introduction to robotics
- implementing lectures about robotics into various courses
- building robotics hardware and software platforms
- using robots as educational toys from very early age
- developing art projects and presentations with robots

We believe all these ways are useful and important ways to increase the competence of the general population and specialized students, and we think there are large unfilled spaces in particularly in finding and developing new platforms with completely new features, approaches and ideas. We argue that even though it is important to support the main-stream product lines such as LEGO Mindstorms NXT, it is also important to search and support different systems. Still, only very little has been done on larger-scale parallelization, modular architectures, non-conventional kinematics, and other areas. We will continue our attempts to actively contribute to at least some of them.

III. Structure of Our Activities

We are a small group of scholars and students with some links with industry. We maintain a student and research robotics laboratory. In our institutions, we teach a few courses related to robotics, and outside of them, we try to maintain robotics clubs in primary or secondary schools. We participate in organizing various relevant activities that are initiated or organized by us or our partners. We mention both kinds for completeness. Our activities spread across several levels:

- events for general public, where everybody can register and visit, the aim is the popularization of science and technology;
- events for schools, where participants (i.e. teams) from schools can register, and participate, these events have a more specific target group and thus can be better tailored for their audience;
- events for selected students from technical universities, for instance organized in cooperation with student networks of technical universities;
- events for students in our institutions, these are local events, tailored for our students;
- activities focused on development of robotics platforms and projects that are made publicly available for those interested;
- student projects in a form of bachelor or master theses, or other types of student projects including students from secondary schools;
- publishing information, articles, and materials related to robotics technologies, methodologies, etc.

A description of individual activity types follow.

IV. Cooperation

Our group is built on cooperation. In the very beginning it put together people from three different institutions: two universities and one private company producing sensor technologies, Microstep-MIS. However, many of our activities would be impossible without efficient cooperation with our partners, who include student organizations (BEST), non-profit associations (e.g. InnoC from Austria, Slovak Society of Electronics, Robotika.cz), foundations (e.g. Children of Slovakia Foundation), primary and elementary schools (e.g. Spojená škola Novohradská, Spojená škola sv. Františka z Assisi, ZŠ Karloveská 61), and private companies (e.g. RLX, Microstep s r.o., AVIR, Freescale Semiconductors, and other). We also cooperate with variouis individuals, for instance, the author of the RoboSapien Dance Machine Project, local hobby photographers who needed a robot-operated camera, or artists who are exploring new art forms utilizing technology.

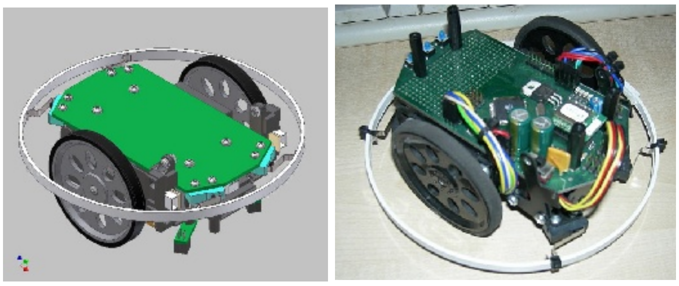


Figure.1 Sbot robot platform with BlueTooth radio communication, autonomous control, line sensors, bumpers, installable IR proximity sensors and encoders, with easily exendable circuit board.

V. Individual Robotics Projects

Various robotics projects represent the type of activities we put our emphasis on, and when also our learning is most intensive. The projects are often developed in cooperation with students, or they are student projects. Sometimes the projects overlap with the contests, when we work as team leaders, or supporters who provide the background, equipment, and guidance. Here, we would like to name a couple of example projects:

Robotnačka v.2 – the drawing robot, controlled from LOGO language

This is a hardware platform, which can be attached to a Logo turtle, which is normally only drawing on the screen. In this way, learning programming becomes much more entertaining, and the robot can also be used for new activities, utilizing its sensors, for example, teaching geometry [1,2].

Robotnačka drawing shapes based on bitmap image

A secondary school student software project: the application received a bitmap image on its input, extracts countours from the image and generates a trajectory to be drawn by Robotnačka [3].

Remotely-Operated Robotics Laboratory

A permanent installation of robots in a laboratory that is always available on the Internet. The robots in the laboratory can be controlled the same way as locally connected Robotnačka – from Logo language, or, alternately, from a web browser, or C++, Java, or another type of application [4].

S-bot and Acrob robot platforms for education and projects

Platforms that were developed in our group for the purpose of simple robotics experiments, bachelor theses, exercises on locomotion and navigation [5, 6], see Fig. 1.

Remotely controlling WowWee family robots

A USB device for sending arbitrary IR signals that could be used to control RoboSapien and other WowWee robots [9]. We also developed a solution for controlling the robots using LEGO IR tower and directly from RCX programmable brick, see Fig. 2.

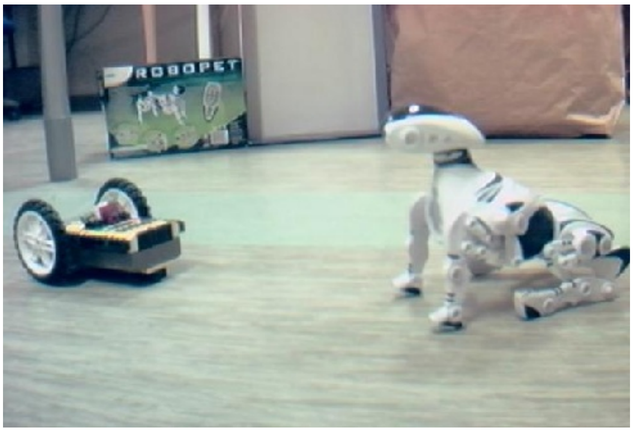


Figure 2. Controlling RoboPet from RCX using IR signals.

VI. Student Work

We use the robotics laboratory to provide the bachelor and master students with a working environment, and the required equipment. In our courses, students get hands-on experience in using robots of different types – LEGO NXT robots, BoeBot robots, Robotnacka, Acrob and Sbot robots. In these exercises, they learn basics about kinematics, signal processing, sensor types, calibration, and control. In the last two years, the following bachelor theses have been successfully completed:

Probabilistic mapping in remotely-operated robotics laboratory (2009)

Bayesian Robot Programming (2009)

SBOT Sokoban (2010)

Localization using distance sensors (2010)

Mobile robot for category line-follower (2010)

and the following diploma theses:

Representations in Evolutionary Design (2005)

Visual Programming of Control System for a Colony of Robots (2007)

Robotic laboratory experiments for secondary school physics (2010)

Cellular Embryogenic Representations for Evolutionary Design (2010)

Didactic materials for the topic robotics construction sets and Imagine Logo (2010)

The exact references can be found at our wiki page [7]. Currently, several other bachelor and diploma theses are in progress.

VII. Seminars and Talks

Our group runs an internal seminar for students and researchers, but more importantly, we invite various speakers to give lectures on topics related to robotics. For instance, we organized a talk about chemical robots (Doc. Štěpánek from VŠCHT Praha), and a talk about Constructionism and Robotics in Schools (Prof. Alimisis from School of Pedagogical and Technological Education in Greece).

Even though our organization is not educational by the definition, one of the best results achieved in previous years is participation in the international project Centrobot, where some joint Austrian-Slovak lectures for the students of secondary schools both from Vienna and Bratislava were organized, Fig.3. There is a big potential of increased motivation of the students from different countries to work on joined robotics projects together. This allows them not only to acquire the knowledge and skills, but also to gain a different perspective, open their minds, and compare their own performance with others.

Figure 3. A joint Slovak-Austrian lecture, Vienna, February 2010.

VIII. Summer Schools

For several years, we have been organizing an event called “Robotic holidays”, a one week intensive lab work with lectures and talks. Typically in the beginning of the summer or in September, interested students and people joined us to work on several more or less challenging robotics projects. During the last three years, together with the student organization BEST (Board of European Students of Technology) and InnoC (Austrian Association for Innovative Computer Science), we organized a summer school for students from technical universities across Europe – twice in Bratislava and one time in

Vienna. This two-week course includes lectures, workshops, excursions, and leisure activities. Fig. 4 shows a group work from our robotics summer school in 2010.

Figure 4. Centrobot robotics summer school 2010.

IX. CONTESTS

Contests are very central part of educational robotics, and they cost a lot of our time and energy. The main advantages of contests are:

- a fixed deadline – improves planning skills, makes it easier to prioritize and focus
- a clearly specified task, which was selected by experienced people in such a way to be solvable, non-trivial, and interesting
- often a standardized platform with a broad user base, which allows good access to information, saves time and efforts
- the possibility for the participants to compare their skills with their peers
- the school or club can make itself visible, this is a great motivation to produce an excellent result
- a nice possibility for building social and professional networks
- contests have a healthy competitive and sporting atmosphere, everything is subordinated to allow a perfect result of everybody

A. Istrobot

Istrobot is the primary contest of our association, where we are the main organizers. The tradition dates back to the year 2000 and a permanent quality growth can be observed. At the present time the contest consists of four different categories: The Pathfollower for linefollowing robots, Micromouse for maze solving robots, MiniSumo for fighting robots, see Fig. 5, and Freestyle for everything else, see Fig. 6. The contest is attracting approximately 100 robots each year and only our internal limits stopped its additional growth. The best experience from this contest is that it really fosters the development of the mobile robotics in our region. With a surprise, we find many research papers in local conferences inspired with robots solving the maze, or line-followers.

B. RoboTour

RoboTour is an outdoor robotics contest organized by the Czech association Robotika in Czech Republic. In the year 2010 it goes international for the first time and it takes place in Slovakia, Bratislava. Design of an autonomous intelligent vehicle appropriate for such contest inspired by the famous Grand Challenge contest is challenging for our university partners and our material support is very useful. Participants from universities and clubs in Czech Republic and Slovakia (wish one exception of a foreign team) compete in autonomous outdoor robot navigation in a leisure park. Robots are allowed to use both global navigation – such as GPS, compass, accelerometers, inclinometers, etc. and local navigation such as ultrasonic distance sensors, laser range sensors, landmarks. Vision is typically the most important component, responsible for keeping the robot on the track, which is necessary to prevent an instant “game-over”. This contest serves also as a good reference testing platform for various image processing problems and generates many interesting solutions of the navigation problems. Members of our association have participated in RoboTour for about four years, and this year, our association has received an invitation to organize the contest in Bratislava.

Figure 5. MiniSumo dead-match at Istrobot 2010 contest.

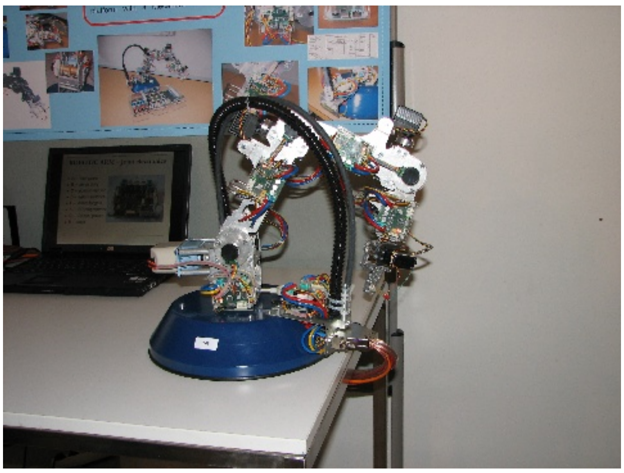


Figure 6. Robotic Arm – Freestyle competition, Istrobot 2010.

C.FIRST LEGO League (FLL)

We are actively involved in organizing regional tournament of FLL in Bratislava that is taking place for the third time this year. It is a contest for teams of 5-10 members in the age of 10-16 years. The strengths of this competition lie in the focus on creativity and team work, excellent preparation of tasks, which are solved by tens of thousands of students round the globe. It is also important that every year, a completely new challenge is to be solved, and thus it is impossible to participate with the same robot year after year. In consequence, also excellent novice teams have a high chance of succeeding. In addition to building and programming the robot, the competition requires completing a research project and preparing a presentation. In this way, the young people get a taste of what it means to be a researcher. However, here we also see some weaknesses. In particular, the research themes are too complex to comprehend for that young people. We would like to see themes that would pose challenges appropriate to their age. For instance, many interesting small research projects in physics and chemistry at the level of elementary school can be completed to demonstrate interesting phenomena. Such experiments are genuine and achieve what they claim, answer the research question completely, and understandably. This is in contrast with typical FLL research projects that, for instance, propose to build dams, reorganize city traffic system, or find cures to diseases... That type of projects resembles somewhat the concept of "Let's pretend" society, where fridges, TV sets and CD players stop to work two weeks after the expiration period. In our local contest, we try to guide the coaches to lean towards easier projects that correspond to the knowledge level of the children. Our association not only participates in organizing the contest, but also provides equipment and staff to the participating teams. Fig. 7 shows a view from our local FLL contest.

Figure 7. FIRST LEGO League regional tournament in Bratislava, 2009.

D.RoboCup Junior (RCJ)

RoboCup Junior is a world-wide educational initiative targeted at young people up to age of 19 years. There is less team work focus in RCJ, individual teams are not an exception. There are also no restrictions on the material and software used as they are in FLL. Succeeding in RCJ (except, perhaps in the RoboDance category) requires several years experience, and advanced technical skills. Access to the information and guidance is a bottleneck, teams guided by students from technical universities or skilled engineers working in relevant industry have an advantage compared to the teams from schools in the countryside. Despite these shortcomings, we feel that RCJ in Slovakia contributes greatly to the interest in science and technology, it leads hundreds of young people through the experience of larger project, and it is a popular contest with good spirit. Our association supports this contest by all means. Fig. 8 shows a scene from RCJ in Slovakia. All information can be found at [8].

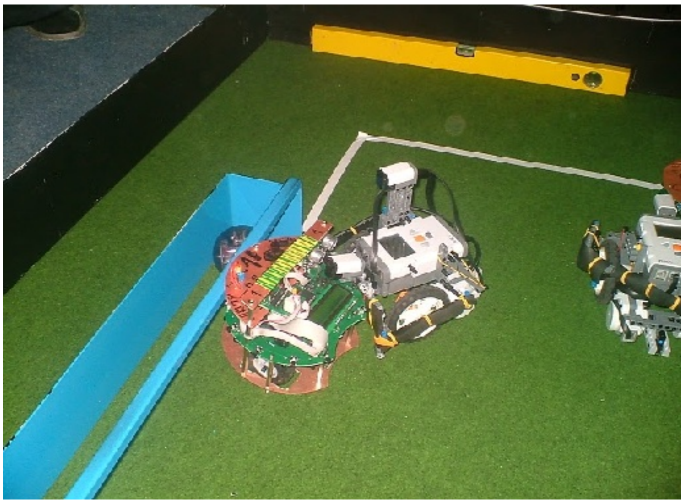


Figure 8. From RoboCup Junior Slovakia, February 2009.

E.Freescale Race Challenge (FRC)

This contest is an initiative of the Freescale Semiconductors company and its goal is to make use of the accelerometers to control the speed of the racing cars autonomously. We supported two student teams with material and advices to actively participate in this contest, see Fig. 9. Resulted autonomous cars are very good attractor also for public presentation and were used in Istrobot contest and in Elosys trade presentation. This contest is also a very good motivational tool to study embedded systems hardware and sophisticated methods of signal processing and even learning and mapping of the toy racing car track.

Figure 9. Freescale Race Challenge, Žilina, 2010, a team from STU Bratislava.

F.Robot Challenge

Robot Challenge is the World's largest contest (from the point of view of the number of registered robots). It takes place in Vienna, and it is organized by InnoC, one of our cooperating partners. Robotika.SK always both actively cooperates and participates in the contest. We have an active exchange of participants between the Istrobot and Robot Challenge contest, which have similar categories. This exchange – best described by "a bus of Slovak participants arriving to Robot Challenge" supports Robotika.SK funding of Slovak-Austrian cooperation.

X. Public Presentations

When possible, we try to present our results to the public. We participate and support presentations of our alma maters at the annual trade show EloSys in Trencin, where we occupy a booth with robots presenting their behavior for visitors. Usually the school groups are attracted and hopefully also motivated for additional studies of technical disciplines. We also participated on the Researchers Night's – a EU coordinated science popularizing project, see Fig. 10. Our presentations were also the part of various international events as the festival of cocktail robotics in Wien Roboexotica, Eurobot national contest in Prague, etc. These are important events for creating new contacts, and attracting young people to the field, which is the aim of our activities in general. Results of the students project and our own platforms make a good jobs here. Moving and operating installations are a base of successful presentation, but the human explanation is always required for public.

Figure 10. Researchers' Night, Bratislava, 2007: vision-guided Boe-Bot that follows a ping-pong ball, and the drawing robot Robotnačka.

XI. Conclusions

Educational robotics is a young field that springs form and connects many different areas. However, it has a place of its own, and it requires separate attention. Not only to prevent repeating the same mistakes, but also to provide a place for exchanging ideas, technologies, platforms, solutions, and a discussion.

In this article, we introduce and summarize the activities and viewpoints of the non-profit association Robotika.SK. We are proud to claim that most of our activities have raised interest in robotics, science and technology among the young people and the target audience. This claim can be supported by the number of participants and students joining our activities and projects and their positive feedback. In the future, our efforts will continue according to the challenges and possibilities we will face, keeping the cooperation and team work as our main working method.

References

[1] Ďurina, D., Petrovič, P. & Balogh, R. (2006), *"Robotnačka - The Drawing Robot"*, Acta Mechanica Slovaca., In Acta Mechanica Slovaca., June, 2006. Vol. 10(2-A), pp. 113-116.

[2] Petrovič, P.: "Mathematics with Robotnacka and Imagine Logo", Eurologo 2005, DrukSfera, ISBN:83-917700-8-7, pp.353-360, 2005.

[3] Boško, L.: "Project Contours", Contribution to the Eucys contest, org. by Mladí vedci Slovenska, December 2007. online project repository: http://webcvs.robotika.sk/cgi-bin/cvsweb/robotika/robotnacka/contours/

[4] Petrovič, P., Lúčny, A., Balogh, R. & Ďurina, D. (2006), "Remotely-Accessible Robotics Laoboratory", Acta Mechanica Slovaca., June, 2006. Vol. 10(2-A), pp. 389-194.

[5] Robotika.SK: Sbot 2.0 – Educational Robot for Clubs and Classrooms, online: http://webcvs.robotika.sk/cgi-bin/cvsweb/~checkout~/robotika/sbot/2.0/doc/dokumentacia_en.pdf?rev=1.1;content-type=application%2Fpdf

[6] Acrob – Arduino Compatible Robot, online: http://ap.urpi.fei.stuba.sk/sensorwiki/index.php/Acrob

[7] Bachelor and Master Theses of Robotics Laboratory @ RoboWiki, online: http://virtuallab.kar.elf.stuba.sk/robowiki/index.php?title=List_of_Master_and_Bachelor_Thesis_Related_to_Robotics_Laboratory

[8] RoboCup Junior in Slovakia, online: http://robotika.sk/rcj/

[9] Controlling RoboSapien using LEGO IR Tower, online: http://www.robotika.sk/projects/robsapien/index.php

On an educational approach to behavior learning for robots

Michel Tokic
Institute of Neural Information Processing
University of Ulm
James-Frank-Ring
89069 Ulm, Germany
Email: michel.tokic@uni-ulm.de

Arne Usadel, Joachim Fessler, Wolfgang Ertel
Institute of Applied Research
University of Applied Sciences Ravensburg-Weingarten
Doggenriedstrasse 38
88250 Weingarten, Germany
Email: firstname.lastname@hs-weingarten.de

***Abstract*—This paper introduces a system for teaching biologically-motivated robot learning in university classrooms that might be used in courses such as *Artificial Intelligence* and/or *Robotics*. For this, we present a simple hardware robot that is able to learn a forward walking policy on basis of a reinforcement signal. Students are able to conduct experiments on a PC with a software called the Teachingbox that controls the robot. This software offers the possibility to control the learning method's parameters throughout the learning process, which allows observing the effects of such parameters on a real robot. Furthermore, learning on the hardware robot is very fast since forward-walking policies are usually learned in about 30 seconds. Due to this quick learning process nearly no waiting time is caused, and in return this fact often impresses the audience and leads to the question: "How does it work?".**

I. Introduction & Motivation

As robots or the environment of a robot become more and more complex, the way of programming robots in the classical supervised way also becomes more difficult. As a consequence, engineers often program just a "working" behavior of a robot, but which can be far away from an "optimal" behavior, e.g. movements of a robot that maximize the forward walking velocity. One possible solution to this general problem is offered by learning behaviors from scratch—in the same way as humans or animals do—instead of manually programming the robot. In literature, learning in such way is called *trial-and-error learning* which has been first studied in the domain of psychology and animal learning [1]. Nowadays, the research domain of *reinforcement learning* (RL) [2] aims to mimic *trial-and-error learning* in a machine-learning approach based on a reward signal (or reinforcement signal) which strengthens or weakens action selections in certain situations with the goal of maximizing the cumulative reward. Since robot learning is just one possible application of RL, the knowledge about this research domain broadens an engineer's skill on behavior programming that can also be applied to other applications.

Neller et. al. said: *"Simple examples are teaching treasures. Finding a concise, effective illustration is like finding a precious gem. When such an example is fun and intriguing, it is educational gold."* [3]. At the University of Applied Sciences Ravensburg-Weingarten, we were looking for such kind of illustration that enables to teach RL within a narrow time-slot of about four lessons of an *Artificial Intelligence* introductory course. Within that given time-slot, we introduce the value-iteration and Q-learning algorithms on discrete state and action spaces and explain the exploration/exploitation problem. In order to explain to the students the field of RL, we found a crawling robot with a simple two-DOF arm as sketched in Figure 1 appropriate that has been proposed by Kimura et al. [4] and built in hardware by Tokic [5]. Furthermore, the reason why we also favor a robot instead of a theoretical problem is due to the fact that robots seem to be encouraging motivators for students as also recently reported by Kay [6].

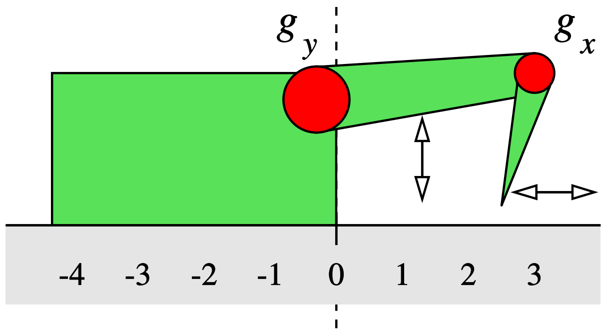


Fig. 1. A model of the crawling robot with its two joints g_x and g_y.

In case of discrete positions and small movement angles of the joints, the state space of the robot arm can be approximated by a grid world. In order to move forward, the robot has to repeatedly perform a cycle of moves as shown in Figure 2 or in the sequence shown in Table I. The task for the learning algorithms is to find a policy (which might be such a cycle) that maximizes the cumulative reward. For this, the reward is the speed of the robot, i.e. the distance that the body of the robot moves forward per time step. Consequently, a move forward gives positive reward whereas any backward move yields negative reward.

In the following we describe the robots architecture and elaborate on simple experiments that students can conduct with a software called the "Teachingbox" [7], which is an open-source framework written in Java. By using this software tool, students are (1) able to send action commands to the robot

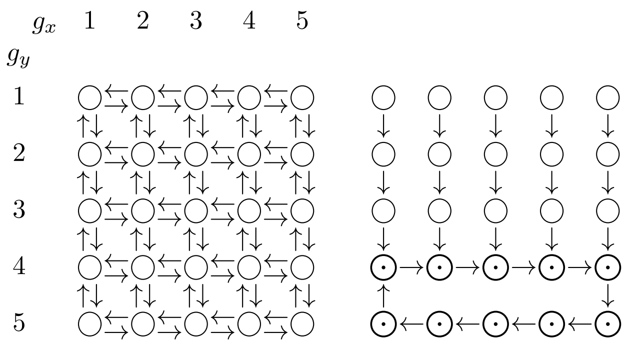


Fig. 2. The 5×5 grid-world model (left) and a cyclic walking policy (right). States within the cycle are labeled as $\odot$.

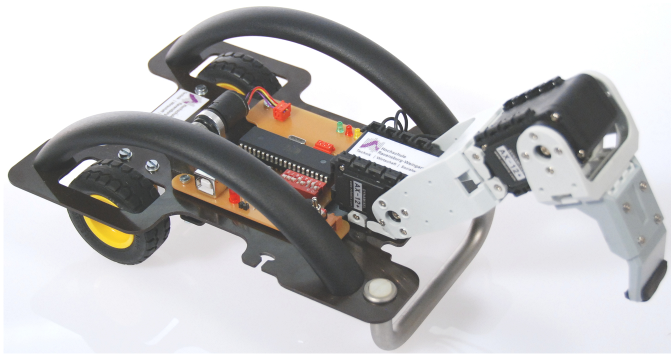
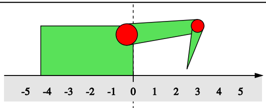
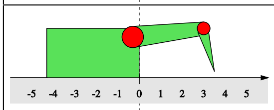
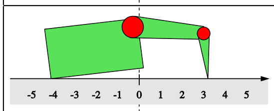
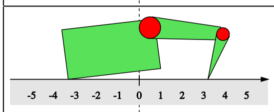


Fig. 3. The crawling robot we use in our laboratory tutorials.

TABLE I
FOUR STEPS OF A SIMPLE CYCLIC FORWARD-WALKING POLICY.

robot	time t	state g_y	g_x	reward x	action a_t
	0	up	left	0	right
	1	up	right	0	down
	2	down	right	0	left
	3	down	left	1	up

in order to observe the robot's behavior and (2) are also able to learn policies on the basis of observed rewards from the robot's environment.

II. Hardware Robot

A prototype of the robot that we use in our "Laboratory on Artificial Intelligence" is depicted in Figure 3. Basically, this robot is controlled by an ATmega32 microcontroller board that is mounted on top of the robot. This board controls the joints of the robot which are driven by Dynamixel AX-12 actuators. These servos communicate with a half-duplex asynchronous packet-protocol on TTL-level with up to 1,000,000 bps. The maximum holding torque is about 1.17 Nm.

The speed of the robot is measured by an optical incremental encoder that is connected via a non-slip belt transmission to a (rigid) wheel axle. The controller board also comes up with outlets for the servos, an outlet for the encoder and a DIP switch for setting up several parameters. For instance, one of these parameters inverts the encoder signal and results the robot to learn a backward-moving strategy instead of moving forward.

On top of the controller board, there also exists a RF04 ER400TRS serial transceiver module, which is used for communication with the Teachingbox software on the PC side. This module is directly attached to the ATMega's serial port and operates by a speed of 19,200 baud. On the PC side we use a RF04 USB telemetry module for communicating with the controller board over a standard RS232 COM port.

III. Reinforcement Learning

We consider the reinforcement learning framework [2] where an agent interacts with a Markovian decision process (MDP). At each discrete time step, $t \in \{0, 1, 2, ...\}$, the agent is in a certain state, $s_t \in \mathcal{S}$—for example, the angular position of the robot's joints. After the selection of an action, $a_t \in \mathcal{A}(s_t)$, the agent receives a reward signal, $r_{t+1} \in \mathbb{R}$, from the environment and passes into the successor state s_{t+1}. The decision which action is selected in a certain state is characterized by a policy, $\pi(s) = a$, that could also be stochastic: $\pi(a|s) = Pr\{a_t = a|s_t = s\}$. A policy that maximizes the cumulative reward over time is denoted as π^*.

In practice, there exist several approaches by which a policy for the robot can be learned. For this, we recently proposed using the value-iteration algorithm [8] with an online model-learning of the environment in parallel which was derived from the *Dynamic Programming* approach. In order to save additional memory required by the model-learning task, we now propose using a different learning algorithm within this paper that belongs to the family of *Temporal-Difference Learning* methods.

In order to learn an optimal policy π^* for the robot, we use Watkins's Q-learning algorithm [9] as depicted in Algorithm 1. This algorithm basically works by assigning a numerical value to each state-action pair (s, a), where each state-action value, $Q(s, a) \in \mathcal{Q}$, is an estimate of the expected cumulative reward, R_t, for following the current policy by starting in state s and taking action a:

$$\begin{aligned} Q_\pi(s,a) &= E_\pi\{R_t|s_t = s, a_t = a\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\} \;, \end{aligned}$$

where $0 < \gamma < 1$ denotes a discounting factor that specifies the influence of rewards received more far in the future.

Furthermore, the parameter $0 < \alpha < 1$ specifies a learning rate that determines how much the value-function estimate is being adapted w.r.t. to the current *temporal-difference error*:

$$\delta = r + \gamma \max_{b \in \mathcal{A}(s')} Q(s', b) - Q(s, a) \ . \tag{1}$$

The affect of both algorithm parameters on the learning process is explored by the students during the conduction of experiments as described in Section V.

Algorithm 1 Q-LEARNING ON ROBOT WITH ε-GREEDY

Initialize Q arbitrarily, e.g. $Q(s, a) = 0$ for all s, a
Initialize start state arbitrarily, e.g. $s \leftarrow (g_x = 1, g_y = 1)$
loop
  $\xi \leftarrow \mathrm{rand}(0..1)$
  if $\xi < \varepsilon$ **then**
    $a \leftarrow$ random action from $\mathcal{A}(s)$
  else
    $a \leftarrow \mathrm{argmax}_{b \in \mathcal{A}(s)} Q(s, b)$
  end if
  select action a
  observe reward r and successor state s'
  $a^* \leftarrow \mathrm{argmax}_{b \in \mathcal{A}(s')} Q(s', b)$
  $\delta \leftarrow r + \gamma Q(s', a^*) - Q(s, a)$
  $Q(s, a) \leftarrow Q(s, a) + \alpha\delta$
  $s \leftarrow s'$
end loop

The robot's action selection policy, which is based on the Q-function learned throughout the interaction with the environment, works as follows. Since the robot is faced with an unknown environment after switching it on, a tradeoff between exploration (long-term optimization) and exploitation (short-term optimization) has to be done [2], [10]. A very simple and commonly used technique for this is ε-Greedy exploration [9], where at each time step the agent selects an action at random with probability $0 \leq \varepsilon \leq 1$ (exploration). With probability $1 - \varepsilon$ (exploitation) the agent selects an action that is greedy with respect to the current value-function estimates:

$$\pi(s) = \begin{cases} \text{random action from } \mathcal{A}(s) & \text{if } \xi < \varepsilon \\ \mathrm{argmax}_{a \in \mathcal{A}(s)} Q(s, a) & \text{otherwise,} \end{cases} \tag{2}$$

where $0 \leq \xi \leq 1$ is a uniform random number drawn at each time step. If there is more than one action having the highest estimated value in state s, a random action of this set of best actions is chosen.

In order to speed-up learning, a commonly used approach is to reduce the exploration rate ε over time. In this case ε is set to a high value at the beginning of the learning process which is decreased by a constant fraction at each time step. This results that the agent is more explorative at the beginning of the learning process, when the environment knowledge is unknown, as later the agent becomes pure exploitative. The final outcome of the learning algorithm where the robot interacted some time with the real world is shown in Figure 4.

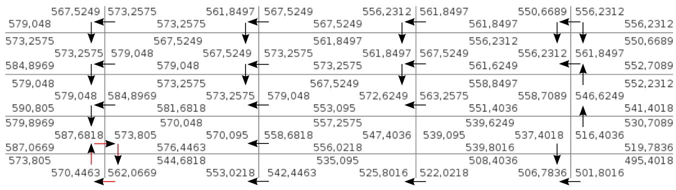


Fig. 4. The Q-values and (greedy) policy learned by Q-learning from a real-world interaction of the walking robot (learned with $\gamma = 0.99$). The corresponding rewards are shown in Figure 5.

IV. THE TEACHINGBOX

When students should learn to understand the behavior of an algorithm, it is didactic supportive to perform experiments with a simple demonstrator. With such it should be possible to easily play with, e.g. in terms of algorithms parameter variations which enable to observe the affect of such parameters on the learning progress. Furthermore, such a demonstrator should also be usable without much effort in order that students focus only on relevant things.

Our recently presented software framework, the Teachingbox (TB) [7], aims at providing a rich library of implemented algorithms for robot learning in a universal robot learning framework. Hereby, the main purpose of this open-source Java framework is to support the development of autonomous agents with learning capabilities. The TB comes up with algorithms for RL, Learning-by-Demonstration, the possibility of manually programming policies and a build-in grid-world editor for modeling simple two-dimensional grid worlds. In particular, the RL-part of the TB currently consists of implementations of the most popular learning algorithms such as value-iteration, Q-learning and SARSA-learning with the support for Softmax action selection and ε-greedy policies [2]. Furthermore, the TB also supports eligibility traces as well as gradient-descent learning of value functions, e.g. by CMACs or radial basis function networks. In order to visualize the learned behavior of an agent, the TB also provides a plotting library for value functions and learned policies.

In our "Laboratory on Artificial Intelligence" students conduct experiments with the TB and the crawling robot by writing simple Java programs. A typical program code that demonstrates the usage of the TB with learning on the hardware robot is depicted in Algorithm 2. At first, a Q-function with tabular approximation is instantiated. Then, the environment (the hardware robot) and the policy to be used are specified. Finally, the Q-learning algorithm (learner) is configured, attached to the agent and a new experiment is started for 1 episode with 300 time steps.

Immediately after the experiment is started, the TB's grid-world editor appears to the user that visualizes the current state of the robot and the rewards observed from the environment in real time, (Figure 5). Furthermore, also a policy window

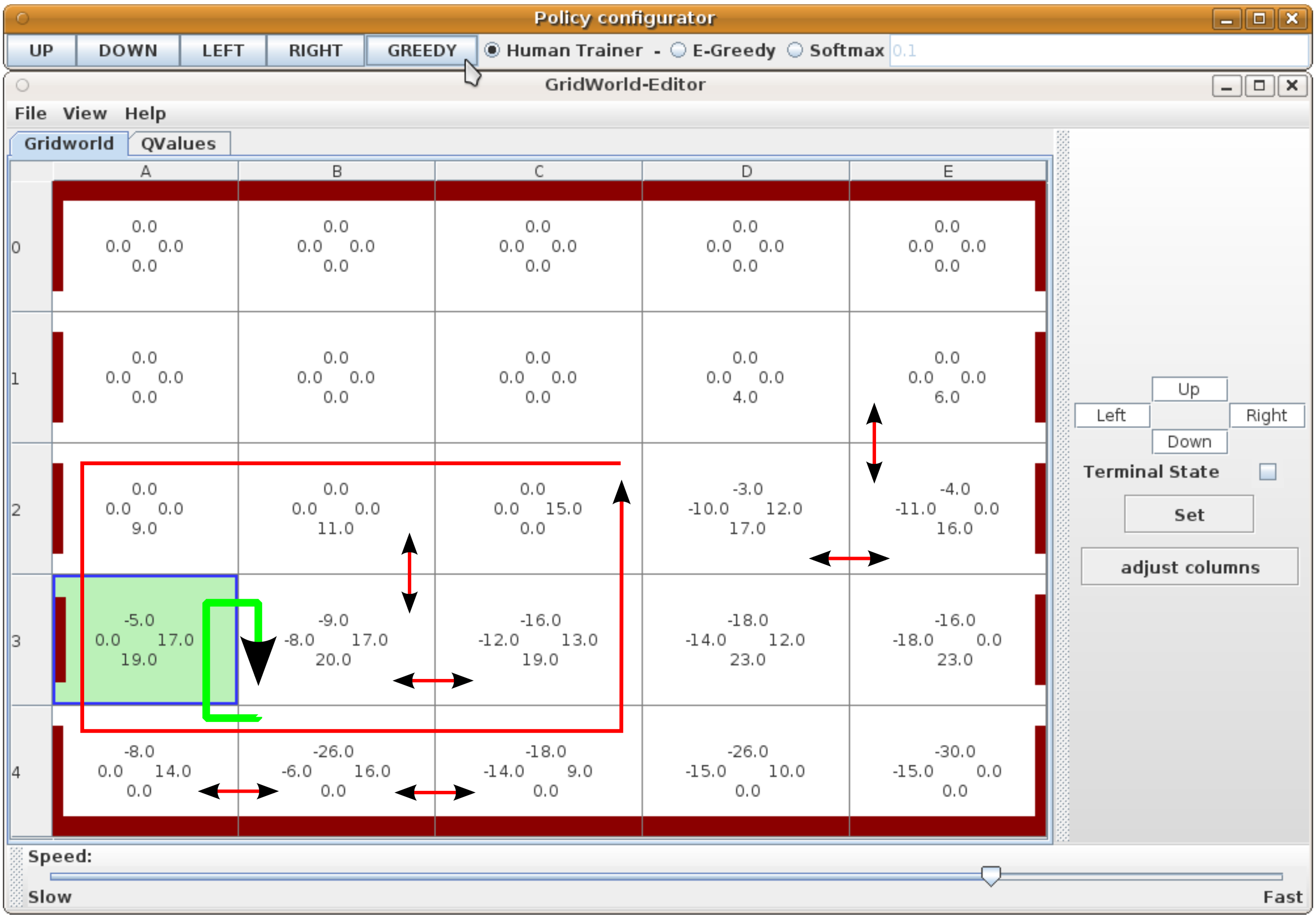


Fig. 5. This figure shows the grid-world editor of the Teachingbox where on top of the window the user is able to configure the policy. Numbers in cells indicate the reward $r(s,a)$ observed from the environment. The cycle A3→B3→B4→A4 indicates the optimal cycle having an average reward of $\bar{r} = \frac{17+20-6-8}{4} = 5.75$ per action. All other marked cycles indicate examples of sub-optimal cycles that have a lower average reward/action compared to the optimal cycle. The cell having the surrounded border (A3) indicates the current state of the robot.

appears (the upper window of Figure 5) in which the user is able to configure the exploration/exploitation policy to be used by the agent. Additionally to the standard policies such as ε-greedy and Softmax, the user can also control the robot "by-hand" when selecting the "Human-Trainer" policy. With this, the robot's actions are controlled by the cursor keys (up, down, left, right) whereby it's also possible to select the "Greedy" action with respect to the currently learned Q-function.

It is easy to adapt the Java code for the use with other environments. For example, if learning should be based in an arbitrary $m * n$ grid-world environment modeled by the user, then only line 2 of Algorithm 2 needs to be adapted to:

GridworldEnvironment env =
new GridworldEnvironment(m,n);

which simply replaces the agent's environment and also demonstrates the flexibility of the Teachingbox which is based on the use of Java Interfaces. This approach standardizes methods of policies, environments and learners with the goal of being interoperable with each other. For example, each environment in the TB implements an `Environment` interface that standardizes important methods such as:

- `double doAction(Action)`
- `State getState()`
- `boolean isTerminalState()`.

where `State` and `Action` are `double` vectors in order to be compatible with each component of the Teachingbox. Another example for the use of interfaces are policies that have to implement a `Policy` interface in order to standardize the methods:

- `Action getAction(State)`
- `Action getBestAction(State)` .

V. Experiments with the Robot and the Teachingbox

In the laboratory tutorial on RL, the first task for the students is to model the rewards of a simulation of the crawling robot by using the TB's grid-world editor. After the modeling of an environment, the policy must be learned by a learning algorithm such as Q-learning, Sarsa or value iteration. During

Algorithm 2 SIMPLE Q-LEARNING JAVA EXPERIMENT

```
1: // initialize new Q-Function with Q(s,a)=0 by default
   TabularQFunction Q = new HashQFunction(0);
2: // establish serial robot link (baudrate, port)
   CrawlerEnvironment env =
       new CrawlerEnvironment(19200, ”/dev/ttyUSB0”);
3: // setup policy configurator
   PolicyConfigurator pi =
       new PolicyConfigurator ( Q,
           CrawlerEnvironment.ACTION_SET);
4: // create agent
   Agent agent = new Agent(pi);
5: // setup experiment with 300 time steps
   Experiment experiment =
       new Experiment(agent, env, 1, 300);
6: // setup Q-function learner
   TabularQLearner learner = new TabularQLearner(Q);
   learner.setAlpha(0.3);
   learner.setGamma(0.9);
7: // attach learner to agent
   agent.addObserver(learner);
8: // start experiment
   experiment.run();
```

this process, students have to conduct several experiments with variations of the learning algorithm parameters α and γ as well as with the policy parameter ε. These experiments lead to the observation that, for example, the discounting-factor $\gamma \in [0, 1)$ has an important influence on the quality of learned policies. For example, if γ is chosen very small, then the agent is more near-sighted and takes not rewards into account received from actions more far in the future, and which often results in learning sub-optimal policies. In comparison, large settings of γ make the agent more far-sighted, but in turn to this, the speed of learning can also be slowly at the beginning of learning since the convergence of the Q-function requires more transition experiences.

While learning the Q-function, the TB can memorize the function on the computer hard-disk, which enables reusing it in other experiments. After the successful learning of the Q-function, students have to evaluate the learned policy from the simulated environment on the real hardware robot. For this, the `GridWorldEnvironment` in the Java code has to be replaced by the `CrawlerEnvironment`. Furthermore, the exploration/exploitation policy for the robot has to be a pure greedy policy ($\varepsilon = 0$), which results that in a given state the action with the highest Q-value is selected. It is important that throughout this experiment learning is disabled in the Java-code, i.e. no `Learner` is attached to the `Experiment`. This results that only the policy of the simulated environment runs on the hardware robot. The idea behind this approach is to enable observing how well the environment has been modeled by the students and how the resulting policy looks like by observing the robots behavior. Therefore, after the selection of an action, the robot transmits the actual state as well as the reward for the most recent selected action to the Teachingbox that visualizes these values in the grid-world editor.

Next, students conduct experiments with the environment of the real hardware robot. In this experiment the `Learner` has to be attached to the `Experiment` again, which enables learning from the robot's environment instead of learning from the simulated environment. Again, each state and reward received from the robot is visualized in the grid-world editor.

Throughout the experiment with the hardware robot, the students task is to vary the learning rate $\alpha \in [0, 1)$ of the Q-learning algorithm that determines how fast the learner adapts the Q-function with respect to the current TD-error δ. The understanding of this parameter is especially important, because the robot interacts in a non-deterministic environment with a noisy reinforcement signal due to the sensor and which also varies due to irregularities of the robot's surface. On the one hand, a large setting of the learning rate causes a fast adaption to environmental changes, for example, when the hardware robot walks from tar to grass. But on the other hand, large settings of α can also be problematic because sensor noise as part of the reinforcement signal may cause the robot to leave a learned ”optimal” cycle. In contrast, when the learning rate is relative small, learning of policies takes more time due to the slow adaption of the (more accurate) Q-function.

The last experiment evolves the understanding for the need of balancing exploration and exploitation, which is also conducted on the real hardware robot and where students vary the policy parameter $\varepsilon \in (0, 1)$ of the ε-greedy method (policy configurator on top of Figure 5). As a result, students will find out that without any exploration, i.e. $\varepsilon = 0$, the agent is very likely to stick in sub-optimal cycles of the state space (due to local minima of the Q-function) that in sum yield to a relative small cumulative reward than compared with the optimal cycle. Such a sub-optimal behavior is observable as the forward walking velocity of the robot that might be significantly slower as in comparison to the optimal cycle. The reason for this behavior on the hardware robot is often due to the fact that the robot hasn't walked through every state transition of the state space and thus actually doesn't know about other (better) cycles. If such behavior occurs throughout the learning process, one can simply solve this misbehavior in the TB by increasing the exploration parameter ε in order that the robot also tries other actions which are not greedy with respect to the Q-function. Furthermore, one may also use the "Human-Trainer" policy and guide the robot into parts of the state space which haven't been explored yet.

Finally, after the experiments with Q-learning, students can perform the same kind of experiment with other learning algorithms, for example, with value iteration or Sarsa and then compare the speed of learning.

VI. Educational Experiences & Conclusions

In order to get an overview whether or not the robot is a good demonstrator for reinforcement learning, we asked our students to participate in a pilot survey. Until the deadline of this paper, the online questionnaire was filled out by 5 of 10 students which represent 50% of the participants of our latest AI course. Because of this low return of responses, the quality of the following results indicates just a rough direction and must be seen as preliminary.

The results of our questions reveal that the students are of the opinion that the robot is a good object of study in general and it seems that the understanding of how reinforcement learning works and how learning method parameters affect the robot's speed (policy quality) got conveyed. Despite of being a good demonstrator, the students also remarked that the robot's hardware is still not comprehensive enough. From our point of view, this answer is not surprising since we cover reinforcement learning in about four lessons (each 1.5 hours), where we're limited in teaching just a small scope of the overall research field, i.e. we just explain discrete state and action spaces and do not consider the continuum. Furthermore, the problem of delayed rewards that exists in real world applications (e.g. the outcome of board games) is not given by the robot example. Anyway, interested students may write their own environments within the Teachingbox, which is possible due to the public availability of the source code on SourceForge. Alternatively, students may also play with standard use-cases such as the mountain-car or inverse-pendulum problem, which are already implemented in the Teachingbox.

The results also indicate that the students sustainably enhanced their skills on differentiating between the two main algorithms we convey: value-iteration and Q-learning. In turn, we couldn't obtain the same good result on the importance of the exploration/exploitation problem. Nevertheless, the students' feedback on both the algorithms and the exploration/exploitation problem was still positive and so we achieved our main goal: To present reinforcement learning in a lively, interesting manner and to support the students learning success. Finally, the students were enthusiastic to see that a behavior which has been learned in a simulation could be transferred to a real robot.

From our teachers' point of view, the robot demonstrator is a versatile instrument which greatly enhanced our lessons on reinforcement learning in order to present behavior learning for robots to the students. With the Teachingbox, we have a reliable software tool that also supports more complex hardware demonstrators that eventually will be constructed in the future. Furthermore, we also provide the robot's construction plans, printed circuit board diagrams and videos on our website[1] and thus enable other interested institutions to rebuild the robot by themselves.

[1] http://ailab.hs-weingarten.de/

Acknowledgment

The authors like to thank Markus Schneider, Richard Cubek, Tobias Fromm, Tobias Bystricky and Andy Stumpe from University of Applied Sciences Ravensburg-Weingarten which were involved by the co-development of the Teachingbox. The authors also thank Günther Palm, Mohamed Oubatti and Friedhelm Schwenker from Ulm University for the scientific discussions on the crawling robot and also acknowledge Philipp Ertle from University of Applied Sciences Ravensburg-Weingarten for proof reading and discussing a first draft of the paper.

References

[1] E. L. Thorndike, *Animal Intelligence*. The Macmillan company, New York, 1911.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[3] T. W. Neller, C. G. M. Presser, I. Russell, and Z. Markov, "Pedagogical possibilities for the dice game pig," *Journal of Computing Sciences in Colleges*, vol. 21, no. 6, pp. 149–161, 2006.

[4] H. Kimura, K. Miyazaki, and S. Kobayashi, "Reinforcement learning in POMDPs with function approximation," in *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 152–160.

[5] M. Tokic, "Entwicklung eines lernenden laufroboters," Diploma thesis, University of Applied Sciences Ravensburg-Weingarten, Weingarten, Germany, 2006.

[6] J. Kay, "Robots as recruitment tools in computer science: The new frontier or simply bait and switch?" AAAI Press, Tech. Rep., 2010.

[7] W. Ertel, M. Schneider, R. Cubek, and M. Tokic, "The Teaching-Box: A universal robot learning framework," in *Proceedings of the 14th International Conference on Advanced Robotics (ICAR 2009)*, 2009, pp. 1–6, http://www.teachingbox.org.

[8] M. Tokic, W. Ertel, and J. Fessler, "The crawler, a class room demonstrator for reinforcement learning," in *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference (FLAIRS'09)*, H. C. Lane and H. W. Guesgen, Eds. Sanibel Island, Florida, USA: AAAI Press, 2009, pp. 160–165.

[9] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge, England, 1989.

[10] S. B. Thrun, "Efficient exploration in reinforcement learning," Carnegie Mellon University, Tech. Rep., 1992.

Experiences using autonomous model airplanes for embedded control education and for bachelor and master theses projects

T. Kittenberger, L. Brodl, N. Vavra
Department of Embedded Systems
University of Applied Sciences Technikum Wien
Vienna, Austria
t.kittenberger@technikum-wien.at

***Abstract*—We present experiences with developing an autonomous airplane from scratch including all the avionics and why we focus on an UAV as an application for educating skills in electronics or control theory. The reasons for the selection of the type of aircraft are discussed in detail. Our avionic system is presented consisting of an inertial navigation system coupled to a GPS receiver, a half duplex 433 MHz link and an ultrasound landing guidance system. All electronics are developed in-house. For simulation and control design the Matlab/Simulink environment is used. The 3D simulation output is done by a network link to the open source flight simulator FlightGear and by a graphical 3D Simulink output window.**

***Index Terms*—Control engineeering education, UAV, model airplane, avionics, inertial navigation system, flight simulation, best practice.**

I. Introduction

The first fully integrated MEMS gyroscope, introduced by Analog Devices in October 2002 [1], in conjunction with already available MEMS acceleration sensors and electronic compasses, enabled for the first time the development of really small and lightweight Attitude and Heading Reference Systems (AHRS), that used to be heavy and expensive before. An AHRS with a GPS-receiver, assistive radio control and other sensors, builds the heart of the avionic system of a modern unmanned aerial vehicle (UAV), an aircraft which can fly radio controlled but in most cases is also intended to fly autonomously. Until the 90's (large) UAV's were mainly of interest for military use, but with the new, lightweight MEMS AHRS's it was possible to shrink the size of autonomous UAV's to that of small model aircrafts and beyond, the smallest weighing only a few grams. The advances in clean and silent electric model aircrafts, like high energy density lithium polymer batteries and high efficiency brushless DC motors did the rest. Many universities, like ETH Zürich [2] or MIT [3] have established UAV groups, dedicated to the research on this new class of aircraft, also called MAV (Micro Aerial Vehicle). Airplanes, helicopters and quadrocopters were built, analysed and programmed to fly autonomously and a lot of scientific papers, master theses and doctoral theses were and are written about UAV's.

But UAV's are not only of scientific interest, they increasingly become a possible application for the education of electronics, radio data transmission, control engineering, object recognition or artificial intelligence in a similar way as we can see it with the well known soccer or sumo robots. Like with these, UAV competition series have started all around the world and they are experiencing impressive growth rates. Examples are the International Micro Air Vehicle Conference and Flight Competition [4] held in Germany or the International Aerial Robotics Competition [5] in the USA.

Although the technical skills could be practised on simpler applications, UAV's have the advantage to exercise a strong fascination on students and applicants, their professors and the public. They rise the motivation and with the right marketing UAV's and other mobile robots have the potential to attract more students for technical universities in general and especially for our University of Applied Sciences Technikum Wien.

II. The UAV Team at the University of Applied Sciences Technikum Wien

The Department of Embedded Systems at our University established it's UAV team in September 2007 with five primary objectives:

- to get challenging, attractive **bachelor and master projects** where most skills of electronic education could be practised
- to build impressive, functional models to **promote our university** at public technical shows and contests
- to **attract more new students** for the university and for our master programme
- to develop interesting **educational models and course materials** for our courses like the two term embedded control course
- to **provide key components** like the inertial navigation system or the telemetry system for other projects and degree programmes

This project is founded by the City of Vienna, department MA27, under grant "Sponsored Professorship Embedded Control and Navigation".

The team currently consists of students from four different degree programmes working on their bachelor or master theses in their last year of study and the team leader and founder Thomas Kittenberger. The team members change each year, new members are selected and introduced in May and June and work on the project and their theses from September to next June. Some students were team members with their bachelor projects as well as with their master projects.

The team started with three master theses on the development of a MEMS based attitude sensor. In study year 07/08 the electronics, the simulation and calibration procedures and a computer controllable rate table for calibration were developed. In study year 08/09 the UAV-Team grew to two master and three bachelor students. They designed a new, flat inertial navigation board, improved the INS calibration so that also a short term position integration is possible and added a GPS receiver for long term positioning. Also the hardware for several other sensors like ultrasound ranging, barometric altitude measurement, battery monitoring or propeller speed measurement was developed. For telemetry a 433 MHz COTS transceiver module was used at first, which was replaced later by a transceiver module developed at our department. In the last year 09/10 the team consisted already of three master and six bachelor students and produced about 500 pages paperwork (which had to be proofread in only a few weeks). The main aim was to install the new avionics hardware on an appropriate model aircraft and realise autonomous flight including take off and landing.

III. Selecting an Appropriate Aircraft

An important and far-reaching decision is the choice of the type of aircraft. We wanted to have a simple and robust construction that was capable to fly indoors and outdoors with a gross weight in the range of 300 to 500 grams to be capable to carry all the electronics. The price shouldn't be too high to avoid a golden sample nobody dares to fly and the propulsion should be done with LiPo batteries and brushless motors.

Helicopters with that weight need already larger size and high speed rotors. The kinetic energy in the rotor blades presents a serious safety issue. An accident can seriously harm students and a broken helicopter is quite expensive to repair. This type of aircraft is only appropriate for experienced teams but not for beginners and in classroom use you have each year new beginners.

Quadrocopters have typically four smaller and safer rotors and the overall complexity is simpler compared to that of a helicopter. Therefore they are better suited for use in electronic education and they are also quite common in the UAV scene. The attitude control is either done by regulating the speed of each motor, what leads to slower response times, or by using variable pitch propellers similar to a helicopter, what makes them more complex and expensive. Because they use simple, less dangerous propellers beginners can fly them indoors as well as outdoors. If the propellers are mounted on a propsaver (rubber band) a hard landing can results in no damage at all or just a simple propeller has to be exchanged. A real crash would destroy the construction, brake the servos and possibly bend the motor shafts. Looking at the price of a full blown variable pitch quadrocopter compared to that of a simple fixed wing aircraft that seemed to risky and expensive to us.

Fixed wing propeller airplanes are the oldest and simplest aircrafts heavier than air. With lightweight LiPo batteries and brushless motors modern 3D acrobatic airplanes are also able to hover vertical like a helicopter, even in small labs, although they use a single small size propeller like a quadrocopter. Instead of four propulsion units consisting of motor, propeller and electronic speed controller, just one propulsion unit is necessary. The fuselage and the wings can be made of cheap an lightweight synthetic foams like extruded polystyrene (EPS) or the viscoplastic expanded polypropylene (EPP), which forgives smaller impacts and is easy to glue if it comes to severe crashes. And if it comes to outdoor level flight an airplane just looks nicer than any other flightgear.

Looking at the above aircraft type evaluation it is clear that from our point of view a classic airplane made of EPP foam would be the best choice. We selected the MS Composit Unique, consisting of EPP wings and fuselage, EPS rudder and elevator and a steel wire landing gear. The wingspan is 94 cm and according to the manufacturer the flight weight should be 320 to 450 grams. The airplane is now powered by an AXI 2217/16 with an 10x4.7" APC Slow Fly propeller and a 3s LiPo battery with 1500 mAh. The gross weight including all our electronics is 650 grams, much more than it should be. At hover flight the motor draws about 13 A. In Figure 1 you see the current setup of our airplane including motor, propeller and the three avionic boards. The battery and the right wing airflow and altitude sensor are missing.

Figure 1. UAV of the Department of Embedded Systems.

After one year of work with this airplane we have made a lot of experiences and we hope it is interesting for you to look back with us at our thoughts from the start of this year and forward to our plans for the next year.

Our decision to select a fixed wing aircraft for level and hover flight turned out to be right. But a fixed wing aircraft hasn't always to be a classic style fixed wing aircraft. The level flight worked perfectly but with hover flight we had some problems. To stabilize the airplane on the roll axis in hover flight the ailerons counteract against the moment induced by the propeller. Depending on the mostly large deflection angle of the ailerons the airstream on rudder and elevator gets heavily disturbed. This induced a strong coupling from the roll control loop to the yaw and pitch control loops. We think that the best solution to this problem would be the change to another aircraft type not considered by us so far, a delta wing aircraft with a shape like a fighter jet. After this idea we found a lot of slow fly delta aircrafts when looking specialised shops. They combine the two ailerons and the elevator to two elevons at the back of the plane. The rudder stays the same. Synchronous control angles on the elevons act as elevator, antisynchronous angles act as aileron. The airstream from the propeller in hover flight mode is always the same on the elevons and the coupling between the control loops is reduced. Other advantages of delta wing airplanes are that it is easier to get large wing surfaces and low wing loadings, that they are not so sensitive to aerodynamic stall like classical wings and that the construction can be very simple when using flat EPS sheets, stiffened by carbon rods. If the rudder is extended to both sides of the delta wing you get a tail sitter aircraft which can also start and land vertically.

Another important point is the weight of the airplane. In summer 2009 we thought that 300 to 500 grams would be a realistic range for our plans. With the avionics boards, motor, propeller, speed controller and battery we ended near 600 grams. On a longer lasting hover test it happened that we lost thrust, we increased the power and lost more thrust. After stopping the motor we realised that it had got really hot. An inspection of the motor revealed that the bearings were okay but the magnets had lost a lot of their power. We found that the maximum temperature for the used neodymium magnets is 80 degree Celsius. Above this temperature they begin to loose their magnetisation which demands more current for the same torque which increases the temperature and so on. The primary cause was the missing airstream in hover mode that cools the motor otherwise in level flight mode. So we added a cooler fan normally used for motors that are used in helicopters and we increased the size of the motor because it seemed to be on his power limit which resulted in a gross weight of near 650 grams.

The point here is not that 650 grams, the point is the cycles you get caught in, always increasing this and that by trying to reach the demanded performance in thrust, payload or battery runtime. In a typical well-balanced airplane design the parts motor, airframe, battery and payload in form of the avionic system have rather fixed mass ratios. Starting with a half weight avionic system could end in a half weight airplane.

Lightweight and small aircrafts are important for a lot of reasons if they are intended for educational purposes. They are cheaper because motor, speed regulator and battery can be smaller. You can build more aircrafts for the same budget and it hurts less if you lose same by crashes. They are easier to transport and store, in the lab as well as on the way to an airfield or a competition. And most important, they are less dangerous. We had to learn that the hard way. Our current 650 g thrust 140 W propeller hit a student without safety gloves on the hand while he was working on control loop adjustments in hover mode. The resulting wound had to be sewed in hospital. On another occasion a thin propeller was unintentionally driven above his specified maximum rotary speed, the blades started to resonate and one blade dismounted and was shot against a wall. Nobody was injured. So take the advice, always wear safety gloves, safety goggles and if possible a coat when experimenting near running propellers. Or even better, try to avoid being in the rotary plane of the propeller.

Another aspect to regard is an adequate place for test flights. Our labs were okay for hover mode flights, but for level flight even large labs or gyms turned out to be too small because of the heavy weight and our missing flight experience. With 650g, a wing span of 94 cm and a wing area of 14.1 dm^2 the resulting wing load is 46 g/dm^2, what requires velocities in the range of 10 to 12 m/s for level flight. We did most of our test flights at a recreation area near the university. Not the best place considering all the other people enjoying their leisure time there and apart from potential legal issues. If possible a dedicated airfield should be the place of choice. If not, a slow flying 200g airplane would be much better than a fast flying and 650g heavy one. In winter or with windy weather a low wing load would also simplify or even enable indoor flight in a large hall or gym.

The last point we want to discuss is the number of aircrafts used. At the beginning of the last study year we wanted to build two or three aircrafts and even more avionic boards to be able to work in parallel and to have spare aircrafts if one crashes severely. We started with the development of boards and software, built our first airplane and as time went by we got more and more behind our schedule. We accomplished to assemble additional boards but there were no time until the end of this year to build a second airplane. We had good luck that no severe crash occurred but near final project presentation we felt quite unwell with that risk in mind. In the next year we will build our new airplanes right at the beginning and select a simple construction so that a few new airplanes can be built in a day. It would also be helpful to choose a modular design where you can change the avionic board(s) quickly from one airframe to another.

Summing all this up we plan following improvements for the next project year:

- switch from a classic wing design to a delta wing design to avoid problems with the airstream on rudder and elevator
- build the airframe in house using EPS-sheets and carbon rods, use a very simple modular design which allows to exchange airframes and avionic boards easily
- build several airplanes and avionic boards at the beginning, there will be no time for that at the end
- tune the properties of the aircraft: similar wingspan, slightly larger wing area, much less weight, the aim is to reduce the wing load from 46 g/dm^2 to 10-15 g/dm^2 and to reduce the minimum level flight speed
- develop a new avionic board with drastically reduced weight, choose a centralised single processor design instead of the current modular multi processor design

IV. The Avionic System

UAV's require much higher cognition of their environment than other robotic vehicles. One way that can be accomplished is with an array of wall mounted cameras and an image processing system to compute the position and orientation of the airplane. The MIT Aerospace Controls Laboratory [3] for instance uses that approach. It has the drawback that it works only indoors in the laboratory environment but the advantage that the airplanes can be built much more lightweight.

Another way, we chose for our project, is to measure all required parameters on board of the airplane. A whole bunch of sensors is needed for that task. Figure 2 shows the system concept.

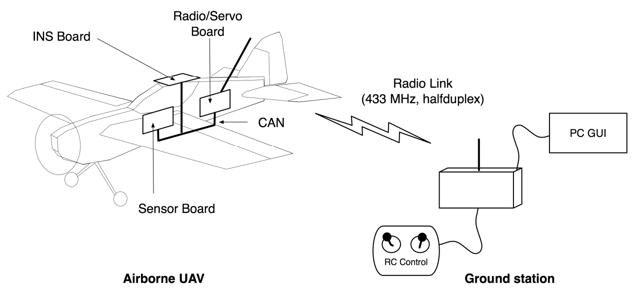


Figure 2. System concept with ground station and remote control unit.

Our inertial navigation system with GPS-support, that was built from scratch at the department over the last three years, is capable to detect the orientation, position and heading of the airplane. A triangulation method based on ultrasonic ranging allows autonomous landing on a runway equipped with ultrasound transponders. To measure the airspeed we developed a hot wire anemometer that we thought would be better suited for low speeds than a pitot tube. When we built a calibration wind tunnel for the airspeed sensors, we discovered that the performance of differential pressure sensors at low airspeeds isn't so bad after all. A pressure sensor is used to provide barometric height computation. Telemetric data and control signals are exchanged over a 433 MHz half duplex radio channel. RPM gauge for the engine speed and battery monitoring provide additional information. The complete independence from a ground based camera system gives the airplane more mobility but increases the weight and requires a very high accuracy of the inertial navigation system.

In the current setup the inertial navigation system, the transceiver and the sensor unit are on individual circuit boards, each equipped with a 40 MHz, 16 bit Infineon XC164 micro-controller, and connected by a CAN fieldbus. The distributed system allows individual development of each subsystem, well suited to bachelor and master theses projects. Integrating all tasks onto one controller could save board space and weight by eliminating duplicate hardware such as the processors, power supplies and other peripherals. A single processor with a higher clock rate could easily handle the necessary computation without the need of a network for data exchange. That however might result in a more complex software structure to enable several team members to develop separately for the same hardware. Due to the amount of calculations and the necessary accuracy, picking a processor with integrated hardware floating point unit would be a good choice for the next hardware redesign.

An important aspect of the project is that all hardware was developed by students at our university. Although prefabricated modules are commercially available, designing them in-house provides a wide range of advantages that outweigh longer development time and involved problems. It opens the opportunity to exactly match the requirements of the own project, especially weight considerations, allows to use brand-new sensors at lower prices than comparable modules and gives students insight into various fields of expertise. Nevertheless the complexity shouldn't be neglected. Reverse engineering a transceiver unit took more than one attempt to obtain a module that was comparable with the original one. Without a fair knowledge of high frequency engineering we spent many hours trying to identify the flaws in the design. So there are occasions where complete modules might be the better choice. But the more difficult it is to get a board up and running, the greater the joy is when it finally works.

V. Simulation and Control Design

To accurately model, simulate and control an aircraft system is a key factor in all aerial robotic projects. Since small sized UAV's follow the same physical laws as full-scale planes, the fundamental theory of flight mechanics as presented in aviation textbooks [6] is also applicable but it makes sense to simplify these models. Especially choosing a simple airframe configuration and restricting the flight envelope to simple manoeuvres allows various simplifications. The overall mathematical description of an UAV can be divided into an aerodynamic model, a propulsion model and a mechanical model including center of gravity and mass distribution. While the aerodynamic parts include static, dynamic and control characteristics the propulsion part models the static and dynamic thrust forces and moments caused by the engine and the propeller. There are several different ways to perform system identification of an UAV to obtain coefficients for the mathematical model.

While parameter identification from real flight test data requires advanced filter methods to estimate coefficients another approach has been established in the field of UAV development called the Digital Datcom method [10]. Digital Datcom is a software tool released for public, allowing an accurate prediction of almost all aerodynamic coefficients by defined vehicle geometry and an expected flight envelope. In our case, Digital Datcom has shown satisfying results for our small sized fix wing UAV. Since Datcom parameter estimation is independent from a flyable aircraft, it can be used concurrently to hardware and software development allowing parts of the team to simulate the dynamics and design control laws without having a flyable prototype available. A similar approach to Datcom is available for estimating propulsion, in particular propeller coefficients (JavaProp).

Having all parameters measured, estimated or assumed the model can be simulated within a mathematics software tool or a stand-alone executable program. In our case we choose the Matlab environment in conjunction with Simulink and the

Aerospace Toolbox to create a 6-degree-of-freedom nonlinear simulation, running in almost real time. As an alternative to the commercial Toolbox the AeroSim Blockset is available for free, given that it is used for education only [7]. This Blockset contains a variety of predefined aeronautical functions like coordinate transformations or calculations of forces and moments.

Having an accurate model of the plant allows to design the control laws. It is important to define levels of autonomy at the beginning and start with the low-level controller (e.g. control of angular rates or angles) followed by mid- and high-level controllers or autopilot state machines. It is obviously that flight control tasks have a wide bandwidth leading from simple stabilization and damping up to fully autonomous functionality. Since the aerodynamic model of the vehicle is a MIMO system utilizing the state space representation makes sense. But for educational use it is also sufficient to consider the aerodynamic model to be decoupled from longitudinal and latitudinal movements and flown at a constant pace. Assuming this simplifications leads to a SISO plant dynamic where linear control theory is applicable [8][11] (e.g. the elevator to pitch angle transfer function can be simplified as a 2nd order system). In our case this approach was chosen for a group of students who designed an angular controller for hovering flight in an embedded control course.

Before our first autonomous flight attempts, we thought of a construction like a mobile (see Figure 3, the airplane is vertical at hover tests) to hang the airplane up onto strings to roughly identify the aerodynamic properties for hover flight mode. It turned out that the mount was not appropriate for that matter because it introduced additional momentum into the system and was much to unstable.

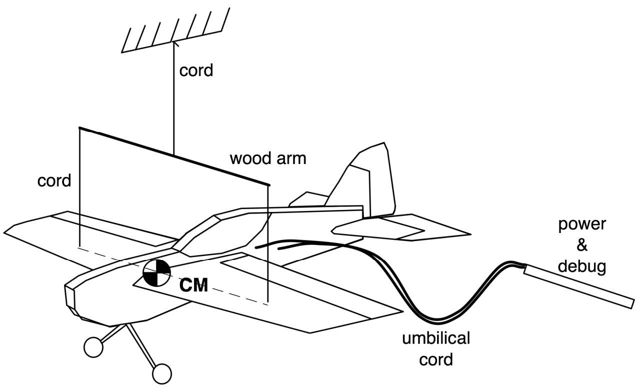


Figure 3. Mounting construction for a system identification attempt.

Another approach to identify simplified dynamics for hovering, that hasn't been tested, is to mount the aircraft with the centre of gravity on a ball head, a so called gimbal mount (see Figure 4). That kind of mounting represents a limited 3-degree of freedom platform which can be used to measure impulse or step responses initiated by the control surfaces.

An even better approach as already mentioned is system identification by data obtained from real, manually controlled flight experiments. That way information can be collected under almost the same conditions like in an autonomous flight. Unfortunately only a small amount of the recorded measurements are suitable for system identification purposes. In our case this method is still in progress and we are looking forward to get our first system models ready for simulation at the beginning of the next study year.

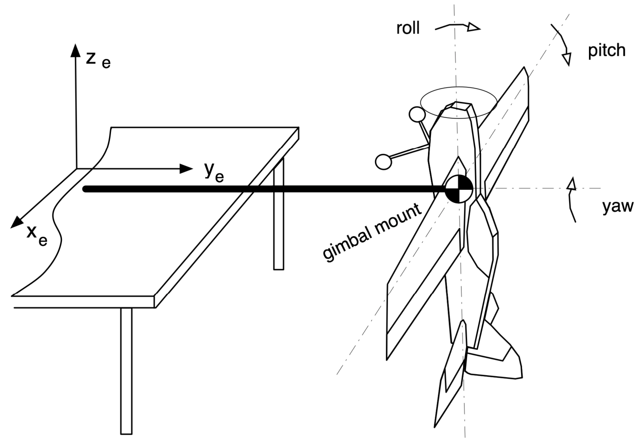


Figure 4. Gimbal Mount construction for a system identification attempt.

For the system identification based on the recorded flight data we use the Matlab System Identification Toolbox. Notice that in our case all flight controllers, sensor systems and actuator systems are designed, modeled, checked and evaluated within the Matlab/Simulink simulation environment.

Figure 5. FlightGear Screenshot [7].

There are many ways to visualize simulation results. While scopes can be used easily and fast to display scalar time responses, Matlab/Simulink is also able to visualize simulation results in an Open Source flight simulator called FlightGear[9]. It can be used as a stand alone flight simulator or it is remote controlled by network interface with the simulation results from the Matlab environment. See Figure 5 for a screenshot.

Within our project a simpler 3D animation window was designed, displaying our current vehicle's geometry as shown in Figure 6.

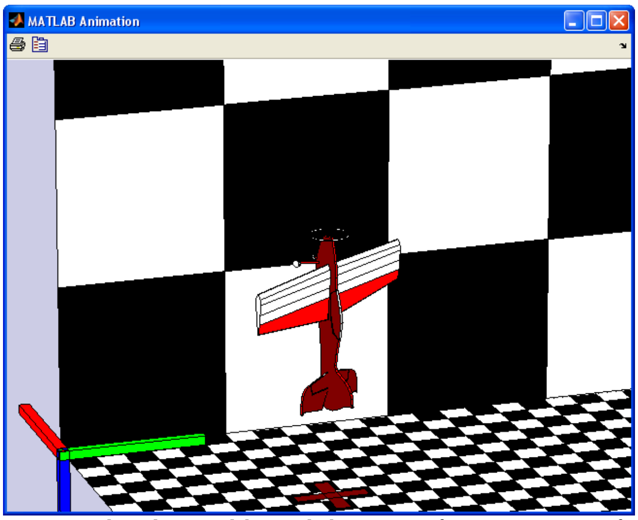


Figure 6. Adapted 3D Matlab/Simulink animation for our current aircraft.

VI. CONCLUSION

For the UAV team members the UAV-project is a great, challenging opportunity to demonstrate most of their acquired skills within their bachelor and master theses projects. Many of them work for the first time in a one year lasting project. From the point of view of a professor it is very satisfying to do this introduction to project life with such a beautiful project setup including so different interacting disciplines like sensorics, electronics, telemetry, control engineering, software design, state of the art simulation and most of all real world, high agility flying machines.

When teaching some of the project topics in classroom courses, care must be taken that the presented problems are not too complex, that the initial training effort is low and that the course materials are well prepared. With a careful introduction to the quite complex theories behind UAV flight there is a good chance to inspire the students to dig deeper in this topic. Using functional models to perceive the concepts will help a lot.

Looking back at our last year airplane we have to improve a lot. It is much to heavy, too fast and therefore too dangerous. Also the wing profile is not optimal and will be changed to a delta wing configuration. We need also more and simpler airplanes from the beginning on. The same is to say for the avionic board. We will change from a modular concept, although it had some advantages, to a hopefully less weight, single board concept with only one MCU that in turn is more powerful. With our Matlab/Simulink/FlightGear simulation environment we are really happy. The tools are mighty and the documentation is comprehensive. We look forward to the next year of fascinating work on UAV's.

REFERENCES

[1] "1st-ever MEMS integrated gyroscope unveiled," InTech Magazine, International Society of Automation, Durham, NC, USA, 2 October 2002, online available at www.isa.org.

[2] UAV group at ETH Zürich, www.uav.ethz.ch.

[3] MIT Aerospace Controls Laboratory, aerobatics.mit.edu and vertol.mit.edu.

[4] IMAV 2010, International Micro Air Vehicle Conference and Flight Competition, 06-09 July 2010, Braunschweig, Germany, www.imav2010.org.

[5] International Aerial Robotics Competition, iarc.angel-strike.com.

[6] M.V. Cook. Flight Dynamics Principles. Elsevier Ltd, 2nd edition, 2007.

[7] Aerosim Blockset User Guide, V1.2, Unmanned Dynamics, LLC No.8 Fourth St. Hood River, OR 97031 (503) 329-3126 http://www.u-dynamics.com info@u-dynamics.com.

[8] D. Lara K.C. Wong, J.A. Guerrero and R. Lozano. Attitude Stabilization in Hover Flight of a Mini Tail-Sitter UAV with Variable Pitch Propeller. In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007.

[9] "FlightGear Flight Simulator", June 2010, http://www.flightgear.org/.

[10] Williams, J. E. and Vukelich, S. R., "The USAF Stability and Control DATCOM", tech. rep., McDonnel Douglas Astronautics Company, St. Louis, MO, 1979, AFFDL-TR-79-3032.

[11] Williams E. Green, Paul Y. Oh. Autonomous Hovering of a Fixed-Wing Micro Air Vehicle. In IEEE/International Conference on Robotics and Automation, Orlando, Florida, May 2006.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

3D Tower Crane as a mechatronic tool for education

Mariusz Pauluk* and Dariusz Marchewka*
*AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Automatics
Mickiewicza 30 av. 30-059 Kraków, Poland
e-mail:mp@agh.edu.pl, dmar@agh.edu.pl

***Abstract*—The paper presents a laboratory model of the 3D tower crane. The arm of the crane is approx 1.2m long and the model is approx. 1.5m heigh. The crane is equipped with three DC motors that control: the rotary movement of the tower, the movement of the trolley and up and down movement of the load. The model is equipped with an unique unit for measuring an angle position of the load. The paper contains description of some mechanical solutions as well as results of chosen experiments. Some aspects of appliance in education are discussed.**

Keywords-component; 3D crane, mechatronic model, real-time control,

I. INTRODUCTION

The laboratory model, presented in the paper was designed and constructed on the basis of the experience, gained by the authors during designing and building up a mechatronic models of a gantry crane and a 3DOF manipulator[5][6][7]. The aim was not to build a copy of any existing industrial equipment, although received a laboratory model reflects many of the phenomena occurring during a transport of a suspended load [2][3].

Cranes type of the tower, in opposite to the gantry cranes work mainly outdoor, where cargo is moved at high altitudes, or when a terrain is rough, and even moving. The most often it is used in: building construction sites, ports and vessels.

A work of cranes is seriously dependent on weather conditions, especially on a wind speed. Extremely difficult place to work are vessels, where in addition to a strong wind come also disturbances caused by the rocking board of a ship [4].

Presented the laboratory model allows to familiarize with: a complexity of controlling such object, designing, developing, implementation and testing control algorithms in real time. The RT-DAC board [9] allows to execute algorithms in the MATLAB&Simulink environment in real time. The results of the experiments may be observed directly on MATLAB scopes. The system also allows to test real time control parameters by the possibility of setting a sampling time, and the frequency, the control is generated

II. GENERAL OVERVIEW

Figure 1 presents a general view of the model. The crane may hoist or lower a suspended payload and also to move the payload along the rail and around the basis. The crane is controlled in real time in the MATLAB & Simulnk environment. A control PC computer is equipped with an analog-didgital borad (RT-DAC USB [9]) that mediates in an exchange of data between the controlled object and a controller running on the PC machine. Digital outputs of RT-DAC are connected to the crane power interface, where the calculated by the control algorithm value of control is converted into a PWM type voltage signal and then distributed to an appropriate DC motor.

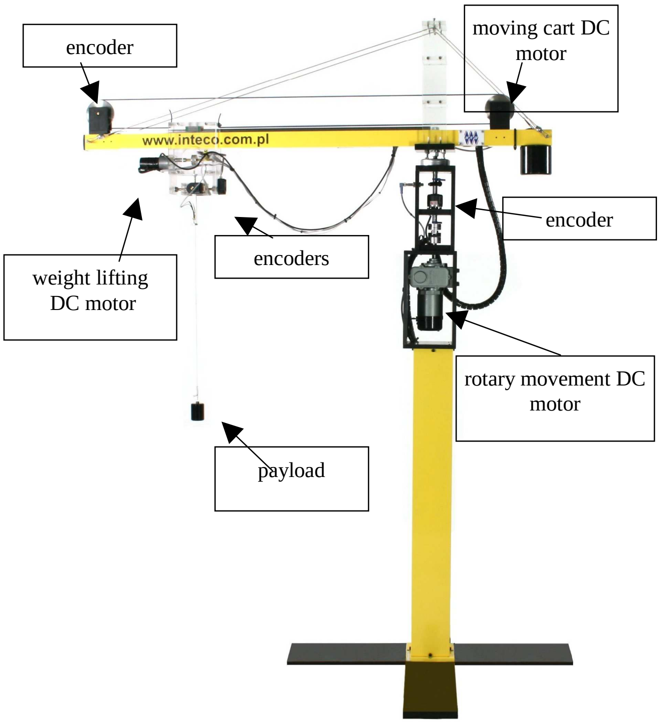


Figure 1 Tower crane – general view

An electric drive consists of three DC gear motors. The motors: weight lifting and moving the cart are mounted on shafts equipped with other gearboxes changing rotary to plane motion. There are also two encoders (measuring rotary position) placed on the shafts. Other two encoders are placed in the trolley mechanism for measuring in two planes a deviation of the rope from the vertical position. The third gear motor is placed directly inside the crane body. The shaft, transfering torque from this motor is equipped with the sixth encoder that measures rotary position of the crane arm with respect to the basis. The crane is equipped with three limit switches to prevent the construction from damages caused by fault control.

The presented crane is not a copy of any existing industrial object. It is a very good tool: for research purposes, for examination of phenomena that occur during movement of a suspended payload and for designing control algorithms assuring a safe transport. Safe means resistant to disturbances like violent gusts of wind or sudden appearance of an obstacle.

A. Trolley drive

The RH158 gear motor (Figure 2) has been chosen as a trolley drive. The motor, equipped with a gearbox 76.84:1, gives 50 Ncm maximal torque [10]. The rotary motion is converted into the flat movement by another gear that pulls a steel link connected to the trolley.

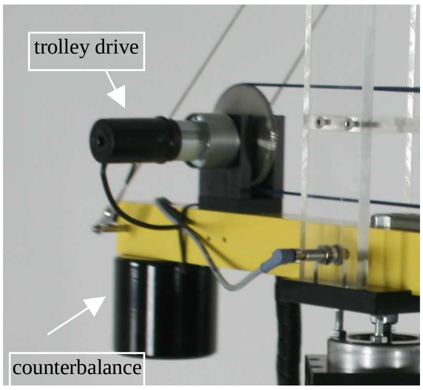


Figure 2. Tower crane – trolley drive

B. Trolley

A mechanism for measuring a deviation of the payload is mounted in the trolley (Figure 3). It consists of two shafts connected similar like in the Cardan coupling. Both shafts have mounted encoders for measuring angles of deviation in two planes.

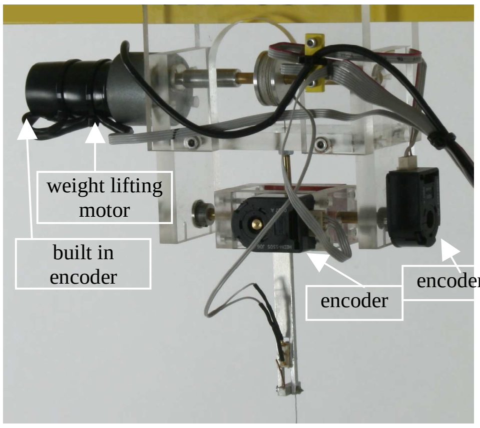


Figure 3. Trolley

In the upper part of the trolley, above the mechanism, there is a weight lifting motor (Type RH158-2S). Rotation of this motor (length of the rope) is measured by the built in two-phase Hall-effect 90° encoder. The rope goes exactly through the center of the mechanism. A deviation of the payload forces a deviation of the mechanism, that is noticed by the encoders.

C. Tower

A construction of the crane tower presents Figure 4. Parvalux motor: PM10MIW has been used as a drive of the crane rotary movement.

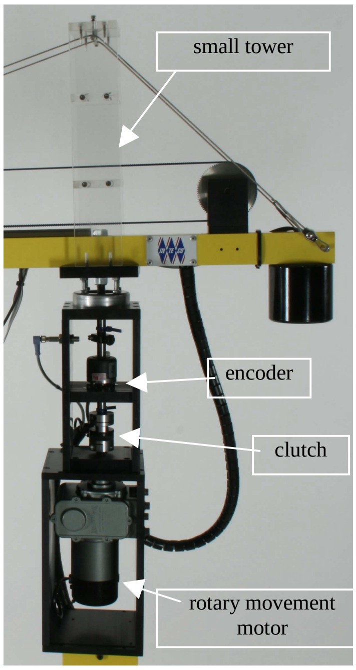


Figure 4. Tower

Its maximal torque is equal to approx. 11Nm. The motor is mounted inside the construction of the tower. The torque is transmitted by the shaft (diameter 12mm) to the arm. The motor is connected to the shaft by the clutch (Figure 4). There is also an encoder mounted on the shaft for measuring an angle position of the arm. The shaft is connected with the arm rigidly. The small tower (Figure 4) assures a rigidity of the arm.

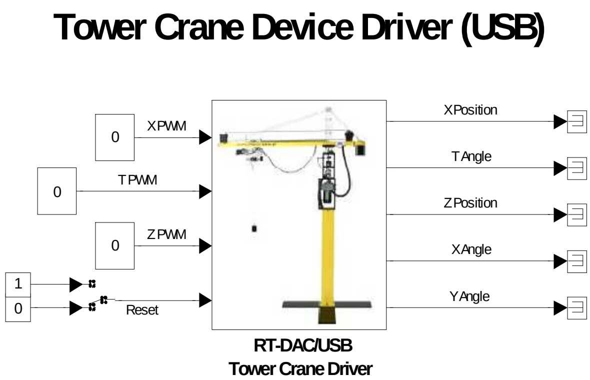


Figure 5. MATLAB&Simulink environment

III. EXPERIMENTS

The crane is controlled in the Matlab&Simulink environment. It was prepared a driver for experimental purposes (Figure 5). The driver has three inputs: XPWM – for control the trolley motor, TPWM – for control a crane rotary movement, ZPWM – for control the weight lifting motor. The control values may vary from 0 to 1. The value 0 refers to no control, value 1 means full control. The control is PWM type. A value between 0 and 1 refers to the fill factor of the control square wave. The switch “Reset” sets the encoder counters to value 0. It is used for calibration purposes.

There are five outputs available for a user: X Position – position of the trolley in reference to the length of the arm, T Angle – angle position of the arm in reference to the crane basis, Z Position – length of the rope with a suspended payload, X Angle – angle deviation of the payload in the arm plane, Y Angle - angle deviation of the payload in the plane, directed perpendicularly to the arm.

A. *Payload movement*

A simple connection of the X Angle and Y Angle signals to the Simulink tool: Scope, makes able to observe in real time movement of the payload in the X Angle vs. Y Angle plane. Figure 6 presents results of an exemplary experiment with an oscillating payload. Small crosses on the figure denote measured points.

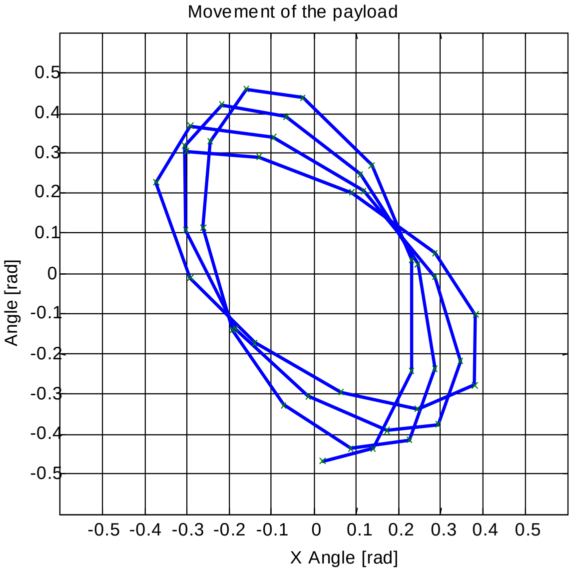


Figure 6. Movement of the payload

B. *Control experiment*

The driver of the crane looks like a typical Simulink model therefore a construction of the controller is intuitive for person familiarized with the MATLAB&Simulink environment (Figure 7.). In the presented experiment, only the x axis is fed by the control signal, values of the GainT and GainZ elements are set to zero. A controller is a type of two position block with a hysteresis. A width of the hysteresis defines a precision of keeping the object position near a desired value. In this case it is a trolley position along the arm. When the trolley goes beyond a specified position, the controller changes the control to the opposite value. When the trolley changes its position, the payload is moving freely. Some signals are connected to the Scope. Figure 8 presents the chosen signals that were observing in real time during the experiment. As the width of the hysteresis is set to 0.3 (the lower value is equal 0.1 and the upper 0.4,) the trolley changes its position among these values. When the trolley position overcrosses the value 0.4m, the

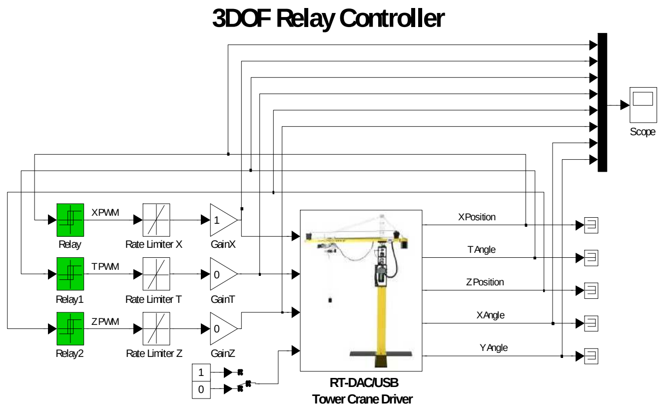


Fig. 7. Relay controller

control is changed from the value 0.5 to –0.5, and when the trolley position achieves the value lower than 0.1m, the control is changed from the value -0.5 to 0.5. During these changes, oscillations of the payload are increasing or decreasing randomly. There is no control algorithm dumping the oscillation activated.

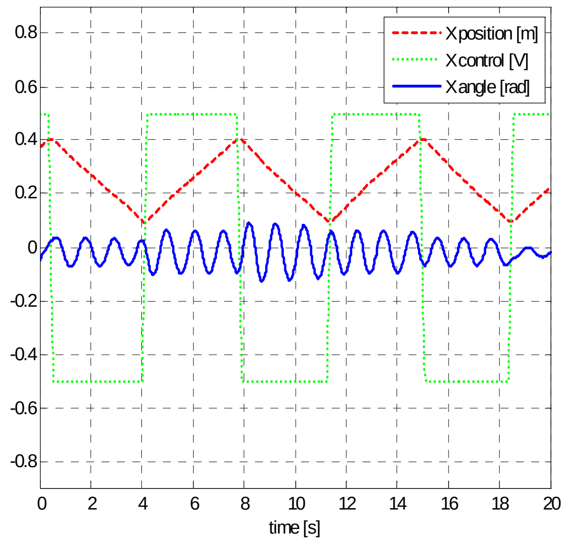


Figure 8. Results of the experiment

IV. EDUCATIONAL ASPECTS

As the system is equipped only with the position sensors (digital encoders), students during a construction of the control algorithms meet a problem with lack of a velocity signal. The simple solution in a form of the element differentiating the position signal, early shows its disadvantages, especially when the system moves slowly. To achieve a better performance, students must think about other solutions, like e.g. observers.

The system also reveals the phenomenon of elasticity of the shaft, coupling the PM10MIW motor and the arm. It is observed in the form of small oscillations of the arm in a final phase of a rotation movement.

A mathematical model of the tower crane it is not a trivial case. Mathematical aspects of modeling a behavior of the crane give a lot of opportunities to study various methods of modeling physical objects for control purposes[8][1]. The basic problems, the students faces:

- modeling of the PWM type control signal and its reference to the force control,
- modeling of the motor equipped with a gearbox,
- modeling of static and viscotic friction,
- designing an LQ controller basing on a linearised model,
- dumping payload oscillations,
- developing strategies of a save transport of the payload,
- comparing the payload deviation measure system to alternative methods, e.g. accelerometer implementation.

According to mathematical equations [8], a movement of the oscillating payload can not be decomposed into two planes: X, and Y. A deviation of the payload in the X plane influences on the payload movement in the Y plane. Experiments show, this effect can be omitted only when the deviation of the payload is lower than approx. five degrees.

V. Conclusions

The tower crane was designed to serve as an educational system. It is connected in it:

- scientific and technical environment MATLAB&Simulink
- system for control in real time, consisting of software parts and hardware
- modern electromechanical solutions like: measuring position sensors, proximity sensors, DC motors, gearboxes, couplings, bearings including thrust bearings.

The whole is a complex mechatronic system, allowing a successful practice with students of many scientific problems in the field of automation, robotics, modeling, identification, electronic, mechanical and first of all, the theory of control.

VI. References

[1] A. Al-Mousa , „Control of Rotary Cranes Using Fuzzy Logic and Time-Delayed Position Feedback Control", Virginia Polytechnic Institute and State University, 27-th November 2000

[2] A. Benhidjeb, G.L. Gissinger: "Fuzzy control of an overhead crane performance, comparison with classic control", Control Eng. Practice, Vol. 3, No. 12, pp. 1687-1696, Pergamon 1995

[3] J.J. Hämäläinen, A. Marttinen, L. Baharova, J. Virkkunen.: "Optimal path planning for a trolley crane: fast and smooth transfer of load", IEE Proc. - Control Theory Appl., Vol. 142, No. 1, 1995

[4] D. Lewis, G.G. Parker, B.Driessen, R.D.Robinett: "Command Shaping Control of an Operator in the Loop Boom Crane", Proc. American Control Conference, Philadelphia, Pennsylvania 1998

[5] D. Marchewka, A. Turnau "Identification of 3DOF manipulator oriented to time-optiml control", 5th international conference Computer Methods and Systems: Vol. 2, pp.125–128, Kraków, Poland, 14–16 November 2005

[6] D. Marchewka, M. Pauluk "3DOF robot, crane 3D – robust and optimal control" , Fundamental problems of power electronics electromechanics and mechatronics" - proceedings XIII symposium : ISBN 83-922242-6-4. — pp. 166–173, Wisła, Poland, 14–17 december 2009

[7] M. Pauluk "Robust control of 3D crane", proceedings of the 8th IEEE international conference on Methods and Models in Automation and Robotics : Vol. 1, pp. 355–360, Szczecin, Poland, 2–5 September 2002

[8] M. Pauluk „Mathematical model of the three-dimensional crane" Automatyka : półrocznik Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie ; ISSN 1429-3447. — 2002 t. 6 vol. 1 pp. 69–102. Kraków, 2002

[9] http://www.inteco.com.pl

[10] http://www.micromotors.eu

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

The wireless communication in the walking robot application

Sylwester Cyrwus
AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics
Mickiewicza 30 av. 30-059 Kraków, Poland
e-mail: cyrwus@student.agh.edu.pl

***Abstract*—The paper presents the identification process of time delays in the development environment created for the wireless control of a hexapod, the six-legged walking robot. RFM12B transceiver modules are used for communication between the host computer and the FPGA board. The board was designed to control all 18 servo motors. The software and hardware components are described in detail. The advantages and disadvantages of the designed communication system as well as the servo motors driver based on the FPGA circuit are listed. Various parameters of the control system are investigated. The experiments that allow statistical analyses of time delays in the communication system are described and the results are included. Thorough analysis of time delays are presented in numerical and graphical forms.**

***Index Terms*—hexapod, walking robot, identification, wireless communication, FPGA**

I. Introduction

The hexapod is a walking robot equipped with the six identical legs. Each of them consists of three servo motors. This kind of construction is a simplification of natural six-legged insects like a cockroach. Hexapods may be used to test biological theories about the insect locomotion, motor control and neurobiology. Insects are chosen for the biological tests because their nervous system is simpler than other animal species. Figure 1 shows the three-dimensional model of the hexapod created using the Autodesk Inventor 2009® software application.

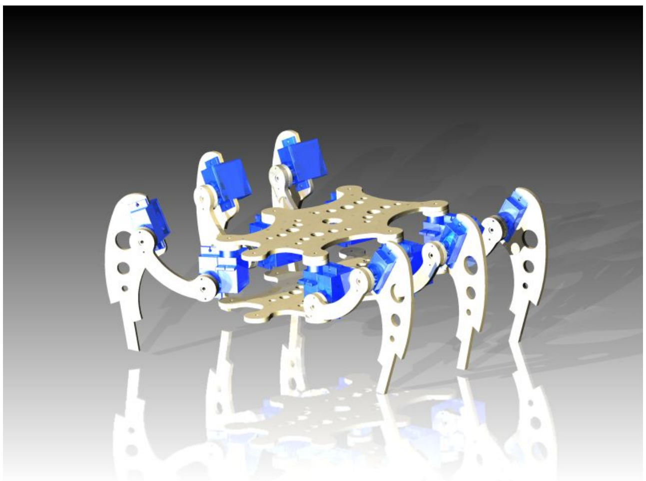


Figure 1. Model of the hexapod - six-legged application

The Turnigy TR-1160A mini servo motors, with a range of movement from about -70 to +70 degrees are used in the application. The control signal of the servo motor is a square wave, similar to the PWM signal. The width of its “high” level corresponds to the desired servo motor position. Figure 2 shows the control signal with expected time and voltage values. The position of the servo motor is measured by the built-in potentiometer. A voltage drop on the potentiometer gives direct information about the servo motor position.

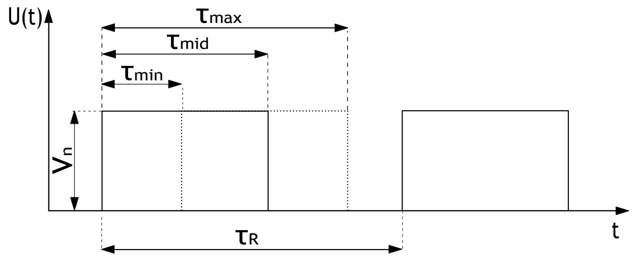


Figure 2. The servo motor control signal.
V_n- 3V - voltage value, τ_{max}- 2.5ms - the servo motor right boundary position, τ_{mid}- 1.5ms - the servo motor mid-position, τ_{min}- 0.5ms - the servo motor left boundary position, τ_R- 20ms - square pulse refresh time

The development environment requires a minimum of 18 PWM-like actuators and the ability to generate continuously the PWM-like signals. These requirements are the result of the fact that the system has to continuously control all of the robot degrees of freedom.

The identification of the time delays in the wireless communication is necessary for the correct implementation of the control algorithm and the proper design of the structure of the wireless communication.

II. The architecture of the development environment

The designed architecture of the development environment is splitted into two main layers: the hardware and the software. The hardware layer is responsible for the setting required position of each servo motor. The software layer is responsible for the algorithm implementation and the hardware configuration management.

A. The hardware layer

The hardware layer is an intermediate layer between the robot and the host computer. The FPGA circuit was applied in this part in order to be used in scientific research. This circuit gives the opportunity to control all the servo motors independently. In the current version of the environment the FPGA circuit is used to generate PWM-like signals to control the angle of the servo motors. The circuit could also be equipped with differential analog to digital converters used to the feedback control of the servo motors position. The AVR Atmega8 microcontrollers and the RFM12B transceivers are used to communicate the host computer with the FPGA circuit.

B. The software layer

The control PC programme was written in Microsoft Visual C++ 2008 Express Edition® environment and runs under the Microsoft Windows® operating system. The main advantage of the Microsoft Windows® operating system and the C++ programming language is the popularity among students. Nevertheless the Microsoft Windows® is not the hard real-time operating system what means that several time delays may appear which are unable to predict.

A host PC is used to control the movement of the robot. The AVR Atmega8 microcontroller analyses the data received from the host computer, calculates the following position of each servo motor and transfers this information to the FPGA using the RFM12B transceivers. The FPGA sets the direct position of each servo motor continuously due to the information from the host computer. The ISE WebPack and the GCC compiler were applied. For the time delay analysis MATLAB® software was used.

III. The communication

The wireless communication between the host computer and the FPGA board is conducted using the RFM12B transceiver modules and the AVR Atmega8 microcontrollers. Figure 3 shows the scheme of the development communication architecture.

The data transmission can be divided into five main parts:

a) during the first part, the host computer sends 8-bit information about the following movement of the legs using the RS-232 standard. This information includes the direction, radius and speed of movement. The parameters of the RS-232 standard are specified below:

- baud rate: 57,600
  - byte size: 8
  - stop bits: 1
  - parity: none
  - flow control: none

b) the next step is to calculate the following position of each servo motor according to the information received from the host computer. The AVR Atmega8 microcontroller analyses all the necessary data. After having finished all calculations the information about the next position of each servo motor is transmitted to the RFM12B transceiver using the SPI protocol

c) the first RFM12B transceiver sends the data to the second RFM12B transceiver. The wireless communication is conducted with the following parameters:

- band: 433 MHz
  - operation frequency: 430.8 MHz
  - base band bandwidth: 134 kHz

d) the second RFM12B transceiver sends the received data to the next AVR Atmega8 microcontroller using the SPI protocol

e) during the last part of the data transmission the AVR Atmega8 microcontroller divides all the received data into 4 bit parts and sends it to the FPGA Spartan III microcontroller.

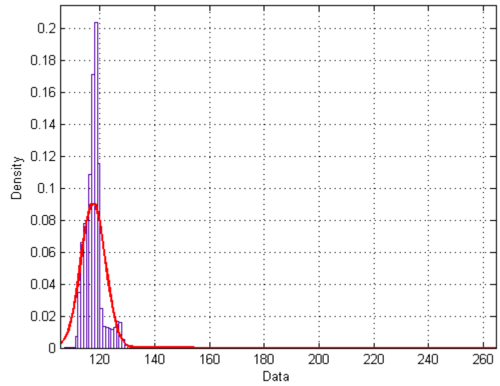


Figure 4. Density distribution of the time delays during data transmission for a distance of one metre and fifty centimetres, without any obstacles between the transmitter and the receiver.

IV. Experiments

Several experiments have beed carried out. During all of them the time delays in the wireless communication were measured with the GetSystemTimeAsFileTime function. This function is exported by kernel32.dll and retrieves the current system date and time. The information is in Coordinated Universal Time (UTC) format. Experiments were repeated for the different distances with or without such obstacles as a wooden board, or a glass or concrete wall. The analysis of the results will indicate the effect of these factors on the time of the communication.

During the first experiment the distance between the transmitter and the receiver was one metre and fifty centimetres, without any obstacles. Figure 4 presents density distribution of the time delays during the experiment. The time delay was measured 20,000 times.

In the following research the distance between the transmitter and the receiver was not changed but the receiver was placed inside the three centimetres thick wooden box. Figure 5 presents density distribution of time delays during the experiment. The time delay was measured 20,000 times.

Another 20,000 measurements were made with the same distance between the transmitter and the receiver as the previous two, but this time the transmitter was placed behind the ten centimetres thick concrete wall. Figure 6 presents density distribution of time delays during the experiment.

Two further experiments were made for the distance of ten metres between the transmitter and the receiver sequentially without any obstacle and with a ten centimetre concrete wall. The distributions of the time delays during the data transmission in these experiments are presented on figure 7 and figure 8.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

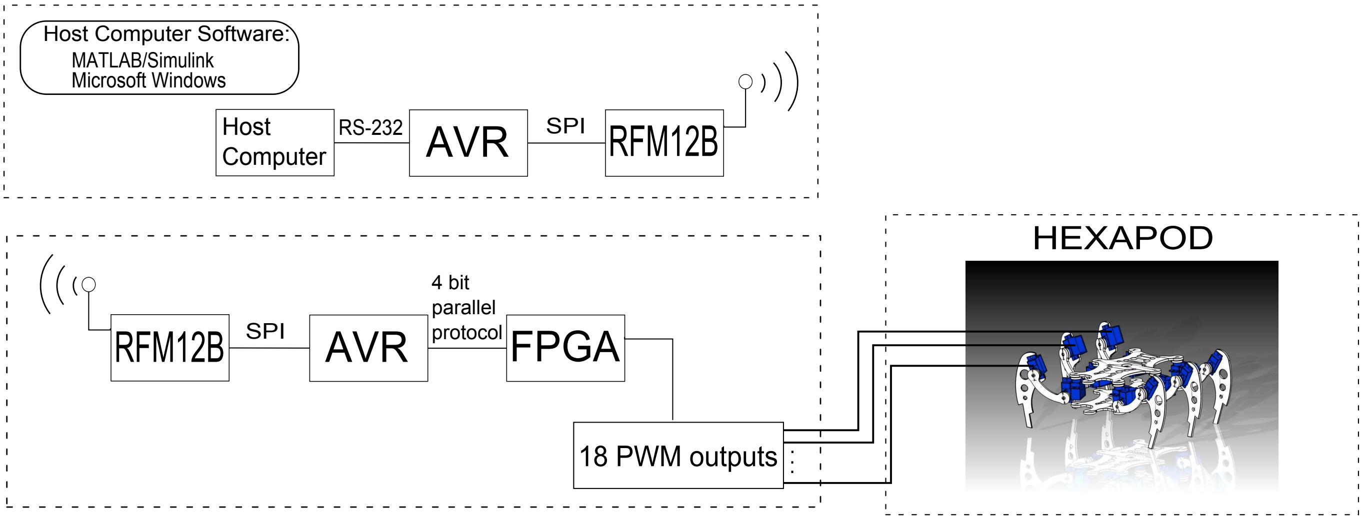


Figure 3. The scheme of the development communication architecture.

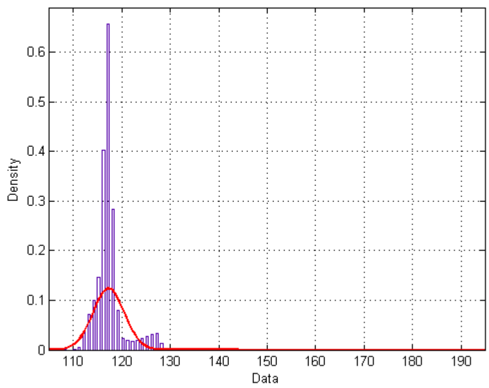


Figure 5. Density distribution of the time delays during data transmission for a distance of one metre and fifty centimetres, with the receiver closed inside the three centimetres thick wooden box.

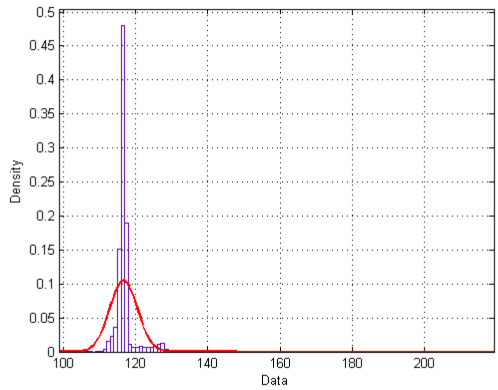


Figure 7. Density distribution of time delays during data transmission for a distance of ten metres, without any obstacles between the transmitter and the receiver.

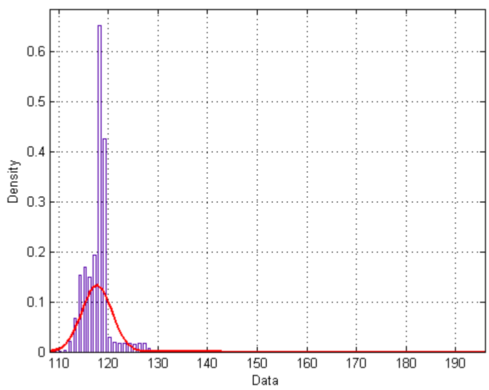


Figure 6. Density distribution of time delays during data transmission for a distance of one metre and fifty centimetres, with the transmitter placed behind the ten centimetres thick concrete wall.

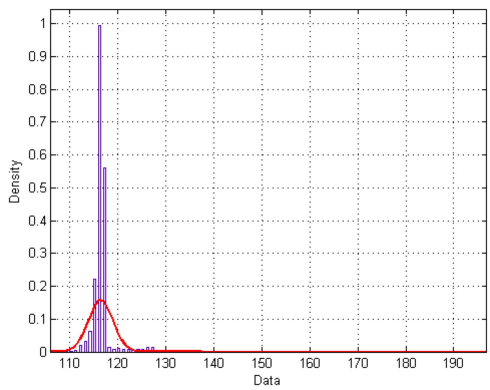


Figure 8. Density distribution of the time delays during the data transmission for the distance of ten metres, with the transmitter placed behind the ten centimetres thick concrete wall.

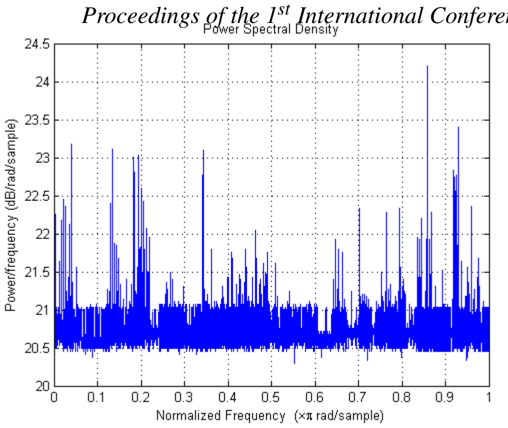


Figure 9. Spectral power density of the time delays during the data transmission for a distance of one metre and fifty centimetres, without any obstacles between the transmitter and the receiver.

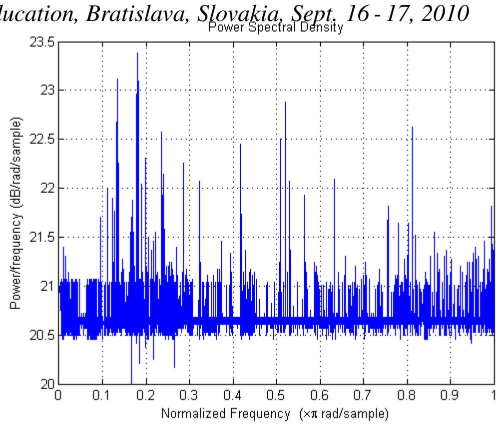


Figure 12. Spectral power density of the time delays during data transmission for a distance of ten metres, without any obstacles between the transmitter and the receiver.

Figure 10. Spectral power density of the time delays during the data transmission for the distance of one metre and fifty centimetres, with the receiver closed inside the three centimetres thick wooden box.

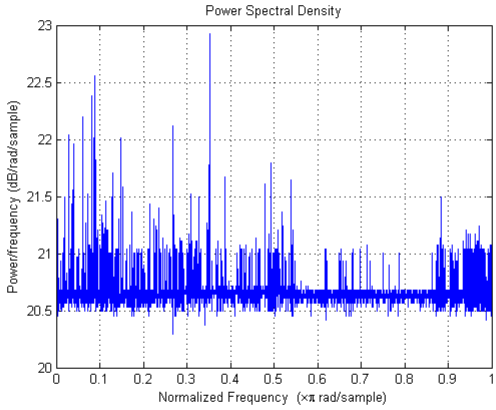


Figure 13. Spectral power density of the time delays during data transmission for a distance of ten metres with the transmitter placed behind the ten centimetre concrete wall.

The next five figures present spectral power density of the time delays measured during the experiments.

V. Summary

Taking into consideration all experiments that were carried out, it can be concluded that neither the various obstacles nor the distance between the transmitter and the receiver have the influence on the time delays in the wireless communication. The expected value of the time delays was nearly the same during all tests. For the proper modelling of the communication system there have to be included the constant delay in the form of $e^{-s\tau}$ with additional delay in the form of the random variable with the parameters such as the identified time delays.

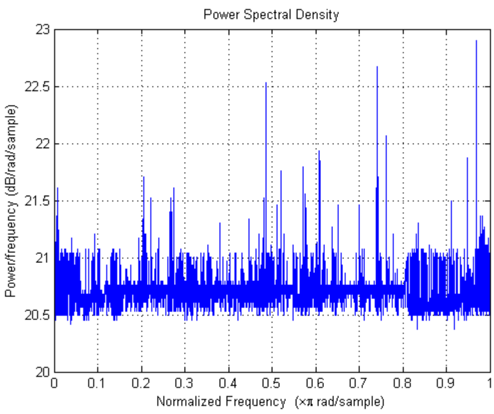


Figure 11. Spectral power density of the time delays during data transmission for a distance of one metre and fifty centimetres, with the transmitter placed behind the ten centimetres thick concrete wall.

The experiments show that the identification of the time delays in the wireless communication was necessary for the correct implementation of the control algorithm and the proper design of the structure of the wireless communication. The results indicate the need to re-design of the actual control structure. In the current version the first AVR Atmega8 microcontroller after receiving from the host PC the information about the following movement calculates the position of each servo motor and record this data in 288 bits (16 bits for each of 18 servo motors). This information has to be sent via the SPI protocol two times (from the first AVR Atmega8 microcontroller to the RFM12B transceiver and from the second RFM12B transceiver to the next AVR Atmega8 microcontroller), through the 4 bit parallel protocol from the second AVR Atmega8 microcontroller to the FPGA and wirelessly between the RFM12B transceivers. The new version of the communication architecture has to be designed to avoid the time consuming 288 bit data transmission

Experiment	Distance [m]	Obstacle	Number of samples	Minimum time [msec]	Maximum time [msec]	Expected value [msec]
1	1.5	none	20,000	107	263	117.5089
2	1.5	3 cm wooden box	20,000	106	194	117.3003
3	1.5	10 cm concrete wall	20,000	109	195	117.6708
4	10	none	20,000	100	218	116.6832
5	10	10 cm concrete wall	20,000	107	196	116.4334

Table I
SUMMARY

from the first AVR Atmega8 microcontroller to the FPGA. In the re-designed structure the 8 bit information about the parameters of the following movement will be sent from the host PC up to the second AVR Atmega microcontroller. Only then all the calculations will be done and data will be transmitted to the FPGA. This modification will have a considerable impact on the time delays in the wireless communication.

Due to the low AVR Atmega8 microcontroller computing power the following position of each servo motor during the different movements will be recorded in the LookUpTables. After having received the information about the parameters of the following movement the AVR Atmega8 microcontroller will retrieve the relevant data form the LookUpTable and rend it.

Despite such long-time data transmission, the system is slow enough to work properly. Even the 200 ms time delays between the successive data have no appreciable effect on the control of the robot.

This tested wireless communication technology will be used to control robots from a PC, and can be applied in laboratories as well as in low cost student projects.

REFERENCES

[1] M. Piątek and P. Piątek, *Development environment for the walking robot application,* Computer Methods and Systems 2009, 295-299, November 2009, Krakow.

[2] M. Piątek, *Identification of the servo motor used in the walking robot,* Półrocznik AGH Automatyka 14(1), volume 14, 2010.

[3] Praca zbiorowa pod redakcją J. Kalisza, *Język VHDL w praktyce*, Wydawnictwa Komunikacji i Łączności, Warszawa, 2002.

[4] M.W.Spong and M.Vidyasagar, *Dynamika i sterowanie robotów,* Wydawnictwo Naukowo - Techniczne, Warszawa, 1997, transladed to Polish.

[5] J. Grębosz, *Symfonia C++*, Editions 2000, Kraków, 2008.

[6] A. Witkowski, Mikrokontrolery AVR programowanie w języku C przykłady zastosowań, Polska Akademia Nauk, Katowice, 2006.

[7] B.Mrozek, Z.Mrozek, *Matlab and Simulink, User Manual*, Helion, 2004, in Polish.

[8] T. Zielińska, *Maszyny kroczące*, Wydawnictwa Komunikacji i Łączności, Warszawa, 2005.

[9] A.Turnau, *Model identyficaion dedicated to the time-optional control*, In 17th IFAC World Congress, 2655-2660, July 2008.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Navigation of autonomous mobile robot using ultrasonic and infrared sensors

Martin Dekan, František Duchoň
Institute of Control and Industrial Informatics
FEI STU
Bratislava, Slovakia
dekdekan@gmail.com, frantisek.duchon@stuba.sk

Abstract – **The aim of this paper is to briefly describe proposed algorithms for an autonomous mobile robot. These algorithms concern data processing from sensors, description of environment from these data and finally navigation on these data. Results from these processes are based on simulation of real mobile robot system. On proposed algorithms can be showed principle of ultrasonic and infrared sensor, principle of environment mapping and basic navigation of real mobile robot. This knowledge can be used in education to show basic principles of robot motion and navigation.**

Keywords - ultrasonic sensor, infrared sensor, occupancy grid, reactive navigation

I. Introduction

Collision free mobile robot navigation in the environment is the basic problem of autonomous systems [6]. This problem occurs either when following the path of mobile robot in a known or partially known environments as well as when crossing completely unknown environment. It is also necessary for creating maps or searching task objective.

The robot used for simulations is mobile robot for indoor environment. It is differentially driven robot with three wheels, which two of them are driving wheels and one is a relieving wheel. Model of the robot can be seen at Fig. 1.

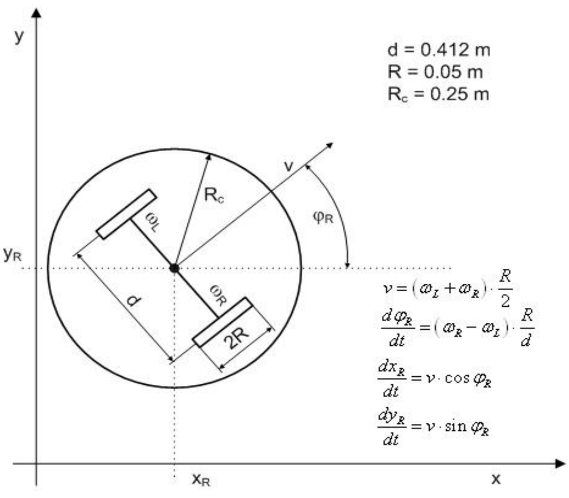


Figure 1. Model of the robot

II. Representation of the environment

We will be using occupancy grid for the representation of the environment [1] [2] [3] [5]. Occupancy grid provides an effective platform for fusion of information from multiple sensors and sensing positions. After receiving the information from the sensor, sensor model is applied to the grid and each cell is updated. The value of the cells takes value -1 for the cell, of which we do not have knowledge so far; <0,1> in the case of the cells that are already known to us, where 0 means the cell that is entirely free to cross and 1 stands for a cell where there is definitely an obstacle. Cell size is 10x10 cm.

Mobile robot intended for navigation uses nine ultrasonic rangefinders (Fig. 2).

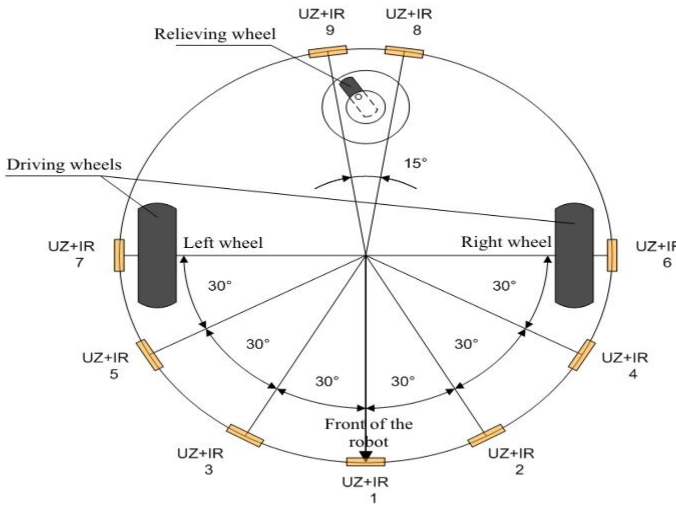


Figure 2. Position of sensors and wheels on the mobile robot.

In order to simplify the calculations, the rotation of the grid regarding the xy axis remains unchanged so that the movement of the robot is represented only by shift of data in the occupancy grid (Fig. 3).

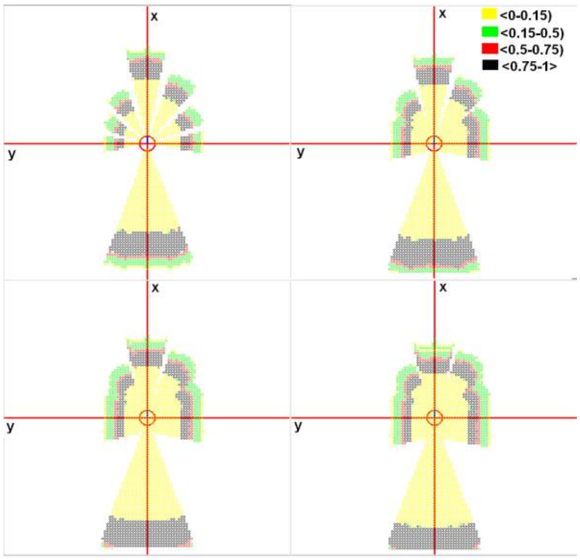


Figure 3. Occupancy grid in the various steps of mobile robot movement in C + + using the OpenCV library. Probabilities of obstacle are expressed by different colours.

A. Ultrasonic sensor in the occupancy grid

We used measuring range from 0.5 meter to 5 meter for the ultrasonic sensor [7].

It is written into the grid by the formula:

$$f_k(\upsilon, r) = \begin{cases} \cos(3\upsilon)\left(\dfrac{1-\tanh(r-r_x)}{2}\right) & |\upsilon| \le 12.5^\circ,\ r \ge r_x \\ 0 & |\upsilon| \le 12.5^\circ,\ r \le r_x \end{cases} \qquad (1)$$

Where the data r_x is measured by the sensor (Fig. 4).

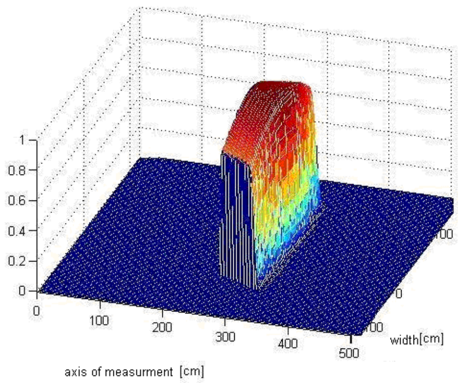


Figure 4. Uncertainty model of the ultrasonic sensor in the occupancy grid.

B. Infrared sensor in the occupancy grid

The infrared sensor takes into consideration only one point of an obstacle. Therefore, it is unsuitable for this type of occupancy grids and implementation of the proposed navigation algorithm – it may cause overlooking of the essential data. To prevent this, the infrared sensors are recorded into the grid on larger area.

For infrared sensor we used a measuring range from 2 cm to 40 cm (Fig. 5).

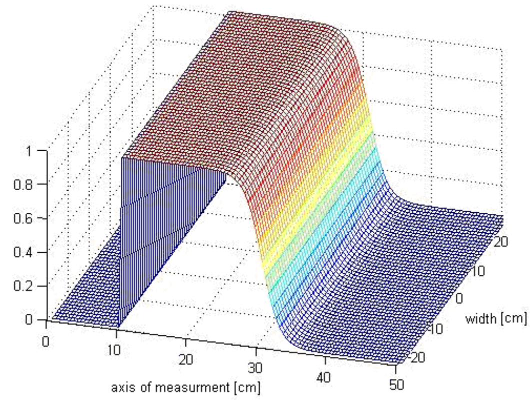


Figure 5. Infrared sensor in the occupation grid.

III. REACTIVE NAVIGATION

We will use the Wandering standpoint algorithm (Fig. 6) for the reactive navigation because of its simplicity and efficiency [4].

The principle of operation:

1. Go directly to the goal if possible.

2. In case of an encounter with an obstacle, count angle-free path for turning left and right .

3. Choose the smaller angle and follow the path around the obstacle.

4. Go to step 1.

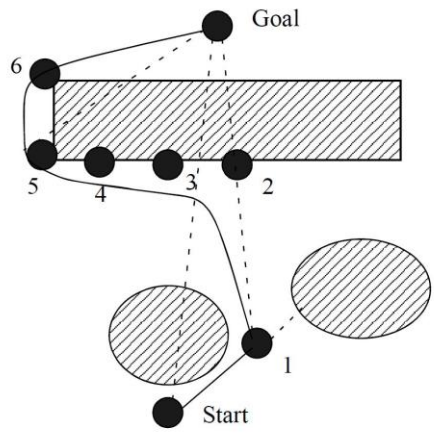


Figure 6. Principle of Wandering standpoint algorithm [4].

The disadvantage of this algorithm is the possibility of loops in some obstacle layouts.

Therefore, we extended this algorithm with the memory of the conflict. The position, in which the robot encounters an obstacle for the first time is called conflict point. If the new trajectory from a place occupied by our robot contains/involves a collision point the trajectory will be affected by this memory. In other words, the algorithm counts the new direction so that the trajectory will eventually avoid this place.

The principle of operation is shown in the following diagram:

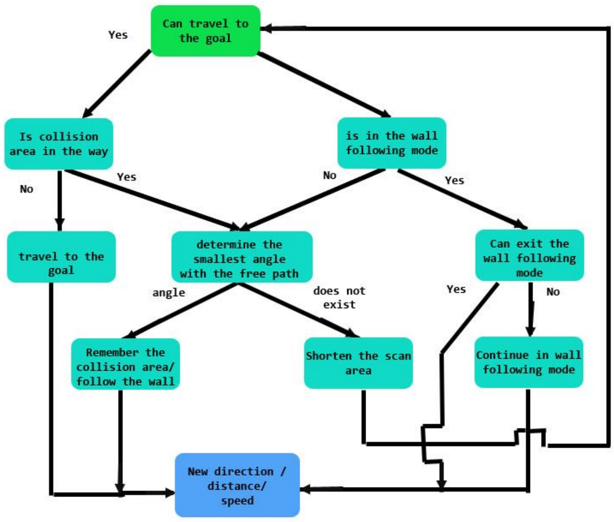


Figure 7. The principle of operation of the reactive navigation.

The direction of free path is designated as a rotation angle, where the area around the robot (Fig. 8) contains no object.

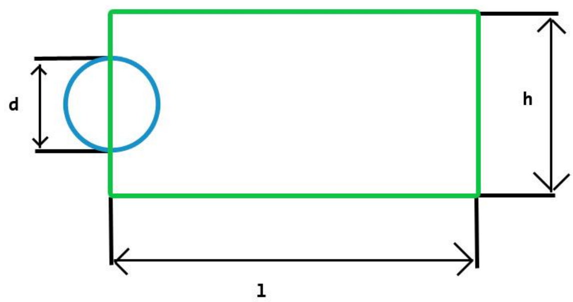


Figure 8. The principle of the safety zone of a mobile robot

Where *d* is the diameter of a mobile robot, *l* is the length of the zone and *h* is the width of the zone.

IV. Experimental Results

In these tests the length of the safety zone *l* is set according to measured distance of the sensors:

Minimal distance>1,3 m *l*=180 cm

0,75< Minimal distance <1,3 m *l*=155 cm

0,3< Minimal distance <0,75 m *l*=130 cm

Minimal distance <0,3 m *l*=105 cm

In order to verify functionality for different types of environments, the proposed algorithm was tested gradually for the cases of three maps. The first environment contained obstacles greater than the size of the robot itself and the distance between the barriers allowed passage of mobile robots (Fig. 9). The second map contained denser distribution of smaller barriers (Fig. 10). The third map contained a classic trap situation (Fig. 11). And finally, the fourth map contained trap situation with some other simple obstacles (Fig. 12).

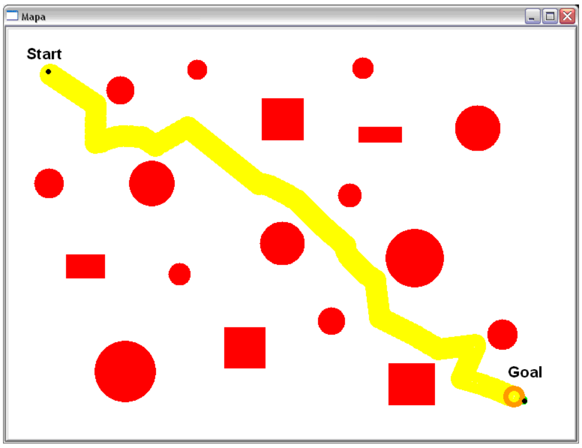


Figure 9. First testing map

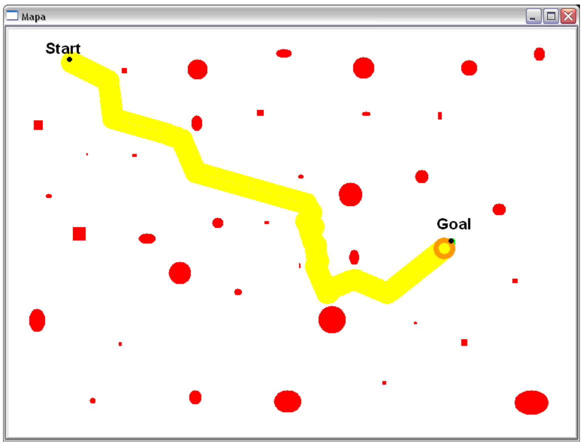


Figure 10. Second testing map

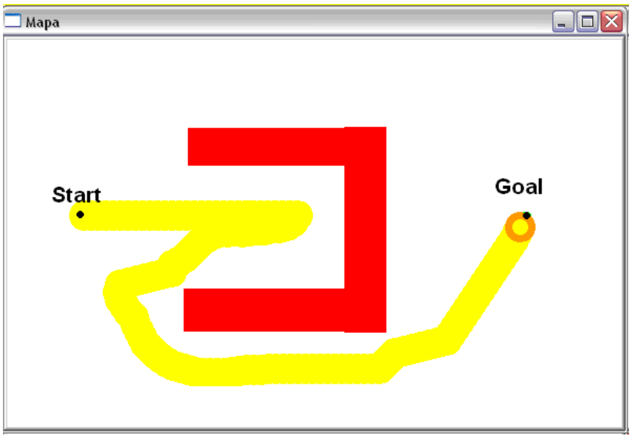


Figure 11. Third testing map

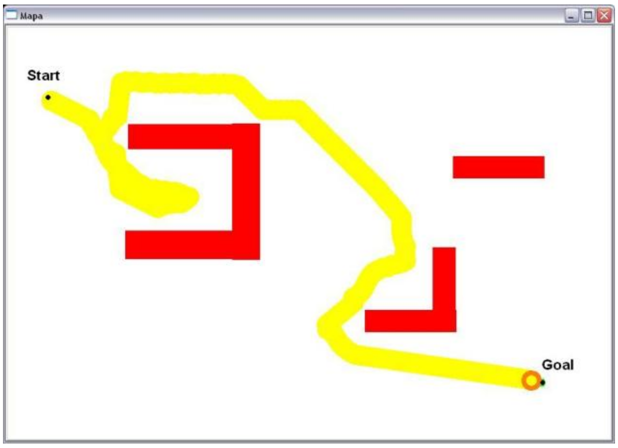


Figure 12. Fourth testing map

As shown in four figures, with use of proposed algorithm, mobile robot has found path from start to goal in all testing environments.

V. Conclusion

The aim of this paper was to present and verify by experiment the algorithm for an autonomous mobile robot. As algorithm for data processing from sensors was presented uncertainty model of ultrasonic rangefinder, which takes into account the width of the scanning angle. As description of the environment was proposed occupancy grid, whose main advantage is its simplicity and efficiency. And finally, navigation on the basis of these data was realized through modification of the wandering standpoint algorithm.

Acknowledgment

This work was supported by MS SR under the contract VEGA 1/0690/09 and KEGA 3/7307/09.

References

[1] Choset Howie, et al. Principles of Robot Motion : Theory, Algorithms, and Implementation. [s.l.] : The MIT Press, 2005 . ISBN 0-262-03327-5.

[2] Siegwart Roland, Nourbakhsh Illah R. Intorduction to Atuonomous Mobile Robots. [s.l.] : MIT Press, 2004. ISBN 978-0-262-19502-7.

[3] Murphy Robin R. Introduction to AI Robotics. MIT Press, 2000. ISBN 0-262-13383-0.

[4] Bräunl Thomas. Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems. Springer, 2006. ISBN 13 978-3-540-3431.

[5] Koniar Dušan, Hargaš Libor, Hrianka Miroslav, Bobek Viktor, Drgona Peter, Fibich Peter. Kinematics analysis of biomechanical systems using image analysis. Metalurgija (Metallurgy), Vol. 49 (2/2010). ISSN 1334-257.

[6] Zdenko Bobovský. Biological principles and their usage in metamorphic robotic systems (Biologické princípy a ich využitie v metamorfných robotických systémoch). Novus scientia 2007. ISBN 978-80-8073-922-5.

[7] Hanzel Jaroslav, Jurišica Ladislav. Experimental identification of parameters of ultrasonic rangefinder. AT&P Journal Plus, vol. 2/2006. ISSN 1336-5010.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Parallelizing the Precomputed Scan Matching Method for Graphics Card Processing

Petr Schreiber, Vít Ondroušek, Stanislav Věchet, Jiří Krejsa
Brno University of Technology

Brno, Czech Republic
petrschreiber@gmail.com, ondrousek@fme.vutbr.cz, vechet.s@fme.vutbr.cz, krejsa@fme.vutbr.cz

Abstract— **Certain methods solving mobile robot localization problem are reliable, but computationally expensive. One possible way how to increase the speed of necessary calculations is to use parallel approach and use graphics card to speed the processes up. Methods such as Precomputed Scan Matching Method (PCSM) or Particle Filters are suitable for parallel processing. PCSM method requires processing the map prior to localization and so far such processing had to be done offline, while the new approach brings computational time reduction in order of a magnitude. The paper describes the modifications of PCSM method and its implementation suitable for modern ATi Radeon and NVIDIA GeForce graphics card series.**

Keywords- Localization; PCSM; GPGPU; OpenCL

I. INTRODUCTION

Determination the position of mobile robot is essential issue in both indoor and outdoor robotics. This issue, called localization, can be divided into two main groups: position tracking and global localization. Position tracking uses information about the motion of the robot (usually from IRC sensors) together with some kind of outer sensor to keep track of the robot position changes. Position tracking therefore requires the initial position of the robot to be known. Outer sensors depend on the environment, compass and GPS receiver are usually used in outdoor, beacons and some sort of feature extractor are usually used indoor. Some sort of probabilistic filter (Extended Kalman filter, Particle filter, etc.) is commonly used as the engine that estimates the robot position as the fusion of motion model and measurement from the sensors.

Global localization must provide the information about the position of the robot with little or no data regarding the initial estimate and it should cope well with multimodal distributions of the estimate (several hypothesis with high probability are possible). Global localization is usually computationally more expensive and when lower power computational means are used, high demands usually limit the speed of the robot or expand the time span of localization steps to higher values, or some of the computation must be performed offline, if possible. The Precomputed Scan Matching (PCSM) method described below is a representative of such a method.

In order to use computationally more demanding methods in commonly available hardware, that is available for students and university robot building teams the methods must be either optimally coded or if possible certain parallelization must be performed. The paper gives an overview of PCSM method computed using graphics card processing enabling to speed up necessary routines and run the method on commonly available hardware. The paper is organized as follows: first the method itself is described, possible ways of performance enhancement are mentioned, the OpenCL technology is overviewed and details regarding the implementation and results are finally given.

II. PRECOMPUTED SCAN MATCHING

Precomputed Scan Matching (PCSM) is method for indoor robot localization, originally developed by Stanislav Věchet on Brno University of Technology[1]. The method is designed for mobile robots equipped with some kind of proximity sensor, usually the laser rangefinder, with the scan range of 180° or more (see Fig 1.).

The execution of PCSM is split into two stages: the precomputing and the localization stage. The first stage performs virtual scan of the map of environment, by shooting 360° rays from points where robot is designed to operate. These rays collide with environment model, and the output data set represents cached information which can be easily accessed in localization stage, which is based on comparison of real scan and precomputed scans. The pair of two scans closest to each other indicates the expected position of the robot. Details about the method itself can be found in [1].

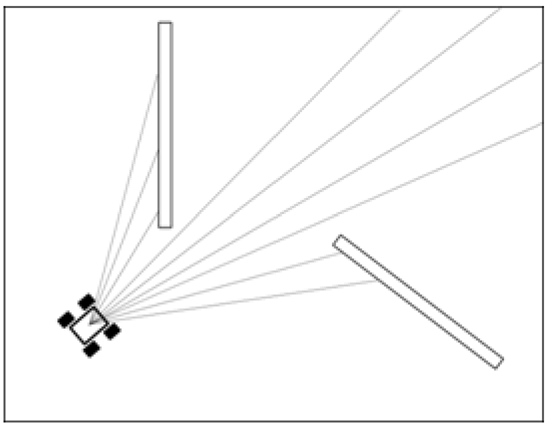


Fig. 1. Distances measured in real environment

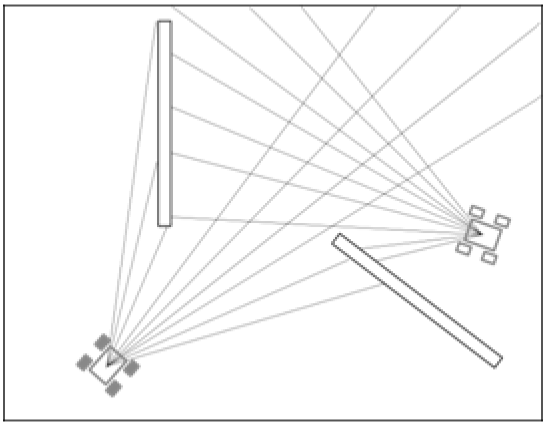


Fig. 2. Building the set of precomputed scans from numbers of virtual positions in environment map

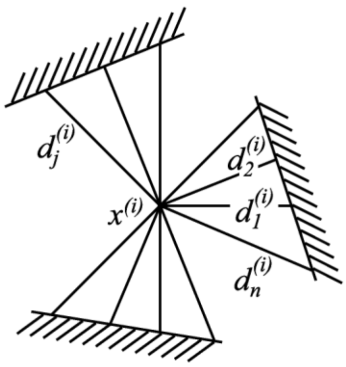


Fig. 3. Neighborhood scan definition

The comparison of real and precomputed scan is essential for PCSM method performance. Comparing function is called Match and it defines the difference of two scans by function

$$M(x) = r(x, a, S)$$

where *x* is the state of the robot, *a* is a real scan acquired by sensors and *S* is the set of precomputed scans. The Match is calculated based on complete neighborhood scan *d* defined as a set of single distances,

$$d = \{d_j\}_{j=1,\dots,n}$$

where *dj* is the distance of single beam.

The set *S* is a number of m precomputed scans which are stored as

$$S = \{x^{(i)}, d^{(i)}\}_{i=1,\dots,m}$$

d(i) is precomputed scan obtained from virtual position *x(i)*

Final match is calculated for all precomputed scans stored in set *S*. As a result, the one with the highest match (represented by the minimum of match function) is returned. The match function is defined as follows

$$r(x^{(i)}, a, S) = \sum_{j=0}^{n} \left(d_j^{(i)} - a_i\right)^2$$

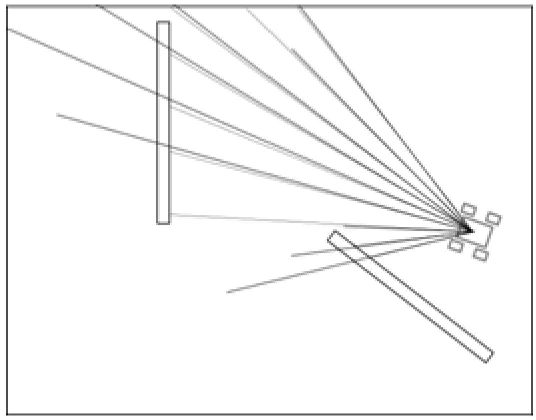


Fig. 4. Mismatch of the real and precomputed scan

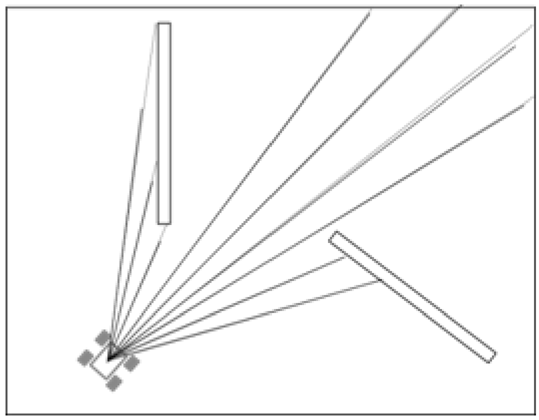


Fig. 5. Sufficient match of two scans

Two different examples of possible matches are shown on Fig. 4 and Fig. 5, illustrating mismatch of the scan (robot is in the position not corresponding with measured data) and successful match (robot is located in position corresponding with the measurement)..

As clear from the description above, the localization itself is based on comparing the real sensor data with precomputed data, and performs very well even on slower CPUs[2]. Originally the method was designed to have the precomputation done offline, because it is very computationally intensive, and can take lot of time to perform for larger environments with high precision. The precision is determined by how close the precomputation points are placed. The goal for further optimization was to make this stage fast enough to be performed directly on mobile robot, removing the need for using high performance workstation.

III. POSSIBLE OPTIMIZATION APPROACHES

While the problem of colliding the rays with obstacles seems to be trivial, the main problem consists in the fact line to line collision has to be performed millions of times. Optimizing this operation by breaking high level code into assembly with heavy use of SIMD instructions could be seen as evident choice, but this approach was completely avoided because it would lead to complication with optimized code maintainability, not speaking of lack of brute force of Intel Atom chip.

The target robot is built on top of Nvidia ION platform, which besides the mentioned Atom processor provides programmer with Nvidia GeForce 9400 as well. While this GPGPU (General Purpose Graphics Processing Unit) is completely ignored by hardcore gamers for its sometimes insufficient graphics performance, the raw computional performance of its 16 stream processors still offers significant advantage over Atom.

Programming the GPGPU today has become much easier comparing to past. There are four major technologies to consider – ATi Stream[3], Nvidia CUDA[4], DirectCompute and OpenCL. The Stream and CUDA, while proven to deliver outstanding performance, are technologies locked to specific hardware, so making optimization like this could badly pay off in case we would choose to change platform in future. DirectCompute is promising technology, but limited to Windows OS.

For these reasons the OpenCL was chosen, as the most independent solution from both software and hardware standpoint.

IV. About OpenCL

OpenCL is technology currently maintained by Khronos organization, which also handles other well known standards, such as OpenGL. OpenCL stands for Open Computing Language and as a such you can use it for programming of both CPUs and GPUs. The range of supported GPGPUs is slightly wider on Nvidia side, where you can run OpenCL on anything from GeForce 8 and up, while on AMD side only the latest ATi Radeon HD 4000 or better the HD 5000 series are supported.

The GPGPUs are the device of choice for data parallel tasks, thanks to their big amount of processing cores, comparing to CPUs. OpenCL allows us to use data parallel programming, which is exactly what we needed for tweaking the PCSM. The architecture of OpenCL can be viewed in 2 parts: run-time host API and OpenCL C language with compiler. The API allows embedding this technology basically to any language, on Windows platform it is every language capable of interacting with industry standard DLL.

Programmer can write so called kernels (procedures executed in parallel) in OpenCL C which is high level language derived from C99, with few restrictions and extensions comparing to this standard. You can learn more about this language and OpenCL in general in[5].

The important point is that the code written in OpenCL is not graphics card specific at all. It is no longer necessary to use graphics specific shader programs for general purpose computations. OpenCL C is also relatively high level language, which, in author's opinion, means significant advantage in maintainability over optimizations written in assembly.

V. Limitations of GPGPU

While the OpenCL language itself is not GPU specific, when programming such a devices you have to keep few facts in mind:

- Double precision performs well only on high end models
- Excessive code branching can hit performance
- The resources usable on each GPU slightly differ
- Precompiling is not an safe option as of Q2 2010

Although the 9400GT is model which does not expose the double precision functionality, the first mentioned problem posed no danger for realization of PCSM. Changing the units of distance to millimeters reduced requirement of precision for many digits after decimal points.

The second mentioned issue is indeed observable when working with OpenCL, and is caused by fact GPU programming, even the pure graphic one, did not allowed dynamic branching for long time. Current models support this feature, but the performance hit is observable, yet not critical.

The problem of GPU running out of resources is the tricky one, and will be further discussed in paragraph VI.

The compilation of GPU code is performed by driver, and as the OpenCL devices are very diverse, there is no standard for binaries. In theory, when using the same code on the same device, it would be possible to rely on once compiled binaries. But as dramatic progress in quality of OpenCL implementations can still be observed today, the preferred way is to perform fresh compile before execution. As the kernel code is usually very brief, with no include files, the compilation is performed in times typically under 1 second, which does not represent big problem.

VI. Precomputation Design with OpenCL

As mentioned in paragraph II, the precomputation phase first find the points where robot can appear during its journey and then calculates the 360° collision in each of this point. This allows splitting the calculation into two kernels. Another reason to do this is to simplify the kernels to reduce GPU resources usage. The resources question is one of the problematic sides of GPU implementations at the time of writing this article, as they fail to report whether the compiled kernel is too resources intensive for the given device. Any auxiliary functions the kernel uses are currently handled as inline, which means the modularization of executions may suffer from excessively large kernel size after the expansion of inline calls.

A. FindPointsForCalculation Kernel

The first kernel, as the name suggests, seeks for the points where robot will operate. That means any place which is not covered by visible or invisible obstacle. The result of the process is shown on Fig. 6.

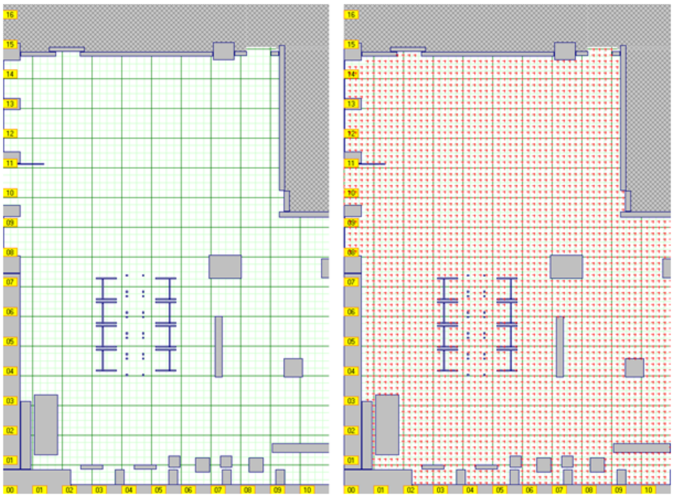


Fig. 6. Initial search for possible robot position

This task is so trivial, it would be possible to perform this on CPU with similar execution time. The reason why this is done on GPU is obvious – to eliminate later transfer over the PCI-E bus, which would become unnecessary bottleneck for our calculation.

The data passed to the kernel are array of points where we need to determine whether they are in obstacle or not, step size determining distance of points from each other on rectangular grid and then set of structures describing the obstacle transformation and visibility. The problem is specified as 2D, which allows filling the point structures with coordinates on the GPU, deducing them from kernel internal information IDs. The result of the calculation, array of points with field "usable" set to 0 or 1, is left in OpenCL buffer object on the GPU, to eliminate transfer over the bus.

B. *CollideRays Kernel*

The second kernel used performs the collision in given points. To modularize the code, two additional subroutines are used by the kernel: *aux_crossDistance* and *aux_rayObstacleDistance*. The first subroutine calculates the intersection between two lines. The second one decomposes the obstacles to set of lines and calls the first one to retrieve the intersection with whole object.

The arguments are array of validated points, information on obstacles and empty array to retrieve the precomputed rays. The design of OpenCL kernels is very flexible, allowing reusing the data already present on the GPU. Thanks to this the validated points are results of the FindPointsForCalculation kernel. The same could be done with the obstacle information, but here the expected problem appeared – excessive resource usage of kernels. Thanks to the nested function calls and dynamic branching, the kernel did not manage to process all the obstacles at once, resulting in *cl_out_of_resources* error.

This was solved by multipass execution of *CollideRays* kernels. In classic programming it would mean to pass all parameters over and over, with OpenCL we can update only the parameters which change. In this case, the only updated parameter will be smaller batch of objects to collide with. The rest of data stays on GPU and is continuously updated during the run of the passes.

The kernel code can be observed on Fig. 7, clearly demonstrating ease of use of OpenCL C.

```
__kernel void
CollideRays(__global const PointValidated2D* pointV,
            __global const Box2D* boxes,
            __global const int* boxCount,
            __global       PrecomputedPoint2D* pointC
           )
{
  int n = get_global_id(0);

  if (pointV[n].validated == 1) return;

  float2 origin;

  origin.x = pointV[n].x;
  origin.y = pointV[n].y;

  float i;
  float angle;
  float dist;

  #pragma unroll
  for(i = 0; i <= 359; i++ )
  {
    angle = radians(i);

    pointC[n].x        = origin.x;
    pointC[n].y        = origin.y;
    pointC[n].rayCount = 360;
    dist = aux_rayRectangleDistance(origin,
                                      angle,
                                    80000.0,
                                      boxes,
                                  *boxCount);

    if (pointC[n].pRay[convert_int(i)] == 0)
    {
      pointC[n].pRay[convert_int(i)] = dist;
    }
    else
    {
      if (dist <= pointC[n].pRay[convert_int(i)])
      {
       pointC[n].pRay[convert_int(i)] = dist;
      }
    }
  }
}
```

Fig. 7. Kernel code in OpenCL C

The ideal batch size had to be evaluated experimentally. Generally it was observed the amount of 200 obstacles per pass is acceptable, but it is recommended to fine tune this value on your target platform manually.

On most Nvidia cards the batch size seemed to have some kind of direct relationship with maximum parameter size, which is information reported by OpenCL run time API. But this was pure coincidence and this method cannot be used for batch size evaluation for real cases. In case of ATi hardware there was no such an observable coincidence.

During the adaptation of the method for the mobile robot it was observed that bigger batches of obstacles used (in amount not causing the mentioned resource problem) the better the performance was.

This indicates the transfer of data over the PCI-E bus still makes the difference, even in case of updating single, relatively small parameter.

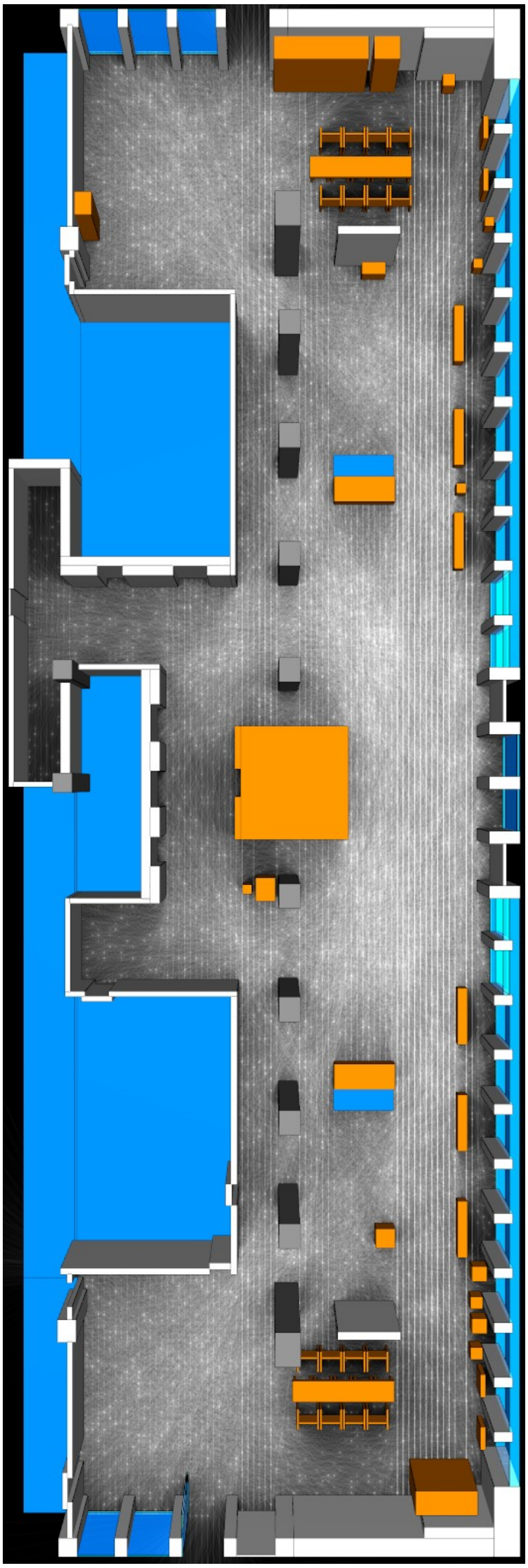


Fig. 8. Visualization of the precomputed rays

VII. RESULTS OF THE OPTIMIZATION

To compare the performance gain, we executed the original C# implementation, OpenCL CPU and OpenCL GPU implementation on AMD Sempron processor, which we observed to be approximately twice as fast as Intel Atom present in our mobile robot. It was very interesting to see the demonstrated OpenCL version on CPU with single core Sempron performs worse than our C# implementation, as seen on Fig. 9. To get the picture why we needed to optimize the precomputation, you can see the calculation time for 120 millimeters precision. It takes almost half an hour for Sempron CPU, which is still significantly faster comparing to CPU platform the robot was equipped with.

As much as the OpenCL CPU performance showed to be insufficient, the OpenCL GPU implementation demonstrated desired speed boost, which shortened the execution time from tens of minutes to tens of seconds, being viable option for practical use.

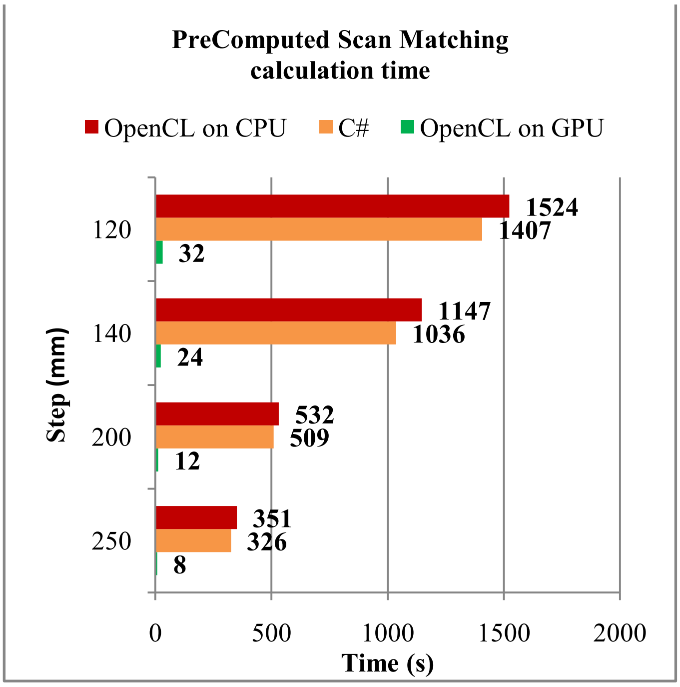


Fig. 9. Timing of the computation on GPU and two CPU implementations

The problem scales well on the GPGPU hardware. During the testing authors observed direct relationship between growing numbers of stream processors and shortening execution time. Even the low end models of graphic cards can provide highly competitive performance for data parallel calculations.

From theoretical point of view, using stronger graphic card could lead to unwanted power consumption problems. In reality this might not be an issue. With robot based on platform with high performance dedicated mobile GPGPU, such as *Nvidia GeForce GTX275M*, we can observe the execution time shortens to just few seconds even for high detail of precomputation. This results in very short peak in power consumption, which finally does not pose big problem for practical use of the technology.

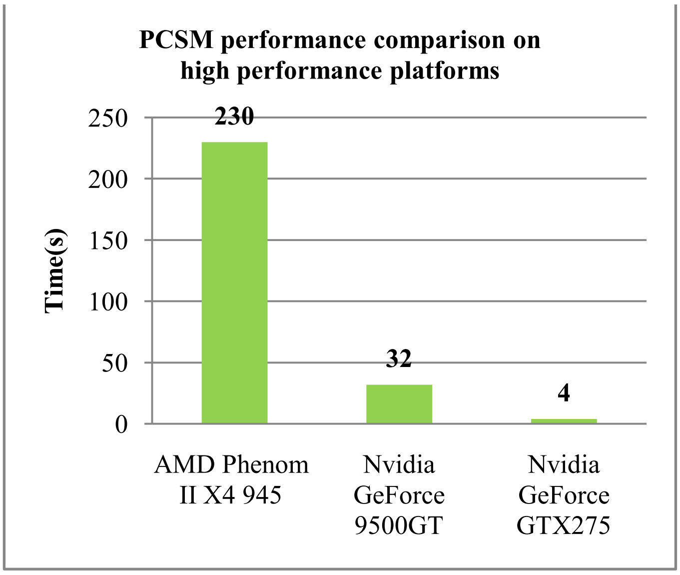


Fig. 10. Timing of the precomputation on high end CPU and two GPUs for 120 mm precision

The same observation cannot be confirmed for the case of using stronger CPU instead. The tested desktop high end 3GHz quad core processor from AMD proven to be still over 7 times slower than the low end GPGPU model implementation, while the power consumption, thanks to fully using power of all 4 cores, raised dramatically. When comparing the CPU performance to the high end GPU, the gap becomes even more significant as shows Fig. 10.

VIII. Conclusion

In the end, the presented GPGPU solution delivers significant acceleration of operation which was traditionally handled as offline task, while using conventional processor based approach. The optimization of the calculation has been realized thanks to using the latest OpenCL technology after very short adaptation time. The authors realize the solution could be further tweaked for even better performance.

The authors believe this successful application of GPGPU programming will encourage students and educators to focus more on the benefits of GPGPU computing for data parallel applications. The ease of use and high level syntax based programming make OpenCL an interesting technology worth the time of study. During the experiments OpenCL was used directly from multiple languages, including interpreted ThinBASIC[6]. This fact confirms that the students can use the technology from within language of their choice.

Solutions realized on OpenCL platform can be targeted to products of both major graphics card vendors, without writing any platform specific code. This makes OpenCL technology easily usable in the university environment, which very often provides highly heterogeneous hardware resources.

Building computationally intensive routines on top of GPGPU brings some significant cost savings when designing the mobile robot platforms as well. The performed tests demonstrated that even cheap GPU solutions can outperform costly modern CPUs of today at the fraction of the price.

IX. Acknowledgement

Published results were acquired with the support of the Ministry of Education, Youth and Sports of the Czech Republic, research plan MSM 0021630518 “Simulation modeling of mechatronic systems”.
While the system was designed for Bender 2[7] mobile robot platform, authors would also like to thank to Ing. Pavel Houška, PhD, Jakub Vodrážka, Jakub Drábek, Petr Liška, Jiří Michalčík, Charles Pegge and Kent Sarikaya for making the extensive testing possible on alternative hardware platforms.

[1] S. Věchet, J. Krejsa, “Potential-Based Scan Matching in Mobile Robot Localization”, in proceedings of Modelling and Optimization of Physical Systems. Gliwice, pp. 185-188, 2007

[2] Krejsa J., Věchet S., Hrbáček J., Schreiber P.: High Level Software Architecture for Autonomous Mobile Robot. In Recent Advances in Mechatronics. Berlin, Springer. 2009. p. 185 - 190. ISBN 978-3-642-05021-3.

[3] “ATI Stream Software Development Kit (SDK) v2.1” *Welcome to AMD Developer Central*. Advanced Micro Devices, Inc. n.d. Web. 30 May 2010. <http://developer.amd.com/gpu/ATIStreamSDK/Pages/default.aspx>

[4] “CUDA Programming Guide for CUDA Toolkit 3.0” *NVIDIA GPU Computing Developer Home Page*. Nvidia. n.d. Web. 30 May 2010. <http://developer.nvidia.com/object/gpucomputing.html>

[5] Khronos Group, “The OpenCL Specification”, Web. 30 May 2010. <http://www.khronos.org/registry/cl/specs/opencl-1.0.48.pdf, 2009>

[6] E. Olmi, R. Biancchi, ”Basic Programming Language :: thinBasic”. n.d. Web. 30 May 2010. <http://www.thinbasic.com/>

[7] J. Hrbáček, T. Ripel, J, Krejsa, “Ackermann mobile robot chassis with independent rear wheel drives”, EPE-PEMC 2010, Ohrid, Macedonia, in print.

Dynamic stability of a five-link biped robot

Nahla Mohamed Abd Alrahim Shannan, Prof. Dr. Shamsudin Haji Mohd Amin and Dr Zuwairie Ibrahim
Mechatronics and robotics: dept.
University Teknology Malaysia (UTM)
Malaysia

Abstract— **Keeping the dynamic stability during walking is one of the essential characteristics of regular bipedal walk, in the existence of unpowered DOF during SSP. To achieve the dynamic stability, there appears a decisive need to a robust controller to the robot movement. Here a new recurrent Neural Network is suggested as a controller for a five link biped robot, for tracking the desired angles trajectories for the legs of the robot.**

Keywords-biped robot; elman neural network; stability; control

I. Introduction

Walking is a fundamental feature of humanoid robots to achieve its goals, whatever these goals are, as the mobility of the robot is the main characteristic that categorizes it. For the biped robot there are three types of walkers (Marchese et al., 2001): static, dynamic and purely dynamic walkers. *Static walkers* are very slow walkers whose system's stability is completely described by the normal projection of the Centre of Gravity (COG), which depends on joints' position only, while *Dynamic walkers* have feet and actuated ankles. In this case the postural stability of dynamic walkers depends on joints' velocities and acceleration too. Dynamic walkers are potentially able to move in a static way, knowing that they have large feet and their motion is slow. Lastly, *purely dynamic walkers* are robots with no feet. In this case the contact area between the foot and the ground is reduced to a point, so that static walking is not possible. Hence normal human walking is a kind of dynamic bipedal locomotion, dynamic and purely dynamic walkers can simulate the human locomotion, which implies that, the five-link biped robot (as a purely dynamic walker) can emulate the human locomotion.

As mentioned by Vukobratovic and Juricic, 1969, the dynamic level collects the information on ambient and use these information for the purpose of control, through appropriate elements, to give the property of adaptability to the locomotion system. That means it is unavoidable to describe the gait of leg locomotion as a continuous process, based on the principles of analytical mechanics, using mathematical model. That's why the dynamic model of a biped robot is very important. On the other hand, practically every dynamic model has some degree of incorrectness and some errors in its parameters values, which cause errors in positioning and/or trajectory tracking which might cause system instability.

II. Dynamic Model

The main purpose of a modeling is to understand the system and to obtain a model that can be used to simulate and test the controllers. To model a biped robot there two main types of models: the kinematic model and the dynamic model. The kinematic model describes the motion of the biped robot without considering the exterior forces that cause the motion, while the dynamic model include all the exterior forces and is used to get the torques that act on each joint.

Many researchers contributed to the dynamic model of the biped robot, the main differences between these models are the number of links and degrees of freedom. Among these models, five-link biped robot has gained the attraction of many researchers (Furusho and Masubuchi, 1987, Tzafistas et al., 1996, Mu and Wu, 2004, Sadati and Hamed, 2007).

The biped under study consists of five links, one for the torso and two for each leg, those links are connected with four joints, two between the hip and each leg and two at the knees.

m_i: mass of link i.
l_i: length of link i.
r_i: is the distance from the centre of mass to the lower joint of link i.
θ_i: the angle between the vertical and link i.
I_i: moment of inertia of link i, with respect to the axis that passes through the COM of link i and perpendicular to the sagittal plane.
(x_e,y_e): the coordination of the swing leg tip.
(x_b,y_b): the coordination of the point of support leg.

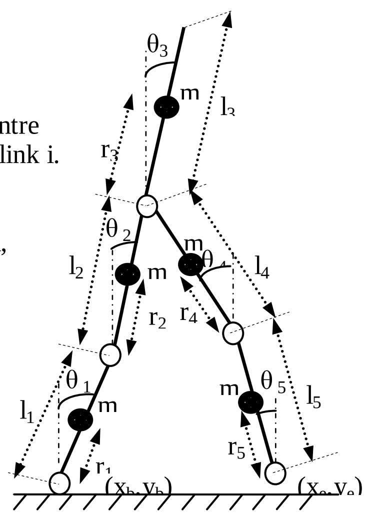


Figure 1

According to the kinematic relationship between the links of the robot that shown in figure (1), the position and velocity of the free end of the swing leg can be defined as follows:

$$x_e = \sum_{i=1}^{2}(l_i \sin\theta_i) + \sum_{i=4}^{5}(l_i \sin\theta_i) + x_b$$

$$y_e = \sum_{i=1}^{2}(l_i \cos\theta_i) + \sum_{i=4}^{5}(l_i \cos\theta_i) + y_b$$

$$\dot{x}_e = \sum_{i=1}^{2}(l_i\,\dot{\theta}_i \cos\theta_i) + \sum_{i=4}^{5}(l_i\dot{\theta}_i \cos\theta_i) + \dot{x}_b$$

$$\dot{y}_e = -\sum_{i=1}^{2}(l_i\,\dot{\theta}_i \sin\theta_i) + \sum_{i=4}^{5}(l_i\dot{\theta}_i \sin\theta_i) + \dot{y}_b$$

While the position and velocity of the centre of mass of each link is shown in the following form:

$$x_{ci} = \sum_{j=1}^{i-1}(bl_j \sin\theta_j) + \sum_{j=4}^{i}(l_j \sin\theta_j) + ar_i \sin\theta_i + x_b$$

$$y_{ci} = \sum_{j=1}^{i-1}(bl_j \cos\theta_j) + a\sum_{j=4}^{i}(l_j \cos\theta_j) + r_i \cos\theta_i + y_b$$

$$\dot{x}_{ci} = \sum_{j=1}^{i-1}(bl_j\dot{\theta}_j \cos\theta_j) + \sum_{j=4}^{i}(l_j\dot{\theta}_j \cos\theta_j) + ar_i\dot{\theta}_i \cos\theta_i + \dot{x}_b$$

$$\dot{y}_{ci} = \sum_{j=1}^{i-1}(bl_j\,\dot{\theta}_j \sin\theta_j) + a\sum_{j=4}^{i}(l_j\dot{\theta}_j \sin\theta_j) + r_i\dot{\theta}_i \sin\theta_i + \dot{y}_b$$

$$a = \begin{cases} 1 & ; i < 4 \\ -1 & ; i \geq 4 \end{cases} \quad ; \quad b = \begin{cases} 1 & ; j < 3 \\ 0 & ; j \geq 3 \end{cases}$$

The dynamic equation of the five-link biped robot is derived using Lagrange equations, where:

$$L = K - P$$

Where K is the kinetic energy, P is the potential energy and L is Lagrange coefficient.
The potential energy (P) is given by:

$$P = \sum_{i=1}^{5} P_i \qquad \text{with} \qquad P_i = m_i g y_{ci}$$

Where $g = 9.8\ m/s^2$ is the gravitational acceleration.
The kinetic energy is given by:

$$k = \sum_{i=1}^{5} K_i \qquad \text{with} \qquad K_i = \frac{1}{2} m_i\, {v_{ci}}^2 + \frac{1}{2} I_i {\theta_i}^2$$

Now the potential and kinetic energy will be substituted in Lagrange formula to solve the dynamic equations for the SSP:

$$\frac{d}{dt}\left\{\frac{\partial L}{\partial \dot{q}_i}\right\} - \frac{\partial L}{\partial q_i} = T_i$$

This can be expressed in the following form:

$$\frac{d}{dt}\left\{\frac{\partial K}{\partial \dot{q}_i}\right\} - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} = T_i$$

Rearranging the equation above, the standard form of the equation of motion can be writing in the following form:

$$D(\theta)\ddot{\theta} + H(\theta, \dot{\theta})\,\dot{\theta} + G(\theta) = T_\theta$$

Where $D(\theta)$ is 5*5 inertia matrix, $H(\theta)$ 5*5 centrifugal and coriolis' terms matrix, $G(\theta)$ is a 5*1 gravity matrix and $T_\theta, \theta, \dot{\theta}, \ddot{\theta}$ are 5*1 vectors of torque, generalized coordinates, velocities and accelerations respectively.

To control this dynamic model we should get a reference values to follow during the walking process. To get these values we have to drive the trajectories of the biped robot.

III. TRAJECTORIES PLANNING

Designing reference trajectories for the joints of the biped robot is one of the crucial aspects of motion control for these robots. Arbitrary planning of these trajectories may lead to instability due to tipping over during walking gait; it may also cause high energy consumption of the tracking actuators, that's why driving these trajectories became a very sensitive and important issue.

Mu and Wu (2004) have introduced a method for synthesizing the gait of a planar five-link biped robot walking on level ground for both the single support phase (SSP) and double support phase (DSP). They use time polynomial to produce limb and hip trajectories, which have the advantage of simplifying the problem by dividing the biped into three subsystems. They determine the joint angle profiles for a full gait cycle including the SSP and the DSP. The constraint functions and gait parameters are chosen to generate a repeatable gait.

First to derive the ankle trajectory a third and fourth order time polynomial have been chosen to represent the x_a and y_a coordinate of the ankle respectively.

$$X_a = \begin{cases} x_a(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \\ y_a(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \end{cases} \qquad (1)$$

Where $0 \leq t \leq T_s$; T_s the time period for SSP.

To solve the above equation the following ten constraints equations are going to be used:

$$y_a(0) = 0\,, \qquad y_a(T_s) = 0$$

$$x_a(T_m) = S_m\,, \qquad y_a(T_m) = H_m\,, \qquad \dot{y}_a(T_m) = 0$$

$$x_a(0) = -\frac{S_L}{2}, \qquad x_a(T_S) = S_L/2\,, \qquad \dot{x}_a(0) = 0$$

$$\dot{y}_a(0) = 0\,, \qquad \dot{x}_a(T_S) = 0\,, \qquad \dot{y}_a(T_S) = 0$$

Where: S_L is step length, T_s step period for the SSP, H_m is the maximum clearance of the swing limb, its location is S_m and T_m is the time corresponding to the maximum clearance.

Secondly, the hip trajectories are characterized using the following equation:

$$X_{hs} = \begin{cases} x_{hs}(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3, \\ y_{hs}(t) = y_h(t) \end{cases} \qquad (2)$$

Assuming that the height of the hips is kept constant during the gait and with the following constraints, equation (2) can be solved.

$$x_{hs}(0) = -S_{S0} \ , \qquad \dot{x}_{hs}(0) = V_{h1} \ ,$$
$$\dot{x}_{hs}(T_S) = V_{h2} \ , \qquad x_{hs}(T_S) = -S_{S0} + S_L$$

Where S_{S0} is the position of the hip at the beginning of the SSP, S_L is the step length, and V_{h1} is is the hip velocity at the beginning of each step, V_{h2} is the hip velocity at the end of the SSP.

The real challenge in designing these trajectories is in choosing appropriate hip velocities during the gait which is mainly a try and error process.

Lastly, the joint angle profiles can be determined uniquely, with the hip and swing limb tip trajectories already designed and the biped kinematic model, and can be described by the following equations:

$$\begin{cases} \theta_1(t) = \sin^{-1}\left(\dfrac{A_1C_1 + B_1\sqrt{A_1^2 + B_1^2 - C_1^2}}{A_1^2 + B_1^2}\right) \\ \theta_2(t) = \theta_1(t) + \sin^{-1}\left(\dfrac{A_1\cos(\theta_1(t)) - B_1\sin(\theta_1(t))}{l_2}\right) \\ \theta_3(t) = 0 \\ \theta_4(t) = \sin^{-1}\left(\dfrac{A_4C_4 + B_4\sqrt{A_4^2 + B_4^2 - C_4^2}}{A_4^2 + B_4^2}\right) \\ \theta_5(t) = \theta_4(t) + \sin^{-1}\left(\dfrac{A_4\cos(\theta_4(t)) - B_4\sin(\theta_4(t))}{l_5}\right) \end{cases}$$

Where

$$A_1 = x_{hs}(t), \qquad B_1 = y_{hs}(t), \qquad C_1 = \frac{A_1^2 + B_1^2 + l_1^2 - l_2^2}{2l_1},$$
$$A_4 = x_{as}(t) - x_{hs}(t), \qquad B_4 = y_{hs}(t) - y_{as}(t),$$
$$C_4 = \frac{A_4^2 + B_4^2 + l_4^2 - l_5^2}{2l_4}$$

IV. A NEW RECURRENT NEURAL NETWORK

Artificial Neural Network (ANN) provides effective techniques for system identification and control of nonlinear systems. As ANN have gained fame in solving problems with high difficulty as it have the ability to approximate nonlinear mappings and model complex system behavior; without prior knowledge of the system structure or parameters; to achieve accurate control through training.

Mainly there are two types of neural networks: feed-forward and recurrent neural network. Feed-forward neural network have no feedback elements, which mean that the output is calculated directly from the input. On the other hand, the recurrent neural networks have feedback connections which makes the output depends not only on the current input to the network, but also on the current (or previous) outputs or states of the network. Because of this, recurrent neural networks are considered more powerful than feed-forward networks, and have important uses in control applications.

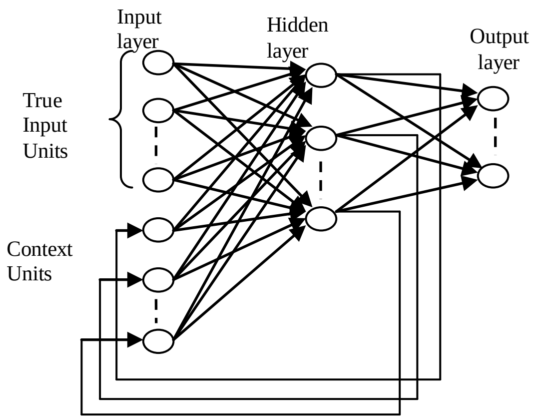


Figure 2: Elman Neural Network

Fig. 2 shows the structure of Elman neural network which is one of the early networks in this field. The main problem with it is that its training and speed of convergence is usually very slow.

In order to enhance the performance, a new recurrent neural network is introduced, which feedback the output of the network to both the hidden and output layer. The output of the neural network is used as a feedback signal due to its importance as the value to be adjusted to reach the desired value according to the specified input.

To analyze our network, a simplified model of the network which consists of one hidden layer and output layer, with zero activation value for the hidden layer and output layer is shown in Fig. 3. The weights of the feed-forward connections may vary, while the weights of the feed-back connections are fixed to reflect the previous situation of the network. The weights of the forward and backward paths are as shown in Fig. 3.

The equations which describe the relation between the input and output of the network are shown below:

$$y(k) = w_2x(k) + w_4r_1(k) + w_6r_2(k) \tag{3}$$
$$x(k) = w_1u(k) + w_3r_1(k) + w_5r_2(k) \tag{4}$$
$$r_1(k) = y(k-1) + \alpha r_1(k-1) \tag{5}$$
$$r_2(k) = y(k-1) + \beta r_2(k-1) \tag{6}$$

Using Z transforms on equations (1) to (4) above, we get:

$$y(z) = w_2x(z) + w_4r_1(z) + w_6r_2(z) \tag{7}$$
$$x(z) = w_1u(z) + w_3r_1(z) + w_5r_2(z) \tag{8}$$
$$Z\, r_1(k) = y(z) + \alpha r_1(z) \tag{9}$$
$$Z\, r_2(k) = y(z) + \beta r_2(z) \tag{10}$$

Equations (5) to (8) give the following transfer function:

$$y(z)\left[Z^2 - (\alpha + \beta + w_2\, w_3 + w_2\, w_5 + w_4 + w_6)Z + [\alpha\beta + (w_2\, w_3 + w_4)\beta + (w_2\, w_5 + w_6)\alpha]\right] = u(z)[w_1\, w_2\, Z^2 - w_1\, w_2(\alpha+\beta)Z + \alpha\,\beta] \tag{11}$$

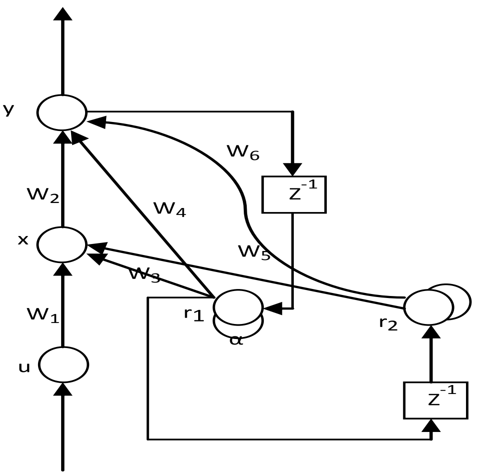


Figure (3)

Equation (11) above implies that:

$$\mathrm{y(k)} = (\alpha + \beta + \mathrm{w_2\,w_3} + \mathrm{w_2\,w_5} + \mathrm{w_4} + \mathrm{w_6})\mathrm{y(k-1)} - [\alpha\beta + (\mathrm{w_2\,w_3} + \mathrm{w_4})\beta + (\mathrm{w_2\,w_5} + \mathrm{w_6})\alpha]\mathrm{y(k-2)} + \mathrm{w_1\,w_2\,u(k)} - \mathrm{w_1\,w_2}(\alpha+\beta)\mathrm{u(k-1)} + \alpha\,\beta\mathrm{u(k-2)} \quad (12)$$

If we compare equation (10) with the discrete form of the PID controller in equation (11); we can see that they have similar form; that implies that if we equate the terms of both equations, which shows our new RNN is similar in behavior to a PID controller.

$$y(\mathrm{k}) = \frac{2T+\Delta T}{T+\Delta T}y(k-1) - \frac{T}{T+\Delta T}y(k-2) + \frac{(K_p+K_iT)\Delta T + K_D + K_pT + K_i\Delta T^2}{T+\Delta T}u(k) - \frac{(K_p+K_iT)\Delta T + 2K_D + 2K_pT + K_i\Delta T^2}{T+\Delta T}u(k-1) + \frac{K_D+K_pT}{T+\Delta T}u(k-2)$$
(11)

Where:

$$(\alpha + \beta + \mathrm{w_2\,w_3} + \mathrm{w_2\,w_5} + \mathrm{w_4} + \mathrm{w_6}) = \frac{2T+\Delta T}{T+\Delta T}$$

$$[\alpha\beta + (\mathrm{w_2\,w_3} + \mathrm{w_4})\beta + (\mathrm{w_2\,w_5} + \mathrm{w_6})\alpha] = \frac{T}{T+\Delta T}$$

$$\mathrm{w_1\,w_2} = \frac{\left(K_p + K_iT\right)\Delta T + K_D + K_pT + K_i\Delta T^2}{T+\Delta T}$$

$$\mathrm{w_1\,w_2}(\alpha+\beta) = \frac{\left(K_p + K_iT\right)\Delta T + 2K_D + 2K_pT + K_i\Delta T^2}{T+\Delta T}$$

$$\alpha\,\beta = \frac{K_D + K_pT}{T+\Delta T}$$

Where T the inertia time and ΔT is the sampling time.

V. Conclusion

From the above we can see that our new recurrent neural network give the same behavior as PID controller in addition the $\alpha\ and\ \beta$ factors can be used to increase the derivative and proportional parts in the PID controller while $\mathrm{w_1}$ and $\mathrm{w_2}$ can be used to reduce the integral part to give nearly a PD controller behavior.

References

[1] J. Furusho and M. Masubuchi, "A Theoretically Motivated Reduced Order Model for the Control of Dynamic Biped Locomotion," Control, vol. 109, 1987, pp. 155-163.

[2] X. Mu and Q. Wu, "Sagittal Gait Synthesis for a Five-Link Biped Robot," *American Control conference*, 2004, pp. 4004-4009.

[3] S. Marchese, G. Muscato, and G. Virk, "Dynamically Stable Trajectory Synthesis for a biped robot during the single-support phase," *International Conference on Advanced Intelligent Mechatronics*, IEEE/ASME, 2001, pp. 953-958.

[4] S. Tzafestas, M. Raibert, and C. Tzafestas, "Robust Sliding-mode Control Applied to a 5-Link Biped Robot," *Journal of lntelligent and Robotic Systems*, 1996, pp. 67-133.

[5] N. Sadati and K. Hamed, "Neural Controller for a 5-link planar biped robot," *International Conference on Robot & Human Interactive Communication*, IEEE, Korea: 2007, pp. 980-985.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Walking robot modelling aspects

Marcin Piątek* and Dariusz Marchewka*

*AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Automatics
Mickiewicza 30 av. 30-059 Kraków, Poland
e-mail: marcin.piatek@agh.edu.pl, dariusz.marchewka@agh.edu.pl

***Abstract*—The paper describes a walking robot leg modelling process. A six-legged walking robot — the so-called hexapod is concerned. Each leg consists of three links driven by Hitec HS-475HB servo motors. The proposed model is used for testing leg kinematics and dynamics of the robot gait. One can examine a number of walking algorithms. The model includes: the identified description of the servo motors, a full state observer (position and velocity), forward kinematics of the position and the velocity, inverse kinematics of the position and the leg movement visualisation. The model structure is explained and depicted. The simulation results are shown in a graphical form. The examples of model applications are applied and described. The advantages and disadvantages of the model are listed. Eventual experiments and applications are announced.**

***Index Terms*—Walking robot, modelling, simulation, identification.**

I. INTRODUCTION

Scientist and engineers have been interested in the walking robots since a long time. There are benefits of walking legs in a contrary to rolling on wheels or moving on tracks. Walking legs can move on almost every surface like rock debris, ocean floor, surface of the other planets and such a simple obstacle like the stairs. However, one can find walking legs drawbacks: a slow motion, high power and a lot of identical elements to be used.

As far as motion is concerned the walking robots resemble animals or more precisely insects.

One of the simplest construction is hexapod — a robot equipped with the six legs, usually three on each side. Thanks to the leg numbers we can easily design walking algorithms in this way that the whole construction will be stable in every walk phase (in the contrary to the four-legged and the two-legged robots).

The six-legged walking robot application discussed in this paper was designed and built to model the walking patterns observed in the six-legged insects (Figure 1). As an example the *Blatta Orientalis* was chosen, it is also known as a cockroach. This species are very well examined by biologists and there is a lot of paper devoted to insect walking patterns [8].

The insect legs motion is controlled at a low level of the neural structure. It means that a high neural structure level is free from signals devoted to walking algorithms. These algorithms have a very simple form. Usually correspond to synchronously repetitive actions.

Figure 1. The hexapod — six-legged walking robot.

In the case of the six-legged insects we consider three characteristic ways of walking:

- the insect moves only one leg in a time instance, the other five legs support the insect body
- the insect moves two legs in a time instance and the other four legs support the insect body
- the insect moves three legs in a time instance, the other three legs support the insect body

These three ways of moving legs will be considered as the three different modes of the insect walk and apply also to the six-legged walking robots. Their analysis will help to implement efficient algorithms of walk for a hexapod.

II. THE ALGORITHMS OF THE WALK

The plane surface without any imperfections is considered. In Figure 2 we can see the scheme of the robot leg. Every leg consists of the three links driven by the servo motors. The numbered legs are shown in Figure 3. Figure 4 shows the way how the legs are moved in the each walking mode. The full cycle consist of six stages in the mode a), three stages in the mode b) and only two stages in the mode c). The analysis of the insect motion helps to develop walking algorithms for the hexapod. One can conclude:

- at the same velocity of the single leg move the robot can achieve three different velocities by changing the walk mode
- the robot can move forward or backward due to the applied sign of the control signal

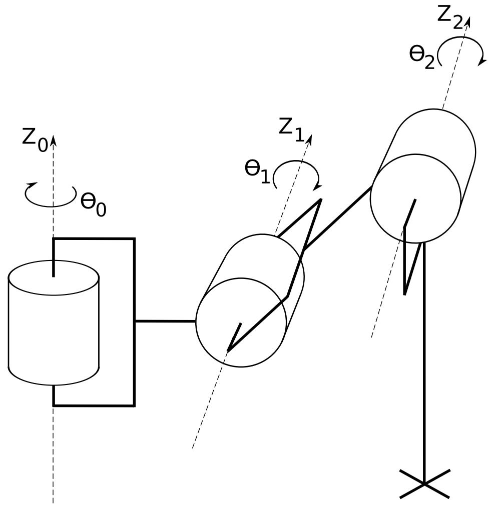


Figure 2. The scheme of the robots leg connections.

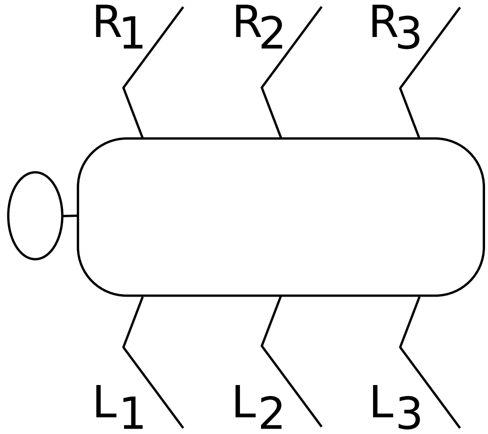


Figure 3. The schematic structure of the six-legged insects.

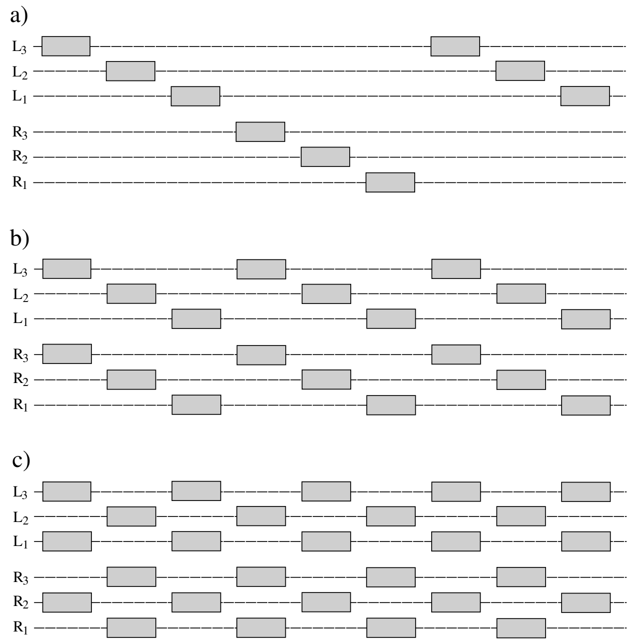


Figure 4. The scheme presenting three basic modes of the six-legged insects walk.

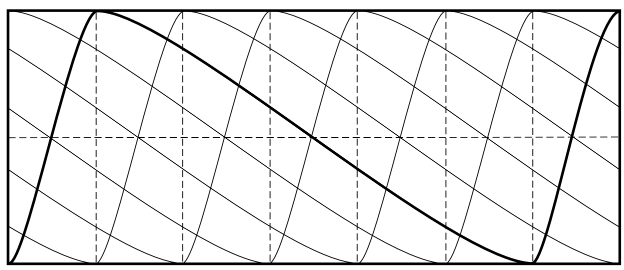


Figure 5. The first DOF control signals in the a) mode of the walk.

- the higher number of legs raised, the faster the robot can move but the legs on which the robot is leaning are more burdened
- when a leg (or legs) are moving forward, the supporting legs have to move backward with the velocity given by the formula

$$v_b = v_f \frac{n}{6-n} \tag{1}$$

  where v_f is the velocity of the legs moving forward and n is the number of the legs moving forward.

The considered velocities relate to velocity of the robot leg tip. The last observation corresponds to the diagrams presented in Figures 5, 6 and 7. They show positions of the first degree of freedom (DOF) of the robot legs vs. time in each mode of the walk.

Figure 5 presents the legs motion in the mode a). It shows that the legs move forward one by one. The position of the one leg is drawn by the bold line to make the diagram more clear. In this mode every leg moves separately and we can easily distinguish two phases: the fast phase — when the leg moves forward and the slow phase — when the leg moves backward. According to equation 1 the leg velocity when it is moving backward is five times smaller then the velocity when it is moving forward.

Figure 6 presents the legs motion in the mode b). In this mode the legs move being coupled in such a way that in the same time instance two of them move forward and the other four supports the robot. The coupled legs move synchronously. It is possible to see only one leg for each couple. The combination presented in Figure 4 is not the only one possible. The legs could be arranged in couples in few different ways. The only requirement is that construction has to be stable in every phase of the walk. The velocity of the legs when moving backward is two times smaller then in the case when moving forward.

Figure 7 presents the legs motion in the mode c). In this mode legs are arranged by three. When three legs move forward the other three move backward. This is the fastest possible way of the walk of the hexapod. To make the construction stable in every phase of the walk there is only one possible way to arrange legs in three. The middle leg from one side has to move synchronously to the front and the back leg from the other side. The velocity of the moving backward legs is the

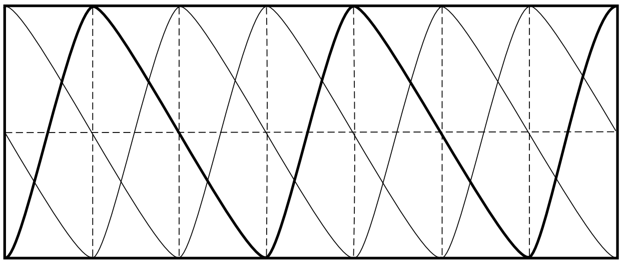


Figure 6. The first DOF control signals in the b) mode of the walk.

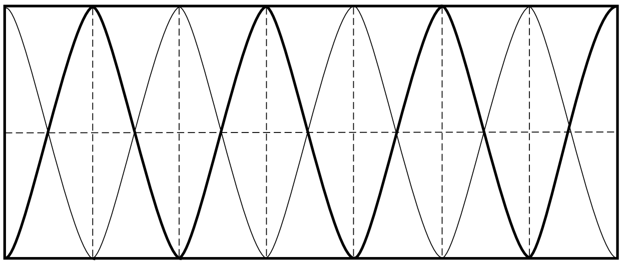


Figure 7. The first DOF control signals in the c) mode of the walk.

same as the legs that are moving forward.
It is important to notice that absolute value of velocity of the robot is equal to the absolute value of velocity of the moving backwards legs.

III. The one leg move

The previous diagram shows only the position vs. time of the first DOF of the legs. To simulate and eventually build the whole walking robot based on the presented algorithm we have to consider motion of the two remaining DOF. It is possible to find it out from the position of the leg tip by the inverse kinematics procedure. Figure 8 presents the trajectory of the robot leg tip during the walk. The leg tip moves only in the two dimensions: the robot front–rear and up–down. The trajectory has a shape of the ellipse where its one diameter is the step length and the other diameter is the step height. In the upper part of the trajectory the leg moves with the velocity equal to v_f and in its lower part the velocity is equal to v_b.

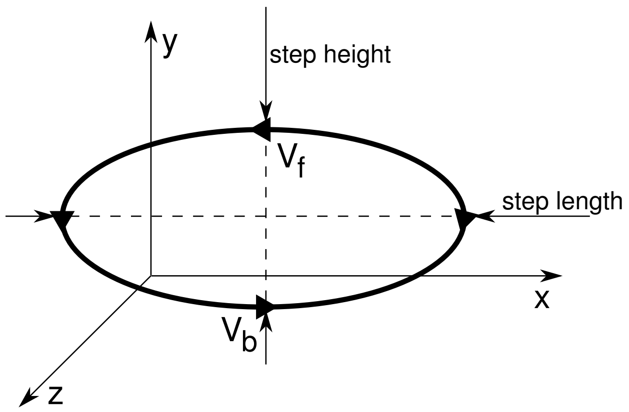


Figure 8. The diagram with the trajectory of the robot leg tip

The conclusion from the diagram is that the position of the second and the third DOF of the robots leg is directly dependent on the position and sign of the velocity of the first DOF. To find the relationship we have to derive the inverse kinematics of the robot leg thus the model of this system was created in the Simulink environment [7] [3].

IV. The robot leg model

The model contains:

- the identified dynamics of the Hitec HS-475HB servo motor
- the state observer to calculate the servo motor velocity and to filter the measured position signal
- the forward kinematics of the robot leg
- the inverse kinematics of the robot leg.

Through the Simulink modelling tool it was possible to identify, simulate and verify the correctness of the created model.
The Hitec HS-475HB servo motor is built with the DC motor, the mechanical gear, the potentiometer (to measure position signal) and the proportional controller in a feedback loop. The model structure is presented on Figure 9. Table I presents names and values of the all parameters. The values were obtained during the identification process [5] [4]. The initial values of the servo motors positions equals $90°$ for the first DOF, $30°$ for the second and $120°$ for the third one. The maximal values for the servo motors positions are equal $\pm 60°$ with respect to the initial positions. The High Gain Observer algorithm was used to estimate the servo motor velocity and to filter the position signal [6]. Figure 10 presents the observer model. Figure 11 presents the architecture of the leg model. In the Figure we can distinguish separate components like: the three servo motors models with the observers, the forward kinematics and the inverse kinematics. The forward kinematics problem of the position and the velocity was solved according to rules described in [7]. The coordinate axes has been designated according to Denavit-Hartenberg convention. Figure 12 presents the leg diagram with the assigned axes. During the modelling procedure of this part some problems were already identified for the future implementation. The most important one is computational complexity of the derived formula. It uses the trigonometric functions (sine and cosine) which are real challenge for the small processors usually embedded in the mobile robots. To simplify the equation entry

Parameter Name	Symbol	Value
DC motor inertia	T_i	0.0181
proportional controller coefficient	k	20.8279
upper control signal constraint	$\overline{S}$	220.9851
lower control signal constraint	$\underline{S}$	-220.854

Table I
THE SERVO MOTOR MODEL PARAMETERS

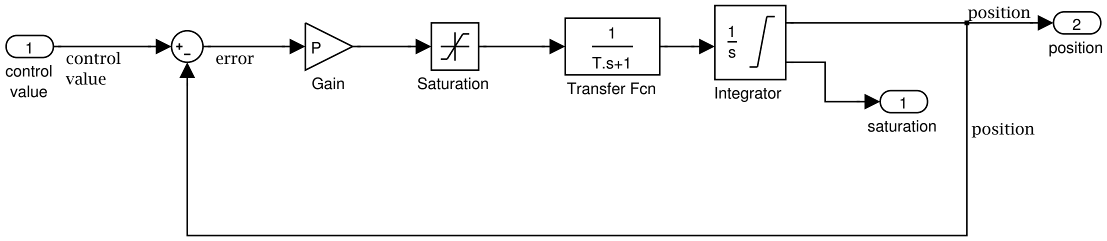


Figure 9. The model of the servo motor

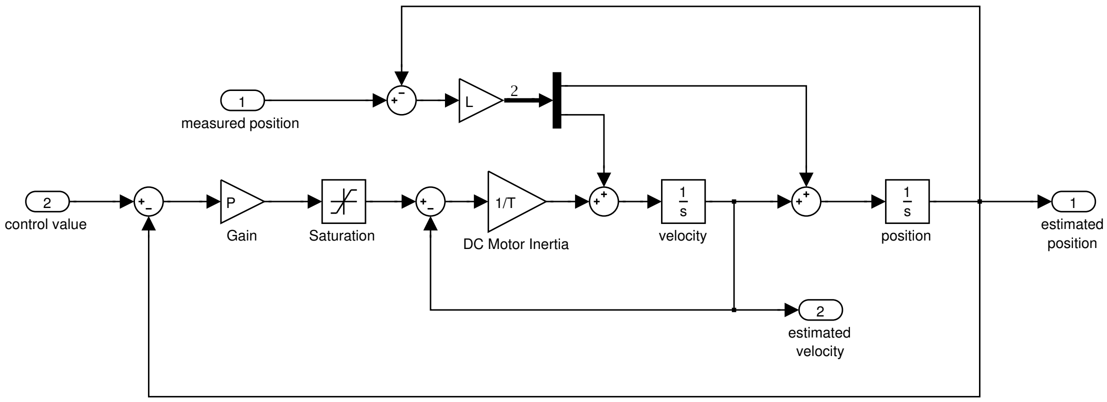


Figure 10. The model of the observer

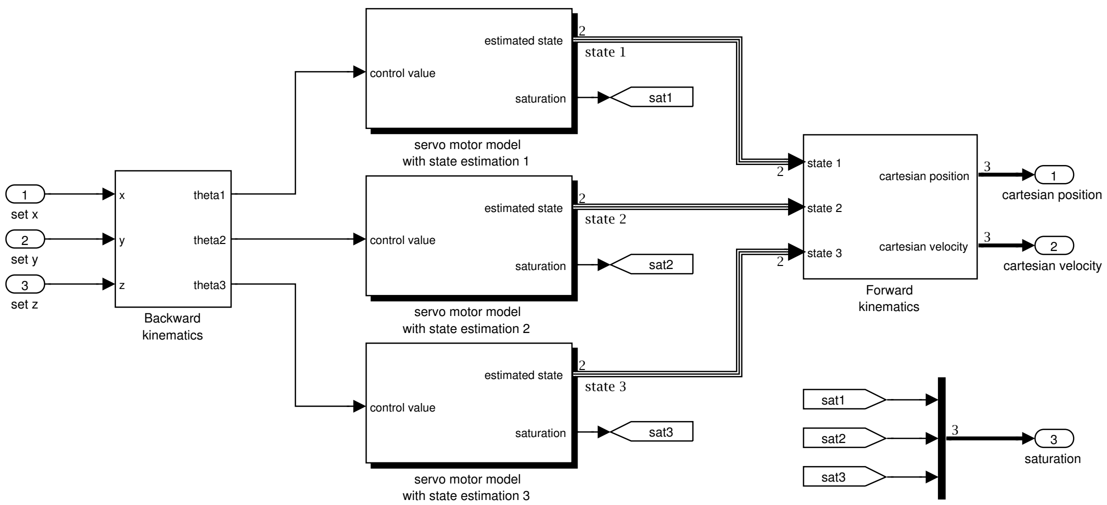


Figure 11. The structure of the leg model

some definitions were made:

$$\begin{aligned} s_1 &= \sin(\theta_1) \\ s_2 &= \sin(\theta_2) \\ s_3 &= \sin(\theta_3) \\ \\ c_1 &= \cos(\theta_1) \\ c_2 &= \cos(\theta_2) \\ c_3 &= \cos(\theta_3) \end{aligned} \tag{2}$$

The equation 3 solves the forward kinematics of the position.

$$\begin{aligned} x &= l_3c_1c_2c_3 + l_3c_1s_2s_3 + l_2c_1c_2 + l_1c_1 \\ y &= l_3s_1c_2c_3 + l_3s_1s_2s_3 + l_2s_1c_2 + l_1s_1 \\ z &= l_3s_2c_3 - l_3c_2s_3 + a + l_2s_2 \end{aligned} \tag{3}$$

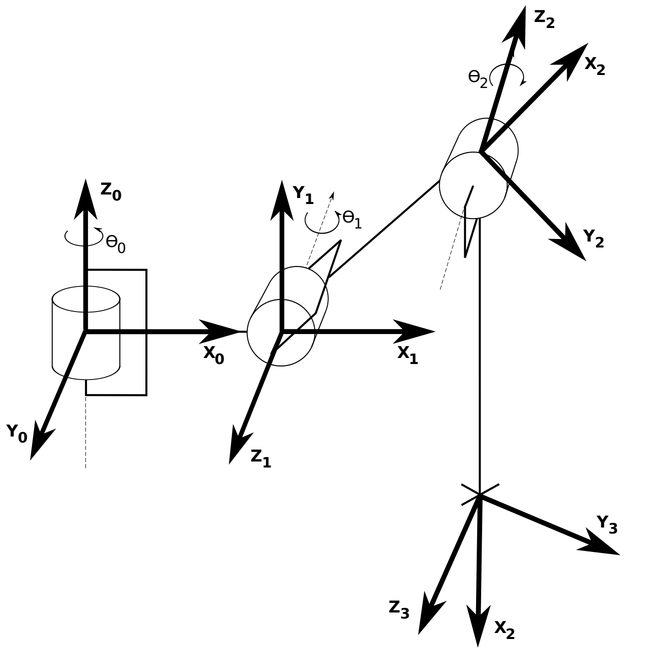


Figure 12. The leg structure with coordinate axes designated in accordance with Denavit-Hartenberg convention

The equation 4 solves the forward kinematics of the velocity.

$$
\begin{aligned}
\dot{x} &= (-l_3s_1c_2c_3 - l_3s_1s_2s_3 - l_2s_1c_2 - l_1s_1)\cdot\dot{\phi}_1 \\
&\quad -c_1\cdot(l_3s_2c_3 - l_3c_2s_3 + l_2s2)\cdot\dot{\phi}_2 \\
&\quad c_1\cdot(l_3s_2c_3 - l_3c_2s_3)\cdot\dot{\phi}_3 \\
\dot{y} &= (l_3c_1c_2c_3 + l_3c_1s_2s_3 + l_2c_1c_2 + l_1c_1)\cdot\dot{\phi}_1 \\
&\quad -s_1\cdot(l_3s_2c_3 - l_3c_2s_3 + l_2s_2)\cdot\dot{\phi}_2 \\
&\quad s_1\cdot(l_3s_2c_3 - l_3c_2s_3)\cdot\dot{\phi}_3 \\
\dot{z} &= (s_1 * (l_3s_1c_2c_3 + l_3s_1s_2s_3 + l_2s_1c_2) \\
&\quad +c_1 * (l_3c_1c_2c_3 + l_3c_1s_2s_3 + l_2c_1c_2))\cdot\dot{\phi}_2 \\
&\quad (-s_1\cdot(l_3s_1c_2c_3 + l_3s_1s_2s_3) \\
&\quad -c_1 * (l_3c_1c_2c_3 + l_3c_1s_2s_3))\cdot\dot{\phi}_3
\end{aligned}
\tag{4}
$$

The parameters l_1, l_2 and l_3 are the length of the separate leg components and the parameter a is a vertical displacement between the first link and the center of the main coordinates system. The inverse kinematics algorithm is less demanding but the same computational problem will appear in future implementations. The pseudo code listed below presents how the inverse kinematics problem is solved:

$$
\begin{aligned}
&\theta_1 = \arctan\left(\frac{y}{x}\right) \\
&\mathrm{if}(\theta_1 < 0) \\
&\qquad \theta_1 = \theta_1 + \pi \\
&\mathrm{end} \\
&r = \sqrt{x^2 + y^2} \\
&c = (z-a)^2 + (r-l_1)^2 \\
&\theta_3 = \arccos\left(\frac{c - l_2^2 - l_3^2}{2l_2l_3}\right) \\
&\alpha = \arccos\left(\frac{l_3^2 - l_2^2 - c}{-2l_2\sqrt{c}}\right) \\
&\beta = \arctan\left(\frac{z-a}{r-l_1}\right) \\
&\mathrm{if}((r - l_1) \geq 0) \\
&\qquad \theta_2 = \alpha + \beta \\
&\mathrm{else} \\
&\qquad \theta_2 = \alpha - (\pi - \beta) \\
&\mathrm{end}
\end{aligned}
$$

Presented algorithm was also extended with procedures preventing errors like division by zero and similar.

V. The model usage

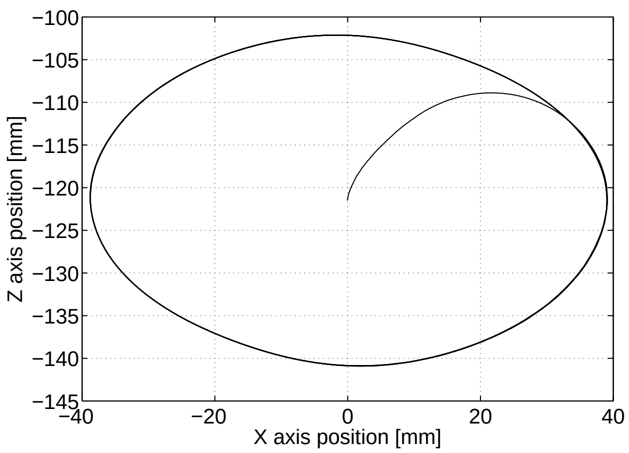


Figure 13. The model of the observer

The model described in the previous section has been and will be used for a few different purposes. One of them is simulation of the observer algorithm, the forward kinematics of the position, the forward kinematics of the velocity and the inverse kinematics of the position. The simulation allows to validate algorithm correctness and verify against different inputs values. The inputs values that can cause the algorithm errors are the most interesting. The errors early identification lets possibly avoid the software reimplementation and even hardware damage in future.

Figures 13 and 14 present the result of the model simulation

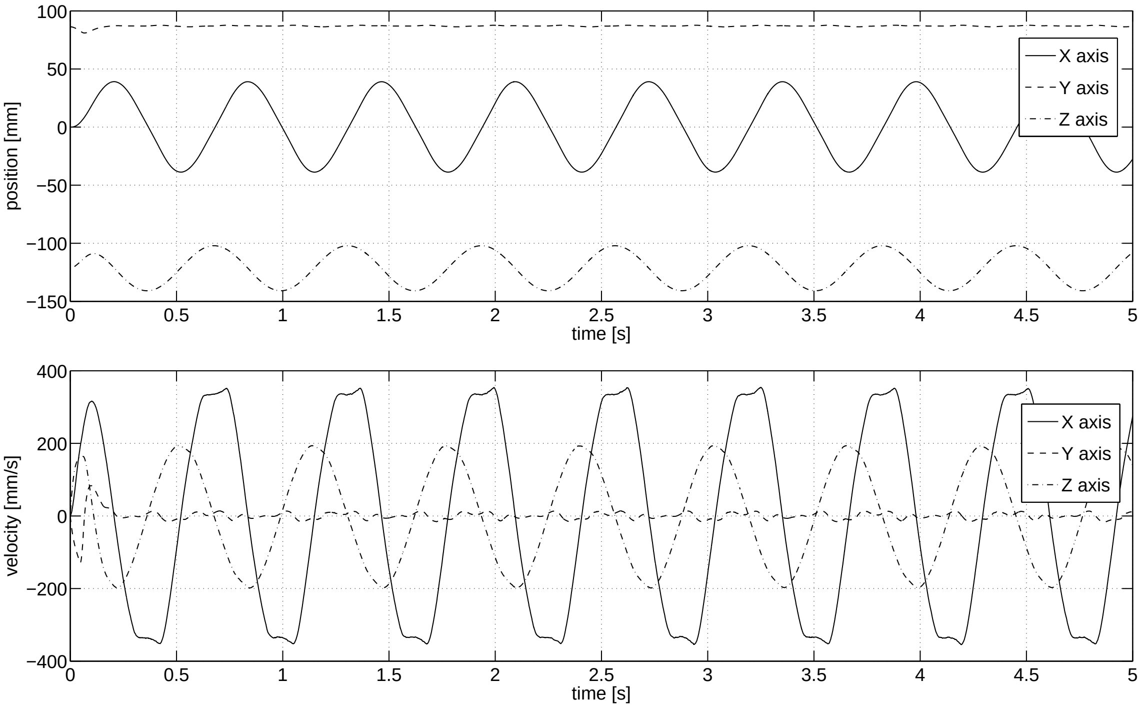


Figure 14. The model of the observer

where the input value to the model was a function generating ellipse trajactory on X and Z axes (like during walking process).

The model can also be used for the code generation, what is one of the Simulink features. The C code could be generated from the prepared model or the parts of the model and then deployed directly to the walking robot control system.

The model is also useful during the leg control algorithm design. As mentioned in the previous sections, the robot leg moves along the ellipse with the given velocity. The model helps to choose the ellipse diameters and the velocities in the different parts of trajectory. It is also useful for testing other control strategies.

As the forward and the inverse kinematics algorithms are computationaly complex it would be almost impossible to implement them on the smaller processors, especially not equipped with the Floating Point Unit (FPU) like ARM7 and ARM9 [1] [2]. As the trajectory of the leg tip is periodic in time, the small processor could be equipped with the Look-Up Table of the servo motor positions over the whole trajectory. The control algorithm could iterate over the Look-Up Table with varying velocity to determine the next servo motors positions. The values retrieved from the array can also be scaled with an appropriate ratio to determine different step sizes. The look-up table can be determined with the designed model during the simulation of the designed control algorithms.

The described applications are going to be implemented in subsequent experiments and projects carried out by the students and the stuff of the AGH-UST.

REFERENCES

[1] Z. Fortuna, B. Macukow, and J. Wąsowski. *Metody numeryczne*. Podręczniki akademickie. Elektronika. Informatyka. Telekomunikacja. Wydawnictwa Naukowo-Techniczne, 7 edition, 2006.

[2] D. Marchewka and M. Piątek. Stosowanie systemów wbudowanych do sterowania robotami mobilnymi. *Półrocznik AGH Automatyka*, 12(2):257–266, 2008.

[3] B. Mrozek and Z. Mrozek. *MATLAB i Simulink. Poradnik użytkownika.* Helion, 2 edition, 2004.

[4] M. Piątek. Identification of the servo motor used in the walking robot. *Półrocznik AGH Automatyka*, 14(1), 2010.

[5] M. Piątek and P. Piątek. Development environment for the walking robot application. In *Computer Methods and Systems 2009*, pages 295–299, Nov. 2009.

[6] M. Piątek and A. Tutaj. Angular velocity estimation in the six-legged walking robot application. In *PPEEm 2009: Fundamental problems of power electronics electromechanics and mechatronics*, Dec. 2009.

[7] M. W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. Wydawnictwa Naukowo-Techniczne, 1997.

[8] T. Zielińska. *Maszyny kroczące*. Wydawnictwa Komunikacji i Łączności, Warszawa, 2005.

Wheeled mobile robot modeling aspects

Dariusz Marchewka* and Marcin Piątek*
*AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Automatics
Mickiewicza 30 av. 30-059 Kraków, Poland
e-mail:dariusz.marchewka@agh.edu.pl, marcin.piatek@agh.edu.pl

***Abstract*—A young designer of the wheeled mobile robot has to answer a wide range of questions before making a final decision to build a prototype. The basic problem is to define an appropriate drive and type of construction. At this stage, the young designers usually duplicate ready-made solutions, however, in general uses an inappropriate drive. The robot behaves correctly only after several failed designs. Unfortunately, the construction of several versions of the robot takes a long time and involves significant costs. The authors suggest how to help and accelerate the robot design process. Following these suggestions the young designer can develop his knowledge how to model and simulate in the MATLAB/Simulink environment. The first part of the article contains a basis for mathematical modeling of the wheeled mobile robots. As an example, a complete mathematical model of the selected robot type is developed. The second part presents methods of verification and validation the prepared model. After selecting the appropriate drive parameters (e.g. maximum torque, maximum speed, etc.) user is able to perform a simulation of the designed robot. In addition, the effects of simulation can be seen in the visualization and a designer is able to assess if the design of robot meets the established requirements.**

Keywords-mobile robot, mathematical modeling, simulation

I. INTRODUCTION

Robotics is a very popular field of knowledge. Since many years human beings have tried to build an autonomous robot. Thanks to the development of a new technologies dreams of human beings become realistic. Robotics attracts ordinary people not only those educated in this domain. There are more and more complex robots constructed. They are equipped with built-in computers therefore they become autonomous system. A perfect example is the Lego Mindstorms NXT [7]. With the ingenuity of engineers from the MIT Media Lab, the school-age children can build their own robot designs and they can write the first control programs with using a simple graphical software. However, for a young designers that is not enough. They use more advanced technologies and build more complex robots. The most popular designs are the wheeled mobile robots. Usually, robots of this type are used during competitions [1] (e.g. a micromouse, a line-follower, a sumo fight etc.). Designers of the mobile robots typically create their projects with the use of arbitrary elements: e.g. DC motors from damaged audio-video devices, encoders from old type computer mouse, optical sensors from broken printers etc. The parameters of these elements usually are unknown, but it seems to be not important to inexperienced developers. They simply wish to win the competition. Motor's parameters (max. torque, max. current, nominal rpm. etc.) are very significant for a good design. We can select the appropriate drive for at least two ways:

- Perform an experiments with different motors
- Prepare a mathematical model of a mobile robot and perform simulation with different parameters of the motor

The first method requires the preparation of several versions of the same robot. It takes a lot of time and involves costs, but is easy. The designer does not need to know the mathematical model. Second method requires knowledge and tools for modeling and simulation. This method is dedicated to people who want to improve their knowledge about the modeling of mobile robots. Principles of modeling of mobile robots will be presented in the second part of the paper.

II. PRINCIPLES OF MODELING OF MOBILE ROBOTS

A. Kinematics and simulation of two-wheeled mobile robot

The most popular design among the mobile robots is two-wheeled robot with differential drive [1], [6], [3]. The configuration of this robot is shown in Fig.1

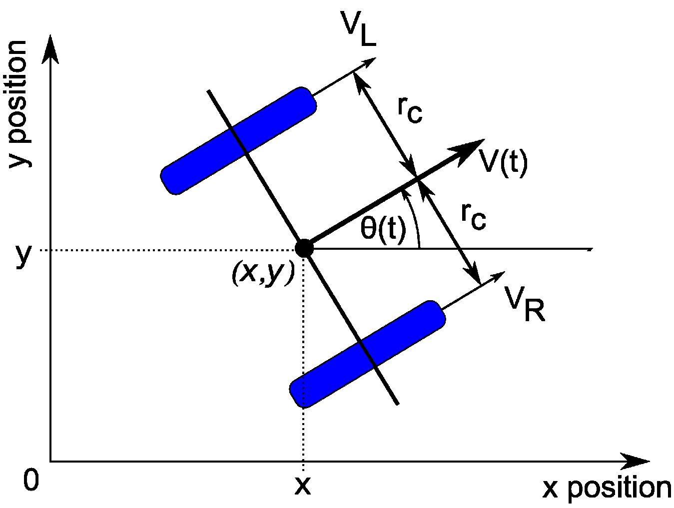


Figure 1. Configuration of the mobile robot.

Two independent DC motors are the actuators of the left and right wheels and one free wheel caster is used to keep the platform stable. This configuration uses independent linear velocities: v_L for the left wheel and v_R for the right wheel to move to a desired point *(x,y)* and desired orientation θ. The linear left(right) wheel velocity $v_L(v_R)$ is directly proportional to the angular velocity of the left(right) wheel (1), where r_w is the wheel radius.

$$\begin{aligned} v_L &= r_w \cdot \omega_L \\ v_R &= r_w \cdot \omega_R \end{aligned} \tag{1}$$

The relation between the linear $v(t)$ and angular $\omega(t)$ speeds of the platform depend on the linear velocity of left and right wheel (2). Parameter r_c is a robot chassis radius. It was assumed that the wheels move on the plane without slip.

$$v(t) = \frac{v_L(t) + v_R(t)}{2}, \qquad \omega(t) = \frac{v_R(t) - v_L(t)}{r_c} \tag{2}$$

Kinematics [3] of two-wheeled robot is as follows (3):

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) \cdot v(t) \\ \sin(\theta(t)) \cdot v(t) \\ \omega(t) \end{bmatrix} \tag{3}$$

One of the best tools for modeling and simulation is MATLAB/Simulink package [7]. The Simulink model of robot kinematics (3) with velocity relation (2) is presented in Fig.2.

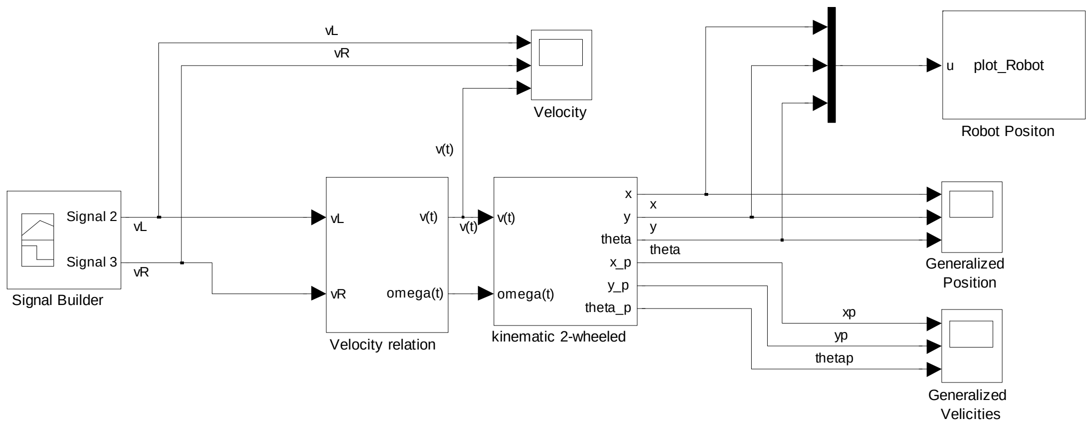


Figure 2. Simulink model of the robot kinematics.

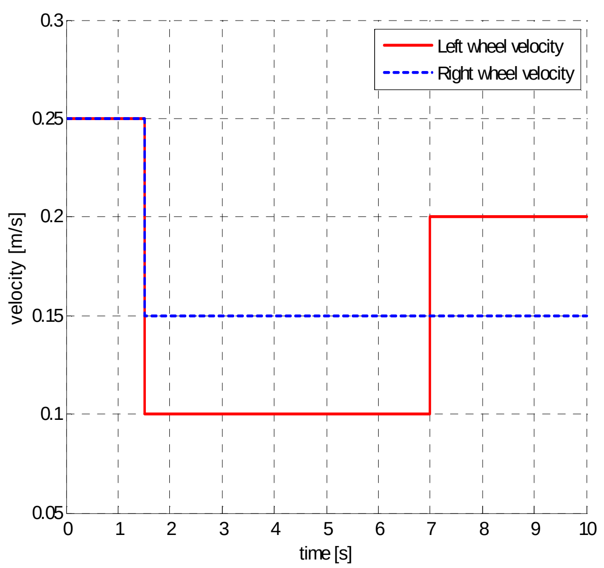


Figure 3. Desired velocity

The ‘plot_Robot’ block is dedicated to a simple visualisation of robot movement.

To simulate kinematic model we should prepare a desired velocity signal. A simple signal is shown in Fig.3 and is divided into three parts:

- velocities v_L and v_R are equal – robot should move forward,
- v_L is smaller then v_R – robot should turn left,
- v_R is smaller then v_L – robot should turn right.

Initial parameters are as follows: robot chassis radius r_c=0.15m, initial position (x_{ini},y_{ini}) = (-0.5[m],-0.5[m]) and initial orientation $\theta = 0$ [rad]. The behavior of the simulated robot is shown in Fig. 4. In the first part of simulation the orientation θ is equal to zero. This means that the robot moves forward. The *x* position changes linearly and *y* position remains unchanged. In the second part, when v_L is smaller then v_R, the orientation θ increases linearly. This means, in accordance with the coordinate system (Fig.1), that the robot turns left. In the last part, when v_R is smaller then v_L, the orientation θ decreases linearly – robot turns right.

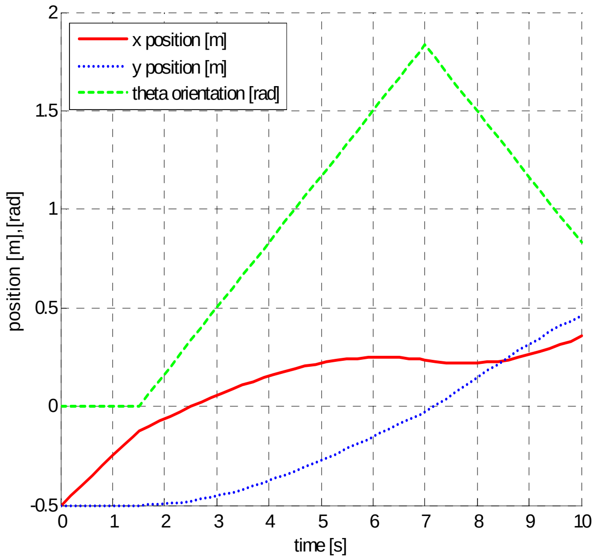


Figure 4. Mobile robot (x,y) position and orientation (θ)

When we have the robot *(x,y)* position on the surface, and his orientation θ, we can prepare the simple visualization of the movement of the mobile robot (Fig.5).

The robot is represented by a rectangle. The shortest edges are the wheels and the center of rectangle is the generalized robot position. At each step of simulation a new position and orientation of the robot are calculated and plotted. Visualization of the position of the robot can be helpful in interpreting the charts. Observing the motion of the robot at visualization, one can quickly determine the accuracy of movement.

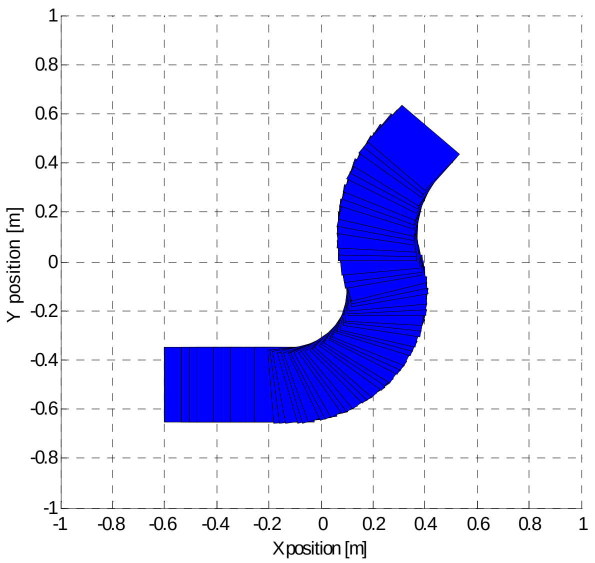


Figure 5. Visualization of robot position on the 2D surface

B. *Modeling of the DC motor*

The mobile robots that take part in competitions are powered by DC motors. The model of DC motor [5], [1] one can describe as a linear second order system. The motor torque *M(t)*[Nm] is related to the armature current *i(t)*[A], by a constant *K* [Nm/A] (4).

$$M(t) = K \cdot i(t) \tag{4}$$

The back electromotive force V_{EMF} is proportional to the angular velocity of DC motor (5)

$$V_{EMF}(t) = e(t) = K \cdot \omega(t) = K \cdot \frac{d\theta}{dt} \tag{5}$$

Based on the Newton's law combined with the Kirchoff's law we can write the equation (6), where *u(t)* is the input voltage, *R* is a resistance and *L* is a inductance of the armature, *J* is a moment of inertia of the rotor, M_F is a damping ratio of the mechanical system.

$$\begin{aligned} M(t) &= K \cdot i(t) = M_F \cdot \omega(t) + J \cdot \frac{d\omega(t)}{dt} \\ u(t) &= e(t) + R \cdot i(t) + L\frac{di(t)}{dt} \end{aligned} \tag{6}$$

The transfer function of system (6) from the input voltage *u(t)* to angular velocity $\omega(t)$ is as follows (7):

$$G_v(s) = \frac{\omega(s)}{u(s)} = \frac{K}{(L \cdot s + R)(J \cdot s + M_F) + K^2} \tag{7}$$

The Simulink model of the DC motor is shown in Fig.6.

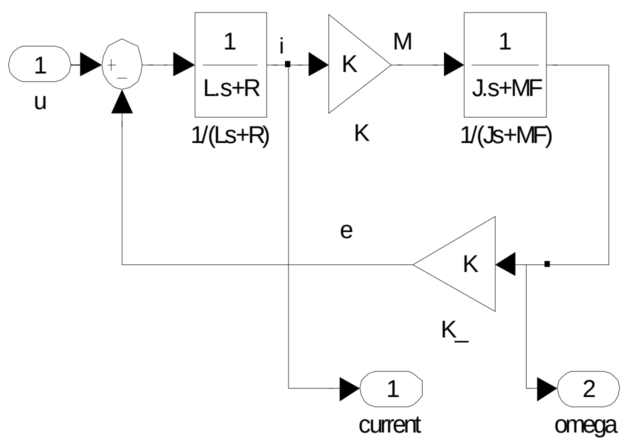


Figure 6. Simulink model of the DC motor

III. SIMULATION OF THE MOBILE ROBOT WITH DC MOTORS

When we have a model of a mobile robot and model of a DC motor we can simulate a behavior of the whole system. First we can define parameters of the DC motor:

R=1[Ω], L=0.5[H], K=0.05[Nm/A], J=0.0025[kgm^2]

The Simulink model of the whole system is shown in Fig.7.

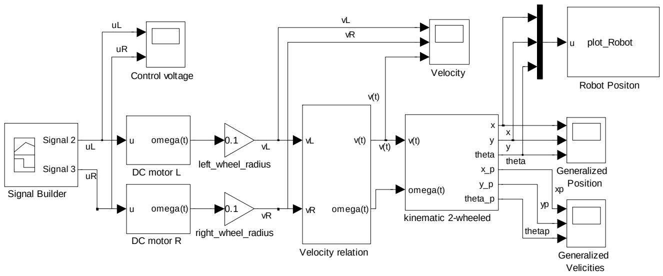


Figure 7. The Simulink model of the mobile robot with the DC motors

The voltage control signals and the corresponding robot position and orientation are presented in Fig.8. As in the previous simulation experiment that signals are divided into three parts, but now simulation results taken into consideration the dynamics of the motors. The orientation of the robot changes smoothly after implementation of the motor dynamics.

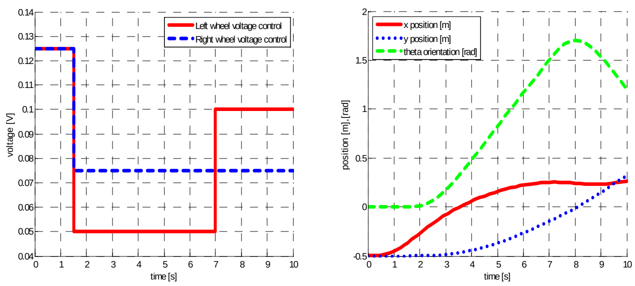


Figure 8. The voltage control signals and the position and orientation of the mobile robot.

The model is prepared to perform the simulation with many different DC motors parametres. For example, Fig.9 presents visualization of the robots with two types of the DC motors. The first type has the torque motor constant K=0.05[Nm/A]. The second type has the constant K equal to 0.1[Nm/A].

a) b)

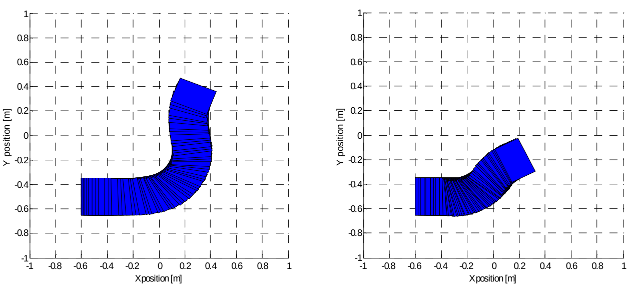


Figure 9. Visualization of the robot position for different motor parameters: a) K=0.05[Nm/A], b) K=0.1[Nm/A]

The control signals and time of the experiments are the same as in the previous simulation. In the Fig.9a we can observe that the robot moves faster than the robot showed in Fig.9b. The final robot position and orientation are different, too. In this way we can quickly determine whether the DC motor corresponds with our requirements.

IV. CONCLUSIONS

The paper presents some modeling aspects of the wheeled mobile robots. Inexperienced designers (e.g. students) usually make robots and they does not consider whether used components will be sufficient. The study shows that with the appropriate software for modeling and simulation one can test the operation of any robot. In this case, the software MATLAB/Simulink with the right tools and libraries is used. With this solution the designer can expand their knowledge of robotics, and saves time and money. The presented modeling aspects should be taken as an introduction to more advanced analysis of the behavior of mobile robots in the future. Also, the described aspects are going to be implemented as a modeling introduction of the students robotic workshop.

REFERENCES

[1] W.P. Aung, "Analysis on Modeling and Simulink of DC Motor and its Driving System Used for Wheeled Mobile Robot", Proceedings Of World Academy Of Science, Engineering And Technology, vol. 26, pp. 299-306, December 2007

[2] T. Bräunl," Research relevance of mobile robot competitions," IEEE Robotics and Automation Magazine, vol. 6, no. 4, pp. 32–37 , December 1999 (6)

[3] G. Dudek, M.Jenkin, Computational principles of mobile robotics. Cambridge, Cambridge University Press 2000

[4] M. J. Giergiel, Z. Henzel, W. Żylski, Modelowanie i sterowanie mobilnych robotów kołowych. Warszawa, Wydawnictwo Naukowe PWN, 2002.

[5] M. Pauluk," Projektowanie algorytmów sterujących w układach sterowanych napięciowo i prądowo", Miesięcznik Naukowo-Techniczny PAR – Pomiary, Automatyka, Robotyka, vol. 12, pp. 37-45, Warszawa 2005

[6] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, R. Muszyński, Manipulatory i roboty mobilne. Warszawa, Akademicka Oficyna Wydawnicza PLJ, 2000

[7] http://www.mathworks.com

[8] http://mindstorms.lego.com

Bayesian Filters in Practice

Low-cost mobile robot position tracking using bearing only beacons

Jiří Krejsa
Centre of Mechatronics
Institute of Thermomechanics AS CR, v.v.i.
Brno, Czech Republic
krejsa@fme.vutbr.cz

Stanislav Věchet
Faculty of Mechanical Engineering
Brno University of Technology
Brno, Czech Republic
vechet.s@fme.vutbr.cz

***Abstract*—Bayesian filters represent the most commonly used tool for state estimation not only in mobile robotics. The filters are widely used in sensor data fusion and robot localization problems. The paper describes in detail our experiences with the filters in robot localization using bearing only beacons. Bearing only beacons are easy to implement, therefore can be realized by the students and relatively complex task of Bayesian filtering can be explained using real data. Both simulation and practical results with Extended Kalman filter and Unscented Kalman filter are given, taking into consideration not only the precision of obtained estimate, but also its robustness against the noise and memory and computational requirements that must be considered when computational resources are limited.**

Mobile robot localization, bearing only beacons, Bayesian filters

I. Introduction

Bayesian filters are commonly used tools whenever certain quantity can not be expressed as a single value / vector, but estimate is used instead, taking into consideration the probabilistic nature of the quantity. Such situation often appears in data fusion, when data measured by the sensors are affected with certain level of noise. As an example the fusion of odometry readings and compass measurements can be given. Typical task that requires the probabilistic approach in quantities description is the position tracking problem (local localization). Bayesian filters are widely used to address this problem even to the extent of global localization and simultaneous localization and mapping problem, when the ability to model the desired quantity (robot position) with multimodal probability distribution is essential [1], [2].

The task of position tracking requires the fusion of robot motion model and sensor data about the environment. This task is essential in mobile robot navigation, and students in the field are often introduced to the state estimation problem when examining the problem.

Robot motion can use either the velocity information from the controllers, or the odometry information read from IRC sensors on robot wheels. Environment data are usually in the form of landmarks, e.g. visual landmarks extracted from the images acquired by the camera [3], [4]. Such approach has the advantage of relatively low cost, as the prices of image acquiring systems are low and still dropping. However the computational cost of image processing is rather high. Our goal was to develop a position tracking method usable in cases when computational resources are rather limited (that is often the case in simple low cost educational robots). The method, described further in more detail uses infrared beacons placed on known locations and receiver located on the robot, capable of detecting the relative angle between the beacons and the robot, together with the information about the beacon identification, thus solving the mapping problem. Robot position is estimated by Bayesian filter that combines the motion model with the receiver measurement.

Low cost of bearing only beacons predestinates its use in education of robotics. Robot that uses beacons based localization can be built by the students themselves in reasonable time, as the further described method does not require odometry readings. Understanding the Bayes filters is essential for almost any task in robotics whenever coping with uncertainty. However although the literature on this field is vast, e.g. [5], it is not easy to find a text directly usable by the students to write down a real robot application. Hopefully this paper can serve as a guide not only to understand the use of the filters in beacon based localization, but also to address the practical problems often neglected in theoretical literature.

The paper is organized in following way. Chapter II gives detailed overview of the task in question and describes Bayesian filters used. Chapter III shows the results of simulations while chapter IV shows the results obtained by the experimental robot and chapter V brings the conclusions.

II. Materials and Methods

A. The task in detail

The problem to be solved is to determine the position of the robot on the plane (2D problem) given the motion model of the robot, beacons relative angles measurement, beacons positions and initial position. In other words, we know the control actions of the robot and absolute fixed position of the beacons and we measure the relative angles of beacons with respect to the robot. From these data we determine the position of the robot.

The position is given by x and y coordinate of the robot $\left[x^R, y^R\right]$ and its heading angle φ^R with respect to fixed global coordinate system. As the position changes in time, such position in time step k is further denoted as the state vector $\mathbf{x}_k$:

$$\mathbf{x}_k = \begin{bmatrix} x_k^R & y_k^R & \varphi_k^R \end{bmatrix}^T \tag{1}$$

The state changes as the robot moves. State change is evoked by actions: the translational and rotational velocities of the robot u^t and u^r . Motion model gives the relation between the actions and state vector change.

There are N beacons available in known fixed locations given by the x and y coordinates in global coordinate system. See the overview in Fig. 1.

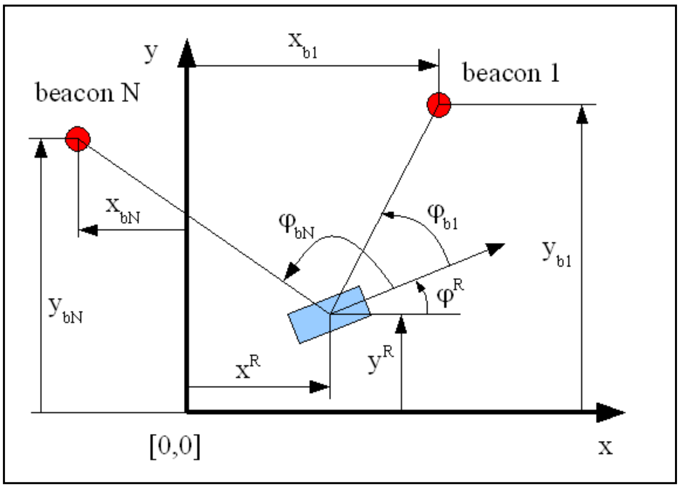


Figure 1. The state of the robot, beacons positions and beacons measurement

Due to the imperfections of the motion model and noise in the measurement, the state can not be represented only as the vector of values and probabilistic approach must be considered. As the problem in hand is of the position tracking, the multimodal probability density can be avoided and simple Gaussian approximation is used. The description of the underlying model can therefore be defined as

$$\begin{aligned} \mathbf{x}_{k+1} &= f\left(\mathbf{x}_k, \mathbf{u}_k, k\right) + \mathbf{v}_k \\ \mathbf{y}_k &= h\left(\mathbf{x}_k, k\right) + \mathbf{w}_k \end{aligned} \tag{2}$$

where $\mathbf{x}_k$ is n dimensional vector of states (robot position), $\mathbf{u}_k$ is action vector (velocities given by the controller), $\mathbf{v}_k$ is white Gaussian process noise (representing the imperfections of motion model) with zero mean and covariance matrix $\mathbf{V}_k$, $\mathbf{y}_k$ is system output, $\mathbf{w}_k$ is white Gaussian measurement noise with zero mean and covariance matrix $\mathbf{W}_k$ and f and h are continuously differentiable nonlinear functions. While in general those functions can depend on time step index k, if further examples such dependency is not used.

The state transition function f defines how the state (robot position) changes when action is applied:

$$\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{u}_k\right) + \mathbf{v}_k = \begin{bmatrix} \cos\varphi_k^R u_k^t + x_k^R \\ \sin\varphi_k^R u_k^t + y_k^R \\ u_k^r + \varphi_k^R \end{bmatrix} + \mathbf{v}_k \tag{3}$$

Regarding the measurements, as there are N beacons generally available (however, there is no guarantee that all the beacons are detected), their positions are (see Fig. 1): applied:

$$\mathbf{x}_{bi} = \left[x_{bi}, y_{bi}\right], i = 1, 2, ..., N \tag{4}$$

For a single beacon the output equation is

$$\mathbf{y}_{1k} = \left[h_1\left(\mathbf{x}_k, \mathbf{x}_{b1}\right)\right] + \left[\mathbf{w}_{1k}\right] \tag{5}$$

where

$$h_1\left(\mathbf{x}_k, \mathbf{x}_{b1}\right) = \left[\operatorname{atan2}\left(y_k^R - y_{b1}, x_k^R - x_{b1}\right) - \varphi_k^R\right] \tag{6}$$

For N beacons the equations are simply added one by one depending on what beacon was measured. In general:

$$\mathbf{y}_k = \begin{bmatrix} h_1\left(\mathbf{x}_k, \mathbf{x}_{m1}\right) \\ h_2\left(\mathbf{x}_k, \mathbf{x}_{m2}\right) \\ \vdots \\ h_{n_m}\left(\mathbf{x}_k, \mathbf{x}_{mn_m}\right) \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{1k} \\ \mathbf{w}_{2k} \\ \vdots \\ \mathbf{w}_{n_m k} \end{bmatrix} \tag{7}$$

Now with the task properly defined, our goal is to produce the state estimate with mean $\hat{\mathbf{x}}_k$ and covariance matrix $\mathbf{P}_k$. This can be done by Bayesian filters.

B. Extended Kalman filter

There is a number of Bayesian filters suitable for the problem in hand. They differ by the representation of the estimate. The Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) both represent the estimate by random vector with Gaussian distribution. Both filters work in usual predictor/corrector manner. The difference is in the way the filters use for linearization of nonlinear functions f and h from (2). EKF uses the Taylor Expansion and therefore requires partial derivatives of the functions, while UKF uses linearization via the Unscented Transform. Let's first take a look at EKF.

The estimator works in two stages. First the motion model is applied (estimate changes when action is performed) and new position estimate is predicted. In second stage the measurement is taken into account (beacons relative angles) and position estimate is corrected. Therefore the input is the current estimate (at the beginning of the whole process this estimate is equal to initial estimate of robot position), the action and the measurement. Necessary relations are given by following equations:

Prediction (state estimate change when action is applied:

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= f\left(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, k\right)\\ \mathbf{P}_{k+1|k} &= \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{V}_k\end{aligned} \tag{7}$$

Update (correction of the estimate using measured data):

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\tilde{\mathbf{y}}_{k+1}\\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\end{aligned} \tag{8}$$

where:

$$\mathbf{F}_k = \frac{\partial f}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}}$$

$$\tilde{\mathbf{y}}_{k+1} = \mathbf{y}_{k+1} - h\left(\hat{\mathbf{x}}_{k+1|k}, k+1\right)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T\mathbf{S}_{k+1}^{-1}$$

$$\mathbf{S}_{k+1} = \mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + \mathbf{W}_{k+1}$$

$$\mathbf{H}_{k+1} = \frac{\partial h}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}}$$

EKF requires the partial derivatives of f and h functions. For the motion model, the partial derivatives forming Jacobian $\mathbf{F}_k$ are given by:

$$\mathbf{F}_k = \frac{\partial f}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} = \begin{bmatrix} 1 & 0 & -\sin\hat{\varphi}_{k|k}^R u_k^t \\ 0 & 1 & \cos\hat{\varphi}_{k|k}^R u_k^t \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

And Jacobian for a single beacon is (the matrix is transposed to fit within the text):

$$\mathbf{H}_{k+1,1} = \begin{bmatrix} \dfrac{1}{1+\left(\dfrac{y_{b1}-\hat{y}_{k+1|k}^R}{x_{b1}-\hat{x}_{k+1|k}^R}\right)^2}\dfrac{y_{b1}-\hat{y}_{k+1|k}^R}{\left(x_{b1}-\hat{x}_{k+1|k}^R\right)^2} \\ -\dfrac{1}{1+\left(\dfrac{y_{b1}-\hat{y}_{k+1|k}^R}{x_{b1}-\hat{x}_{k+1|k}^R}\right)^2}\dfrac{1}{x_{b1}-\hat{x}_{k+1|k}^R} \\ -1 \end{bmatrix}^T \tag{10}$$

Jacobains for more beacons are added the same way as the output equations (7).

C. Unscented Kalman filter

The Taylor series expansion applied by the EKF is not the only way to linearize the transformation of Gaussians. The Unscented Kalman Filter uses so-called unscented transform. The principle is following: UKF deterministically extracts so-called sigma points from the Gaussian and passes them through nonlinear function. Resulting mean and covariances are then extracted from transformed sigma points. This method might remind of the Monte Carlo method or Particle filter that uses samples from the distribution as distribution representation. The difference is that the sigma points are chosen deterministically, not randomly.

The generation of sigma points $\mathcal{X}$ for n-dimensional random vector $\mathbf{x}$ in UKF is given by the following rule:

$$\begin{aligned}\mathcal{X}_0 &= E(\mathbf{x})\\ \mathcal{X}_i &= E(\mathbf{x}) + \left(\sqrt{(n+\kappa)\mathbf{P}_\mathbf{x}}\right)\\ \mathcal{X}_{n+i} &= E(\mathbf{x}) - \left(\sqrt{(n+\kappa)\mathbf{P}_\mathbf{x}}\right)\end{aligned} \tag{11}$$

Where $E(\mathbf{x})$ is the mean of random vector $\mathbf{x}$ and $\mathbf{P}_\mathbf{x}$ is the covariance matrix. Parameter κ determines how far the sigma points are spread around the mean. n-dimensional random vector produces *2n+1* sigma points.

Sigma points are transformed by the nonlinear function (in our case there two such functions, f and h).

$$\mathcal{Y}_i = f(\mathcal{X}_i) \tag{12}$$

Each sigma point is accompanied by the weight used to determine the mean and covariance after the transformation.

$$\begin{aligned}W_0 &= \kappa/(n+\kappa)\\ W_i &= 1/2(n+\kappa),\ i = 1 \ldots 2n\end{aligned} \tag{13}$$

The mean is given as the weighted mean of transformed sigma points

$$E(\mathcal{Y}) = \sum_{i=0}^{2n} W_i \mathcal{Y}_i \tag{14}$$

And corresponding covariance matrix as

$$\mathbf{P}_y = \sum_{i=0}^{2n} W_i\left(\mathcal{Y}_i - E(\mathcal{Y})\right)\left(\mathcal{Y}_i - E(\mathcal{Y})\right)^T \tag{15}$$

In order to use UKF in position tracking task, the predictor / corrector structure is kept, however the way the mean and covariance of the estimate is calculated is different from EKF. As there are two nonlinear functions (process and measurement), the unscented transform must be used twice. During the prediction first the sigma points are generated according to (11) from the current estimate mean and its covariance matrix, together with corresponding weights (13). Sigma points are then transferred through process function f (12) and prediction mean and covariance are calculated according to (14, 15), with process noise added. The prediction step formulas are therefore as follows:

- $$\mathcal{X}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k|k} + \left(\sqrt{(n+\kappa)\mathbf{P}_{k|k}}\right) \\ \hat{\mathbf{x}}_{k|k} - \left(\sqrt{(n+\kappa)\mathbf{P}_{k|k}}\right) \end{bmatrix}^T$$
- $W_0 = \kappa / (n+\kappa)$
- $W_i = 1/2(n+\kappa),\ for\ \ i = 1 \ldots 2n$
- $\breve{\mathcal{X}}_{k+1|k} = f\left(\mathcal{X}_{k+1|k}, \mathbf{u}_k, k\right)$
- $\hat{\mathbf{x}}_{k+1|k} = \sum_{i=0}^{2n} W_i \breve{\mathcal{X}}_{i,k+1|k}$
- $\mathbf{P}_{k+1|k} = \sum_{i=0}^{2n} W_i \left(\breve{\mathcal{X}}_{i,k+1|k} - \hat{\mathbf{x}}_{k+1|k}\right)\left(\breve{\mathcal{X}}_{i,k+1|k} - \hat{\mathbf{x}}_{k+1|k}\right)^T + \mathbf{V}_k$

In the correction (update) step, the new set of sigma points is generated (based on the mean and covariance from prediction step), sigma points are again passed through h (nonlinear function of measurement) and the mean is calculated together with the covariance. Finally the mean and covariance of the new estimated is calculated. The correction step formulas are as follows:

- $$\mathcal{Z}_{k+1|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k+1|k} \\ \hat{\mathbf{x}}_{k+1|k} + \left(\sqrt{(n+\kappa)\mathbf{P}_{k+1|k}}\right) \\ \hat{\mathbf{x}}_{k+1|k} - \left(\sqrt{(n+\kappa)\mathbf{P}_{k+1|k}}\right) \end{bmatrix}^T$$
- $\breve{\mathcal{Z}}_{k+1|k} = h\left(\mathcal{Z}_{k+1|k}, k+1\right)$
- $\hat{\mathbf{z}}_{k+1|k} = \sum_{i=0}^{2n} W_i \breve{\mathcal{Z}}_{i,k+1|k}$
- $\mathbf{S}_{k+1} = \sum_{i=0}^{2n} W_i \left(\breve{\mathcal{Z}}_i - \hat{\mathbf{z}}_{k+1|k}\right)\left(\breve{\mathcal{Z}}_i - \hat{\mathbf{z}}_{k+1|k}\right)^T + \mathbf{W}_{k+1}$
- $\mathbf{P}^{x,z}_{k+1|k} = \sum_{i=0}^{2n} W_i \left(\breve{\mathcal{X}}_i - \hat{\mathbf{x}}_{k+1|k}\right)\left(\breve{\mathcal{Z}}_i - \hat{\mathbf{z}}_{k+1|k}\right)^T$
- $\mathbf{K}_{k+1} = \mathbf{P}^{x,z}_{k+1|k}\mathbf{S}^{-1}_{k+1}$
- $\tilde{\mathbf{y}}_{k+1} = \mathbf{y}_{k+1} - \hat{\mathbf{z}}_{k+1|k}$
- $\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\tilde{\mathbf{y}}_{k+1}$
- $\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}^T_{k+1}$

The UKF does not require the calculation of Jacobian matrices and it better copes when the resulting distribution is far from Gaussian (typically when there are large differences in variances). However, these nice features are devalved by increased computational expense.

III. Simulation Results

Both methods were first implemented for simulation purposes using Matlab. The implementation is straight forward from the equations above, as Matlab supports vector / matrix operations. The number of simulation tests were performed with various levels of process and measurement noise, different probability of the beacons to actually work at all, etc. The example of comparing the estimate generated by both EKF and UKF is shown on Fig. 2. and Fig. 3. Fig. 2 shows the mean course together with the confidential ellipses drawn for x and y coordinate only (angle is not considered). Fig. 3. shows the same example with the robot chassis sketch, so the actual angle of the robot can be seen.

During the tests the probability of the beacon to be seen was set to 50% in each step. Simulated variance of beacons relative angle measurement was set to $\pi / 8$.

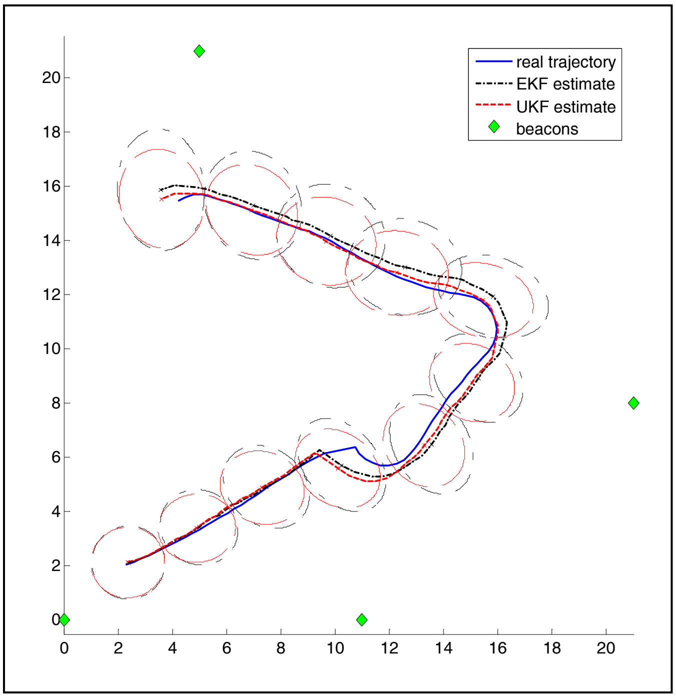


Figure 2. Comparison of real (simulated) trajectory and estimates produced by EKF and UKF filters.

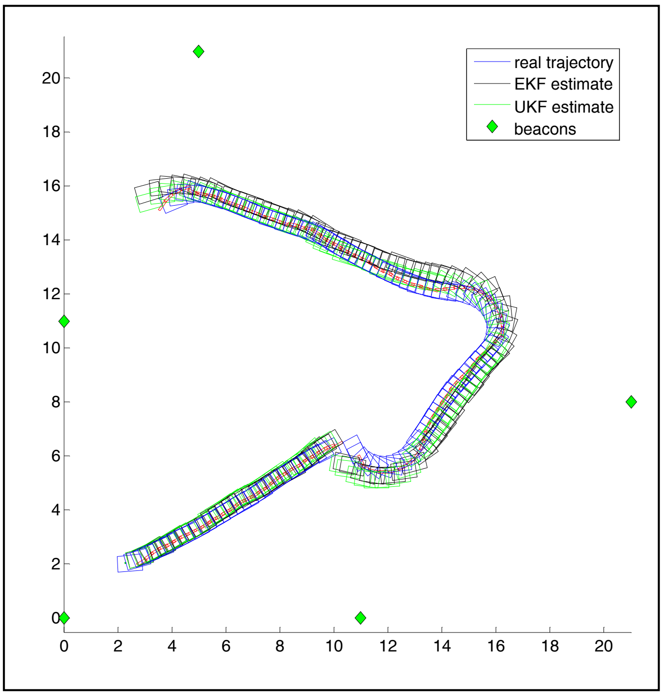


Figure 3. Comparison of real trajectory and the estimates with robot actual size drawn

To illustrate the robustness of the method, during the test a sudden change in robot's position is applied (robot is moved for 1 meter in x direction, 0.5m in y direction and turned clockwise for $\pi/2$). Actions are generated by the simple planner (motion towards randomly generated goal).

We can see that both filters can localize the robot successfuly; there is no significant difference in the mean nor the covariances. Filters can handle the sudden change in robot position and converge towards correct position. Further tests indicate that localization still converges for probability of beacons to be detected dropped to 15 % (for 4 beacons).

While the implementation of UKF is somewhat simpler (there is no need to calculate the Jacobians), the single step of the filter is about 4 times slower compared to EKF. This is not a huge difference, however, when computational resources are limited, such difference can be essential. This was proved during the experiments with real robot, described in following chapter.

IV. Experimental Results

Both methods were verified with the experimental robot Leela, equipped with the infrared beacons scanner, see Fig. 4. for the ring of receivers around the neck. The scanner covers full 360 degrees range. The robot has differential drive chassis and while equipped with IRC sensors on wheels, the odometry readings were not used to determine the actions, both translational and rotational velocities were taken from the controller. Controller performance imperfections, wheel slippage and other noise sources were all modeled by the process noise matrix $\mathbf{V}_k$.

Figure 4. Experimental robot Leela used for filter verification

The beacon scanner and the emitters communicate with each other using two wireless technologies. The low power comsuption radio modules with free 433MHz modulation are used for one way data transmision, in the direction from the scanner to the beacons. This way the proper beacon is selected for transmission and thus beacon identification problem is solved. The beacon starts with data transmition immediatelly after the proper identification number is received. The transmition from selected beacon to the scanner is also one way and it is based on infrared principle. The beacon uses infrared LED (880nm light-wave) with carrier frequency 38kHz. This signal is detected by infrared receivers on the scanner. As the emitted infrared light is easy to missdetect, software filter is implemented to calculate proper beacon relative angle.

In order to determine the true position of the robot, the image processing of images acquired during the test by the static camera mounted above the experimental grounds was used.

The implementation of both filters for the robot was done in C language, with our own implementation of necessary matrix operations. All the code runs on 8-bit AVR family processor. ATmega128 is a low cost 8-bit processor with programmable 128kB flash memory, two serial communication interfaces and eight analog to digital converters running on 16MHz frequency. The processor acts as the main control unit and it runs both the localization algorithm and path planner.

The path planner is based on the position estimate, so we can not directly compare the filters during single run. Moreover, both filters can not run simultaneously as the processing time would prolong the sampling rate to unusable level. Therefore the examples, given on Fig. 5 and Fig. 6. show independently the comparison between real trajectory of the robot and position estimate produced by the filters during the travel. The tests were performed in the lobby of A4 building at the Faculty of Mechanical Engineering, the obstacles are drawn on the figures just to give an idea about the environment.

One can see that with the real robot the estimation corresponds to the simulation results for both filters. The UKF position estimation error is slightly higher, however, this is not caused by the behavior of the filter itself, but by the lower sampling rate due to higher computational demands of the filter.

.

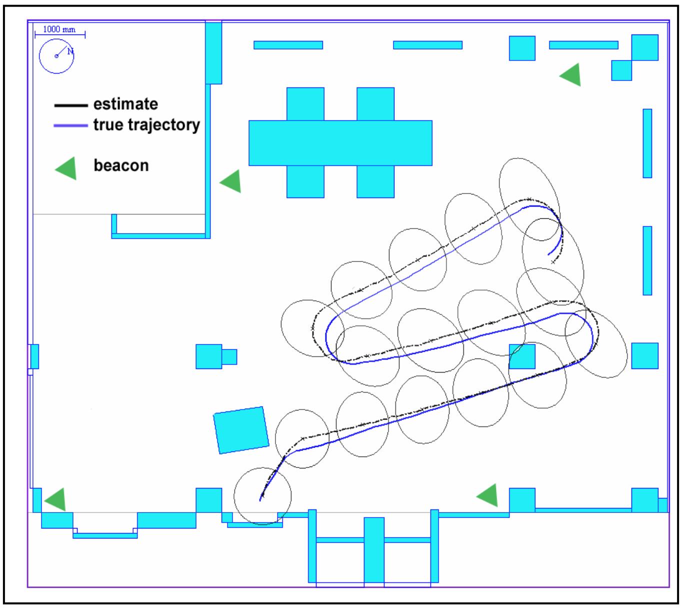


Figure 5. The real trajectory of the robot and corresponding estimate obtained using Extended Kalman filter

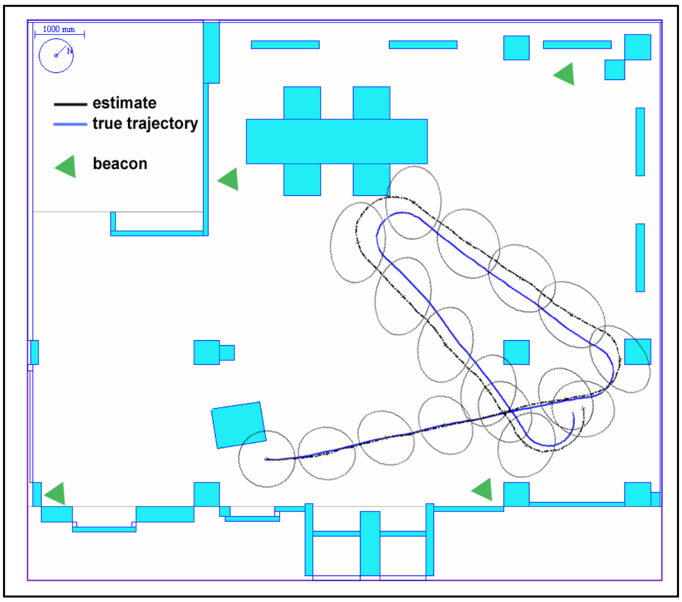


Figure 6. The real trajectory of the robot and corresponding estimate obtained using Unscented Kalman filter

V. CONCLUCIONS AND FUTURE WORK

Bayes filters are the basic tools for state estimation in robotics. The Extended Kalman filter and Unscented Kalman filter are among others (e.g. Particle filter) the most popular estimators nowadays. The mobile robot localization problem can be considered solved using landmarks extraction and further Bayesian filtering, however, there are applications where the beacons can justify its main drawback – the necessity to position the beacons in given locations. Such application can profit from very low computational requirements set for such a system. Thus the beacons based localization is (apart from other uses) ideal for the students in robotic teams, as they can solve nontrivial task of localization with cheap hardware.

Further advantage of beacons based localization is in the fact that number of robots that use the beacons is virtually unlimited. Therefore beacons can provide the base for the students to compete with different localization approaches while all their robots (of possibly different nature) have the same environment information. According to our experiences this aspect is strongly underestimated in competitions (frustration of the students with uneven conditions).

The aim of the paper was to describe in understandable way the method ideal for low cost robot localization that can be implemented by the students while helping them to understand the basics of Bayesian filtering. What type of filter to use in the task is not the main goal, however, when computational power restrictions are strong, the EKF simply wins the race. UKF benefits from better approximation of true non-Gaussian distribution are erased by higher computational cost leading to longer sampling rates and less stable estimates.

The future work will be focused on using the beacon / receiver combinations mounted on the robots, so the multiple robots can help to localize themselves. This should lead to reduction of number of beacons needed to install statically, however such need can not be reduced to zero, as only the relative information can be extracted from dynamic beacons measurement.

ACKNOWLEDGMENT

Published results were acquired with the support of the Academy of Sciences of the Czech Republic under the research plan AV0Z20760514 "Complex dynamical systems in thermodynamics, fluid and solid mechanics" and with the support of the Operational Programme Education for Competitiveness, project number CZ.1.07/2.3.00/09.0162 „Knowledge and Skills in Mechatronics - Innovations Transfer to Practice“. Authors would also like to thank Petr Schreiber for his help with C coding and Michal Růžička for his work on beacons hardware.

REFERENCES

[1] H. Choset, et al., “Principles of Robot Motion", MIT Press, 2005.

[2] S. Thrun, W. Burgard, D. Fox, “Probabilistic Robotics”, MIT Press, 2006.

[3] K. Arras, N.Tomatis, “Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision”, *Third European Workshop on Advanced Mobile Robots (EUROBOT ‘99), Zurich, Switzerland*, Sept.6-8, 1999.

[4] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera”, *9th ICCV*, vol 2, pp. 1403–1410, October 2003.

[5] K. Bekris, M. Glick, L. Kavraki, “Evaluation of Algorithms for Bearing-Only SLAM", *IEEE Intl. Conf. on Robotics and Automation, May, Orlando, FL*, p. 1937–1944, 2006.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Sensor Data Fusion for Mobile Robot

Věchet Stanislav*, Krejsa Jiří*
*Brno University of Technology, Faculty of Mechanical Engineering, Brno, Czech Republic,
e-mail: *vechet.s@fme.vutbr.cz krejsa@fme.vutbr.cz*

***Abstract*— Autonomous mobile robot must be equipped with a number of sensors of various measurement principles. The data fusion of measured data is essential for successful navigation of the robot. The paper describes the data fusion method based on Bayesian network. Apart from theoretical grounds of the used approach, the example is also given fusing the compass, GPS and odometry sensor data, because such sensors are commonly present in outdoor robots.**

***Keywords*— Automotive Application, Robotics, Sensor.**

I. Introduction

The understanding of basic principles in data fusion is the key knowledge that must be gained by the students of robotics regardless their later specialization. Bayesian filters are often difficult for the students to handle without prior understanding of underlying statistics. Bayesian networks can help the students to cope with the principles while at the same time the gained knowledge can be used directly as a tool in data fusion. The paper gives the detailed description of such a case.

Sensors data fusion belongs to one of the essential issues in mobile robotics. When the sensor suite of a mobile robot includes several sensors of different types the data fusion is necessary. Combining the sensor readings, the robot is designed to accomplish various tasks such as constructing a map of its environment, localizating itself in given map or recognizing objects that should be avoided [1]

There are several different aproaches, how data fusion methods are designed and used.

The data fusion is often used, because of its robustnes, for calculating the position and orientation of an autonomous mobile robot[3], as the fusion system is distributed, robust, and asynchronous. It is robust because the system is designed to keep working properly in spite of the failure, removal, or change in sensors configuration. The implementation of the reliable fusion system is based on distributed version of the popular Kalman filter developed by Durrant-Whyte and Rao.

Data fusion can generally be divided into the three main groups which can be classified according to sensors configuration as follows:

competitive - different types of sensors are used to measure the same attributes of given environment; usually there is information redundancy, which could be a source of errors,

complementary - each sensor reads different attribute of the same environment, this could be and disadvantage in case of sensor failure

cooperative - in that case one sensor depends on the other; they have to work together.

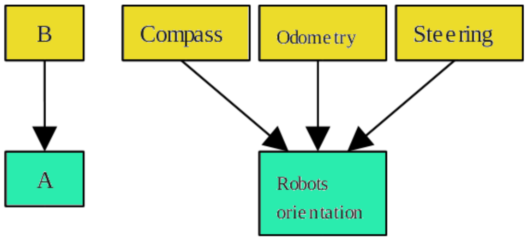


Fig. 1. The simplest Bayesian network (left) and the practical example (right).

Presented paper deals with a method which was succesfuly tested for complementary data fusion on mobile robot for outdoor environment equiped with several sensors of different nature. In particular, the experimental robot carries a popular laser range finder SICK, ultrasonic and infrared distance sensors, compass, GPS recever, digital camera and odometry IRC sensors. As the sensors are of variant sensing principle and purpose, the basics idea is to provide the proper information regarding robots position and internal states. In such a case the data fusion is a key algorithm.

So far we have been using complementary data fusion in all our robots. For such a purpose we have been developing a method of data fusion based on Bayes theorem, which is the base for most probabilistic method in robotics.

Presented paper describes the basic method for fusion of data acquired via odometry, compass and steering angle. Naturally, its easy to widespread this basic set of used sensors with other sensing devices. Therefore this method can be used as a main tool for measured data fusion.

Finally, two different environments were prepared to test the method with the real robot and the results from those experiments are discussed in chapter V.

II. Bayes Theorem

Bayes theorem belongs to a basic methods to deal with conditional probability. More precisely it relates the conditional probability of events A and B. Its well known how to derive it from basic conditional probabilities equations (for example in [4]), so here just the final state of the Bayes theorem equation is presented:

$$P(X|Y) = \frac{P(Y|X)}{P(Y)} \tag{1}$$

where:

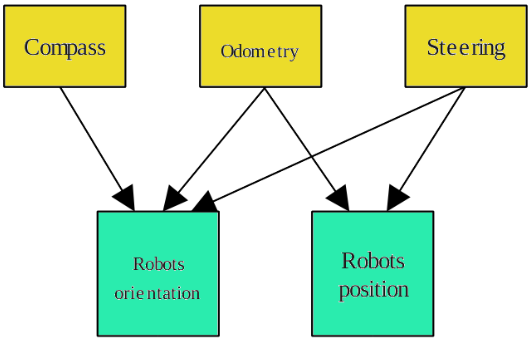


Fig. 2. Data fusion via Bayesian network.

- $P(X)$ is the prior probability of X; in sense what we know about X at the beginning, it is independent on any others variables,
- $P(X|Y)$ is called posterior probability; the conditional probability of X given$Y$, usually this probability represents the information what we are interested for: what is the probability of beeing in given state X if the robot sensors measured data Y.
- $P(Y|X)$ is the conditional probability of Y given X, sometimes called also the invers probability beacause it represents the situation: what is the probability that the collected measurement Y was measured in given state X.
- $P(Y)$ is the prior probability of Y; it acts as normalizing constant and if we have complete sets of possible states and measurements, this could be calculated based on total probability law.

As its shown, the main idea is based on $P(X|Y)$ calculation, if we know so-called inverse probability $P(Y|X)$. This probability can be obtained as the invers model of solved problem or by measurements on real system with collected input and output information. Afterwards, the paired set of inputs and related outputs is used for probability calculations.

Bayes theorem can be easily used in more complicated relationships with more then two events. One of possible applications of the Bayes Theorem is in Bayesian networks [6] or filters.

The primitive relation of two events described above could be considered the simplest network (see figure 1 - left). Basically, the Bayesian networks are primarily used for more complex relationship description (see figure 1 - right). On that figure the relations between some of the sensors discussed above are shown. Those relations are naturally created as it is intuitive for human. Its easy to derive how the final orientation of the robot is influenced by the sensors. Moreover, the existing Bayesian network can be easily modified to solve a more complicated problem (see figure 2).

III. Bayesian Network Simulator

There are three different sources of information which possibly could be used to calculate the robots true orientation (the heading angle of the robot):

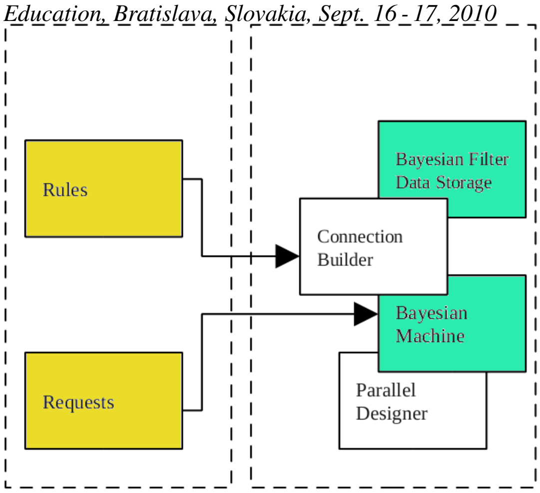


Fig. 3. Bayesian network simulator internal structure.

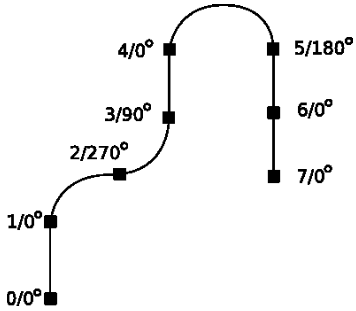


Fig. 4. Path followed by autonomous robot; Position/Robot orientation relative to robots previous orientation.

- the compass is used for absolute orientation measurement. Information from this kind of sensor can be easily converted to robots orientation,
- odometry give the indirect information about a relative change of the orientation, which has to be calculated from traveled distances of left and righ wheels,
- steering angle give also the indirect information about the relative changes in robots orientation. The change has to be calculated from the driving model (ackerman, differential, etc.).

To obtain the resulting orientation of the robot, the additional calculations performing the data fusion are necessary. Usually the data fusion from such different information sources is not straighforward.

We have prepared the simulator of Bayesian network to be able to work efficiently under various conditionas. The simulator internal structure is shown on figure 3.

The implementation of Bayesian network simulator consists of six important blocks:

Rules - user definition of problem structure, it means that the user is able to define given problem by set of high-level commands. These commands are based on written rules for probabilistic equations.

Requests - user requests to the simulator, represented by a set of possible commands, which define the requests to the Bayesian machine.

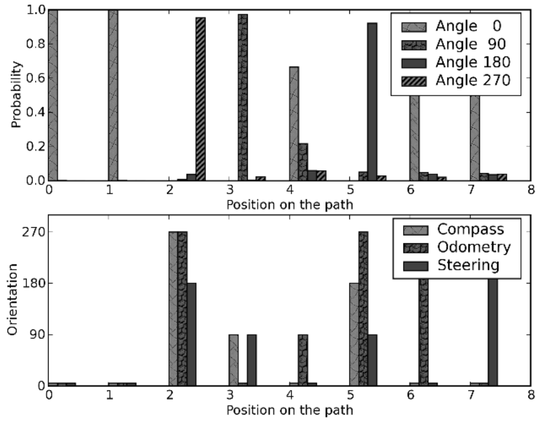


Fig. 5. Robot orientation identification via Bayesian network.

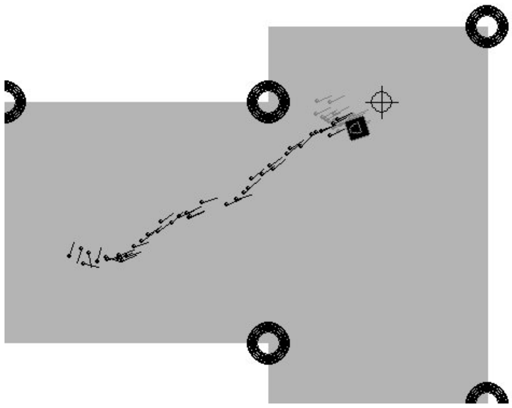


Fig. 7. Complete path traveled by the robot.

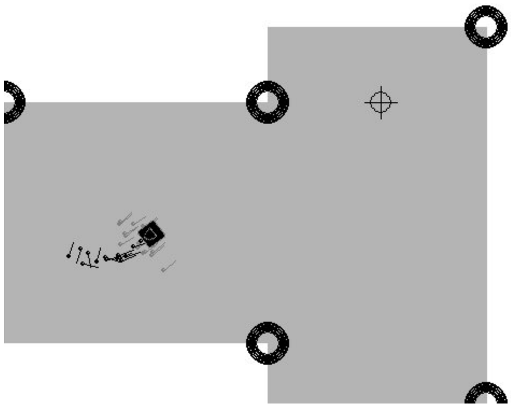


Fig. 6. Initial part of the path traveled by the robot in the Map-I.

Connection Builder - interface between user inputs definition and the real data structure, the probabilistic rules are translated into the form to be understood by the Bayesian machine.

Bayesian Filter Data Storage - the main data strorage, it holds the complete structure of given problem, probabilities and possible results.

Bayesian Machine - acts as a main computation framework, it computes all necessary probabilities requested by the user, there is an automatic inference mechanism based on Bayesian network theory as well as implementation of the set of mathematical equations and necessary probabilistic laws.

Parallel Designer - Bayesian network can be implemented as a parallel algorithm so this block organizes the parallel operations.

Experimental results obtained from presented simulator are shown in followed chapter as we have used it as a main tool for data fusion of robot orientation measurements.

IV. Data Fusion via Bayesian Network

To detect that our method to data fusion works properly, a simple simulation experiment with autonomous robot was first prepared. The robot was equipped with three means to determine its relative orientation on followed path. First one was thecompass which was used to measure absolute orientation of the robot. The second one was an odometry reading and the change in orientation was calculated from the difference in traveled distances of each single wheel. The orientation of the robot calculated from the steering angle represents the third variable.

The path traveled by the robot is shown on figure 4. There are seven number of checkpoints in which the orientation was measured. On that figure one can see the position number/true robot orientation in degrees relative to the robot previous location. The orientation was measured by all three methods and we use previously described tool to fuse the data.

The simulation results are shown on figure 5. On the bottom graph the orientation measured by compass, odometry and steereing in each position on the path is shown. The upper graph shows the probabilities that the robot is oriented 0, 90, 180 or 270 degrees. One can see that the highest probability (the output of the Bayesian network) in each point corresponds with the true orientation of the robot (see figure 4).

The best example illustrating how the data fusion works can be seen in point 5 on figure 5. The robots real orientation is 180 degrees and the compass measured that orientation properly. On the other hand the orientations calculated from odometry and steering were wrong (odometry 270 degrees, steering 90 degrees), but the probability that the robots orientation is 180 degree is still the highest. This is caused by different probabilities of correct orientation measurement.

Note, that compass measures with the same probability in all directions independently $P(Compass) = 0.95$. However, odometry and steering have high probabilities of success for angle 0, for example $P(Odometry = 0) = 0.95$, less for angle 90 or 270 $P(Odometry = 90) = 0.5$ and even lower probability for angle 180 $P(Odometry = 180) = 0.3$. These probabilities were measured in various practical experiments and they causes that if compass measured angle 180 degrees and odometry and steering 90 or 270 degrees, than these two values have less influence to the calculated orientation.

Presented simple example illustrates how to use Bayesian network in basic task. The method can be easily extended to be used in more challenging problems including continuous variables [4]. In such a case the discretization of corresponding variable is necessary.

V. Real experiments

The method described above was tested in various environments to ensure that the method is capable of working under the real conditions. The real experiments were prepared as follows.

Two different environments of different dimensions were prepared. The smaller one has the size of approximately 5x5 meters (see figures 6 to 10), while the bigger one is about 8x15 meters (see figures 11 to 13).

A. Map I

The robots goal was to reach the target position while the map was initially known and it performs the localization procedure during the travel. Only the orientation variable out of the localization estimate is depicted as it was fused by the Bayesian network. The xy coordinates were obtained by different localization method (not presented in this paper).

Each figure shows different part of the way of the robot. The real position of the robot in actual step is marked with black rectangle and the target position is marked by black circle.

Localized positions and fused orientations through the whole path traveled in the environment is marked by points with short line as the robots heading angle direction.

Figures 6 to 7 shows different stages of the path traveled by the robot through the given map. As the reader can clearly see, fused orientations are in corelations with the true orientation.

The complete path traveled by the robot is shown on figure 8. There can be seen that the estimated path is close to the true path of the real robot.

While the path traveled by the real robot is smooth (see figure 8 - real position), the estimated path is more discontinuous. This is caused by the precision of the probabilistic estimator. The error in position estimation is shown on figure 9.

Figure 10 shows the difference (in degrees) between the real and the fused orientation of the robot obtained via Bayesian network. The error in heading angle estimate is caused by nonhomogenous magnetic field and tires slip, however we found the results reasonable.

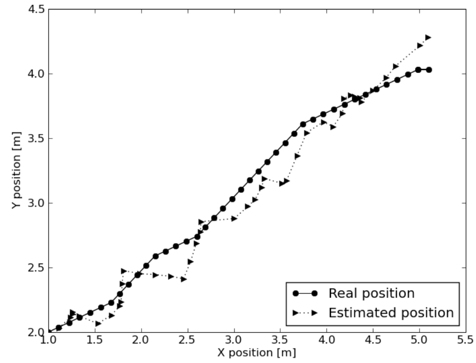


Fig. 8. Real and estimated path of the robot.

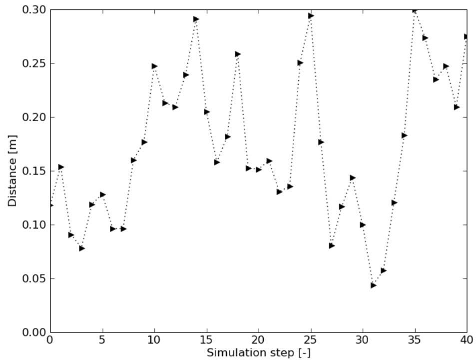


Fig. 9. Diferences between estimated and measured position.

B. Map II

Another test was performed in larger area covered about 100 square meteres. The task the robot should perform is to travel to the target position about 12 meters away.

The complete path with the dimensions of the experimental area is shown on figure 11. The estimated path folows the real path traveled by the robot and the precision is in same range as in experiments performed in Map I and is shown o figure 12 (compare with figure 9).

The comparison of real orientation of the robot and the fused orientation obtained via Bayesian network is shown on figure 13. The precision is also similar to the smaller map presented in previous chapter.

VI. Conclusion

We have presented the basic simulation and real experiments with data fusion via Bayesian network. Three different sensors (digital compass, odometry and steering) were used as the information source to estimate the orientation (heading angle) of the mobile robot and the fusion method based on Bayesian network was successfully applied.

The key issue in the method itself is proper determination of condition probabilities of mutually related events. The simplest approach to determine the probabilities is to directly use known errors in sensor measurement, or to create an inverse model of the problem.

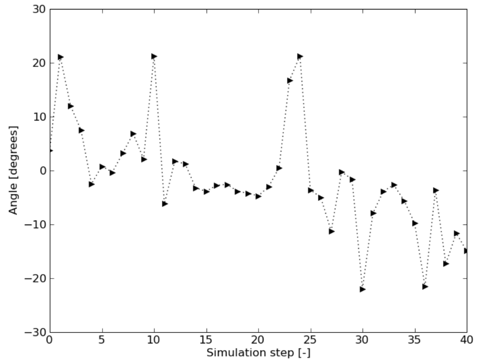


Fig. 10. Difference between the real and estimated orientation.

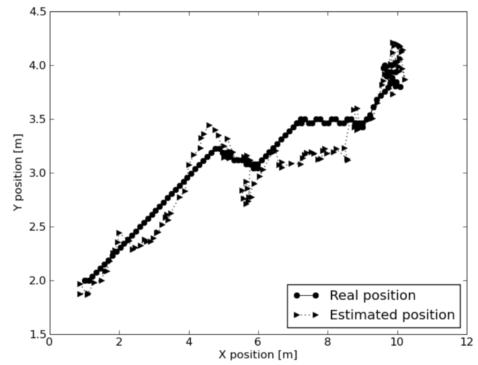


Fig. 11. Comparison of the real and estimated path of the robot in the Map II.

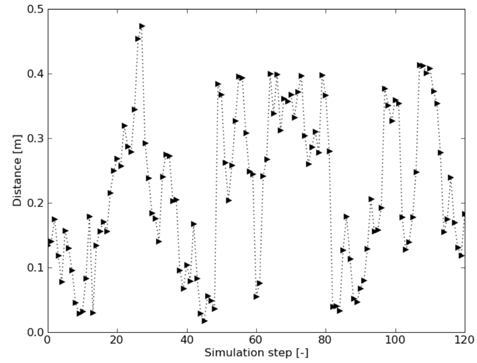


Fig. 12. The diference between real angle and fused angle via bayesian network, Map II.

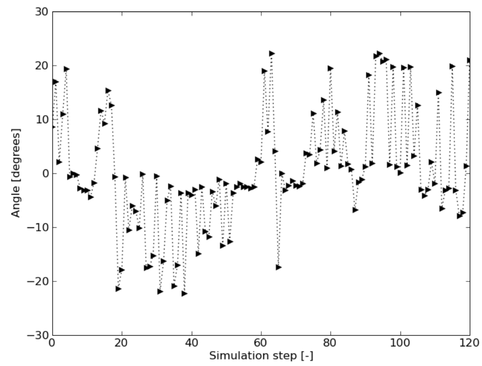


Fig. 13. The diference between real angle and fused angle via bayesian network, Map II.

Bayesian networks are simple yet powerful tool in data fusion field. Understanding the Bayesian theorem that is the base for probabilistic approach so popular nowadays in mobile robotics is essential when more complex methods and algorithms are applied.

We found described examples as both directly usable and of high educational value as the problem of heading angle determination from several sensors is for students easy to understand and thus uncover the underlying principles of probabilistic robotics.

ACKNOWLEDGMENT

Published results were acquired with the support of the Brno University of Technology, research plan FSI-S-10-29 "The development of sensors and control of unconventional mechatronic devices".

REFERENCES

[1] M. Kam, Z. Xiaoxun, P. Kalata, "Sensor fusion for mobile robots navigation," *Proceedings of the IEEE, Data Fusion lab., Drexel univ., Philadelphia.,* pp.108-119, 1997.

[2] Jin, Tae-Seok and Lee, Jang Myung and Tso, S. K., "A new approach using sensor data fusion for mobile robot navigation," *Robotica,* volume 22, pp. 52-59, 2004.

[3] J. A. Lpez-Orozco, J. M. de la Cruz, E. Besada, "An Asynchronous, Robust, and Distributed Multisensor Fusion System for Mobile Robots," *The International Journal of Robotics Research,* Vol. 19, No. 10, 914-932 (2000).

[4] A. Singhal, Ch. Brown, "Dynamic Bayes Net Approach to Multimodal Sensor Fusion," *Proceedings of the SPIE - The International Society for Optical Engineering,* pp.2-10, 1997.

[5] F. V. Jensen, "Bayesian networks and Decision Graphs," *Statistics for Engineering and Information Science,* Springer, Berlin, Heidelberg, 2001.

[6] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," *PhDthesis,* UC Berkley, Computer Science Division, 2002.

[7] F. V. Jensen, "An Introduction to Bayesian Networks," *UCL Press,* 1996.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

Looking for the path: image segmentation

Jan Coufal
Faculty of Mechanical Engineering
Brno University of Technology
Brno, Czech Republic
jan.coufal.cz@gmail.com

Jiri Krejsa
Centre of Mechatronics
Institute of Thermomechanics AS CR, v.v.i.
Brno, Czech Republic
krejsa@fme.vutbr.cz

Abstract—**Processing of the images acquired from the camera attached to the mobile robot in outdoor environment can be used for feature extraction or to distinguish the path the surrounding in the environment. Such information is further used in the planner and/or position estimator. The paper gives the overview of image segmentation method used on real data gathered by mobile robot Bender II during the tests for Robotour 2009 competition.**

Keywords: vision-based navigation, image processing, image segmentation, mobile robots

I. INTRODUCTION

Robust, two-dimensional path following for autonomous robot in outdoor non-urban environment is challenging task of monocular vision navigation. Characteristic features of this problem are shadow and illumination changes, no clear boundaries, changes of road surface and little or no prior knowledge of the roads. The task itself can be simply stated as extraction of road representation features, usually of relatively low data size from the image or a sequence of images acquired by the camera mounted on the robot. Road representation is further used as an input in path planning algorithm that determines subsequent motion of the robot, or together with other sensory inputs in the position estimator in cases when certain map features are known (typically the detection of the cross road)

Motivated by the Robotour 2009 challenge, we have developed a segmentation method invariant to illumination conditions with embedded adaptive database of path color model and direction extraction.

II. RELATED WORK

Using vision like a supplement in robot and vehicles navigation has been popular research field. Areas of the research can be classified depending on road conditions, used sensors and vision algorithms. In our survey, we focused only on monocular color cameras, as the hardware cost in binocular systems capable of well performed synchronization, that is the necessary condition for restoring the 3D distance map from the stereo images is still relatively high. The computational cost of algorithms in processing images from binocular systems is also an issue.

For robots running on well-structured roads, such as roads in urban areas, the primary attention is focused on lane tracking and curve fitting. Since the road has a relatively uniform surface, techniques such as color-based road segmentation [1] and edge detection [2] are used with high percentage of successful features extraction. Even in such well defined roads the road segmentation algorithms are required to exhibit invariance to shadows and luminance. For this purpose, many approaches contain transformation of the source image into different color space [3, 4, 5]

When robot is running in an unstructured environment, terrain classification and obstacle avoidance are in primary focus. Method proposed in [4] is based on construction of 2D scene model of outdoor environment. Image is converted to illumination invariant Ohta/Gevers color space, segmented by hybrid of thresholding and region growing methods and obtained clusters are classified into predefined classes using Support Vector Machines technique by their color and texture properties. Approach gives good results, but is highly dependent on predefined classes. In our method, we wanted to avoid implementing pre-learned knowledge database about environment.

Motivated by DARPA challenge, [5] developed method for direction extraction in desert terrain. His approach uses color transformation to c1c2c3 color space published in [6] for shadow elimination and better segmentation performance. The motion planning is based on a vision vector space, which is unitary vector represents collision-free directions in the image coordinate system. This vector space is projected to a preprocessed set of trajectories and the best candidate is chosen and used for motion planning. With respect to this approach we should state that while the ultimate goal is to develop the universal road/surrounding extractor, certain apriory knowledge regarding the environment can be successfully incorporated into the method. Images acquired from desert terrain exhibit features hard to find in images taken from the park or forest path.

Interesting approach is to involve the mean-shift algorithm for road segmentation and to use graph cuts for region merging. Reference [8] proposes a novel road following method, which firstly uses the Mean-Shift algorithm with embedded edge confidence to partition the images into homogenous regions with precise boundary. Then, according to the color statistic information of the road/non-road model obtained from previous frames, the Graph Cuts algorithm is used to achieve the final binary images and update the road/non-road model simultaneously. This combination of the advantages of Graph Cuts algorithm and Mean Shift algorithm effectively solves some difficult problems of conventional methods, such as the adaptive selection of road model under complex environments, and the choice of effective criteria for the region merging. Problem and main disadvantage is enormous computation time, which makes this method not suitable for real-time. In spite of that the method seems to be reasonable to use in initialize stage or the high evaluation time can be reduced by using the computation on GPU.

III. PROPOSED METHOD

We accept a few assumptions and simplifications for our solution. We assume that the vehicle maintains the contact with ground at all the time during its travel. We also assume that ground surface is relatively flat. Therefore, it can be treated as a ground plane.

The overall scheme of the method is drawn on Fig. 1. Image data are represented by the image matrix of particular color components extracted from the camera, usually in RGB color space.

A. *Preprocessing*

The purpose of preprocessing is to filter the noise in the image and to cut upper part of the screen. In [8] the road-following technique was presented, which can be used also as a horizon detector. Because of flat ground assumption, we don't have to implement horizon detector. The position of the horizon is constant and it depends only on the known construction parameters of the robot. Data included in the part of the image up to the horizon are useless for path detection and segmentation. Therefore we can cut this part off the image to speed up the performance of the algorithm.

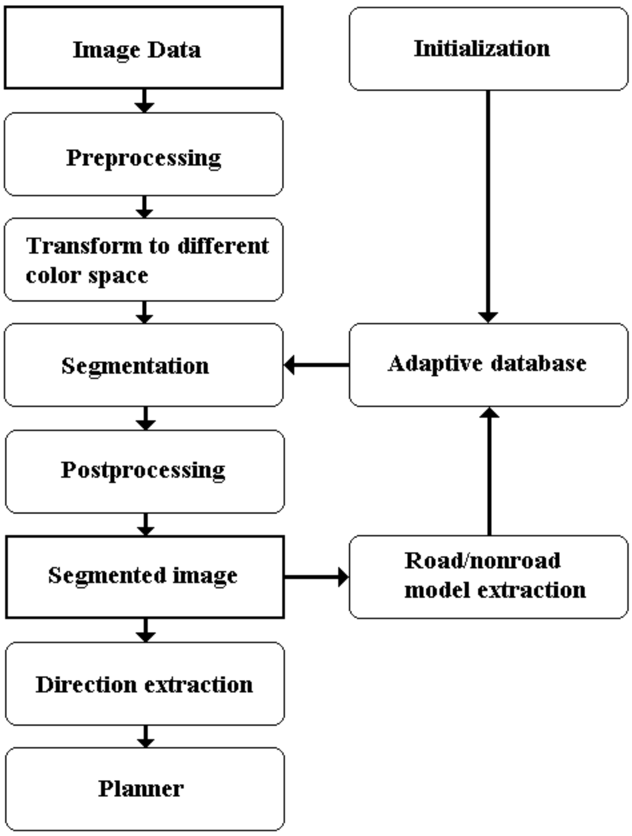


Figure 1. Principle of vision-based navigation algorithm.

Fast convolution with gauss kernel optimized in assembler is used as a noise filter.

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$
$$x,y\in\langle -n,n\rangle \qquad (1)$$

where $G(x,y)$ is radial symmetric function for a x-th and y-th element of the kernel matrix of size $(2n+1)^2$.

B. *Color correction*

The main purpose of color correction is to reduce the shadow and illumination change in the scene that effects acquired original image, as segmentation must be robust against such phenomena. We tested several color models e.g. popular HSI color model, Otha's color space [9], c1c2c3 and l1l2l3 color space described in [6]. Our segmentation experiments show, that c1c2c3 color model is the best shadow and illumination invariant color model for outdoor vision algorithms.

$$c_1=\operatorname{atan}\left(\frac{R}{\max(G,B)}\right)$$
$$c_2=\operatorname{atan}\left(\frac{G}{\max(R,B)}\right) \qquad (2)$$
$$c_3=\operatorname{atan}\left(\frac{B}{\max(R,G)}\right)$$

The coordinates of c1c2c3 are rescaled to byte values and stored in the bitmap data structure for further use.

C. *Segmentation*

The essential task of the whole process is to distinguish a road from nonroad surface and this is done by image segmentation. We have tested three bottom-up segmentation principles: simple thresholding, region growing and mean-shift clustering. Our experiments show, that satisfactory results are provided by thresholding method. Region growing and mean-shift methods have better segmentation results, but they are more computationally expensive, as in their nature is to classify all the pixels in the image to the class containing similar points with reference to their spatial domain. In our case, complete segmentation is not necessary. Problem of dealing with wrongly classified pixels is efficiently solved during post-processing.

Thresholding method is computationally fast and efficient segmentation technique. The distance between actual pixel and road representative pixel is computed using chosen metric for

Figure 2. Example of thresholding segmentation and postprocessing. Image was converted to c1c2c3 color space and segmented by thresholding technique. Reference point was obtained from trapezoid region in front of a robot and threshold value is 20. After that, combination of postprocessing methods was applied.

each pixel in image. If the distance between both pixels is less than distance criteria, then the pixel is classified as a road. The condition can be described as follows:

$$dist\|px, ref\| \le t_{\max} \tag{3}$$

where *px* and *ref* are points in three dimensional space determined by c1c2c3 color coordinates, *dist* is metric and t_{max} is maximum distance between pixels in chosen metric. For our purposes the Euclidean metric is sufficient, but other metrics can be implemented, e.g. Mahalanobis metric [10].

In our approach, road representative pixel is provided by adaptive database, which stores road representatives from previous images, therefore initialization phase is needed when the first representative is chosen.

Segmented data are stored in array of bit values, where value of one represents road while zero represents the background. This data structure is very useful in post-processing.

D. Post-Processing

In this section, data obtained in segmentation stage are further adjusted. The combination of dilatation and erosion methods is used to delete incorrectly classified pixels out of road and merge larger areas together. Sequence of postprocessing methods is dependent on the magnitude of the threshold value. If threshold value is smaller than hypothetical optimum, number of unrecognized road pixels will be bigger. In this case is better to use “close” method first (dilatation, erosion). If threshold value is higher, number of nonroad pixels classified as a road will be larger. For that reason is better to use “open” method first (erosion, dilatation).

Our experiments shows, that robust results are provided by relatively higher threshold value and corresponding postprocesing combination of methods. In our case we use a small number of iterations of the open method (erode-dilate) and larger amount of iterations of the close method (dilate-erode).

After that, we need to fill small regions in the image. For that purpose we used the connected component labeling technique. It is an algorithmic application of graph theory, where sets of connected regions (components) with the same value in the bit map are uniquely labeled. Number of elements with the same label is counted and if component have less than minimum number of elements, then its value is inverted and component is merged with closest one. Result data structure of the post-processing is bit mask.

E. Direction extraction

In this section we present our approach to achieve the control angle further utilized by mobile robot planner module. Our inputs are a bit mask, which contain information about collision-free space, and angle φ between GPS coordinates of next waypoint (goal) and actual mobile robot coordinates. It is assumed that camera is mounted in the middle plane of the robot, so the vertical center line in the image is the projection of robot actual heading angle. We also assume that velocity of robot is not significant. Therefore the trajectory of robot motion in the image can be represented by the straight line.

Now we can formulate motion planning quantitatively. Lets w_α is a weight function for trajectory in direction determined by steering angle α . We can compute it as follows:

$$w = d(\alpha)^k \cdot \cos(\alpha - \Phi) \tag{4}$$

where *k* is constant determining nonlinearity between distance of two points in image and distance between two points in real world. Angle ϕ is a difference between waypoint angle φ and heading angle of the robot.

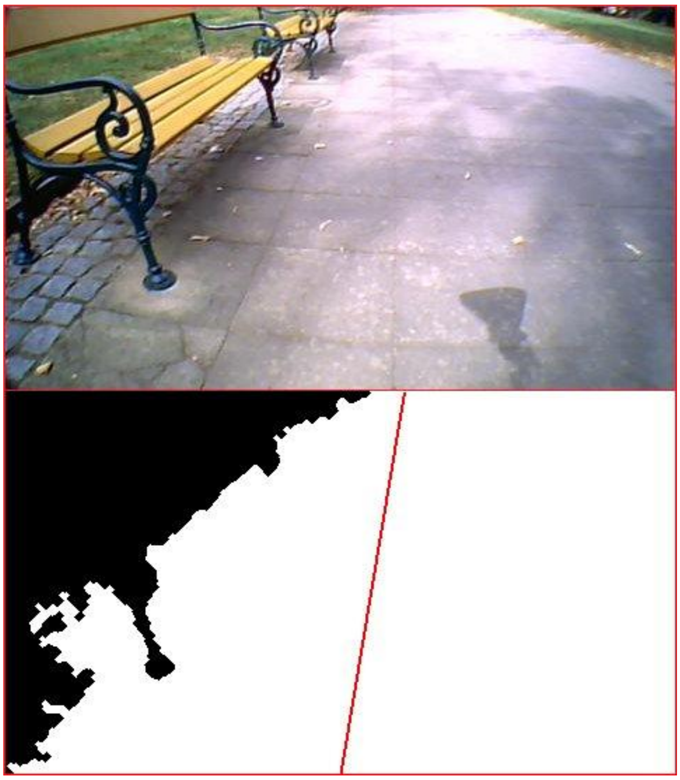


Figure 3. Result of direction extraction algorithm.

Function *d(α)* is number of collision-free pixels in the bit map in direction determined by angle α . We can compute *d(α)* as follows:

$$d(\alpha) = \sum B(x, y) \tag{5}$$

where B is the bit mask obtained during the post-processing and *x*, *y* are image coordinates. Value of function *B(x,y)* is 1, when value in bit mask at *x*,*y* coordinate is true, otherwise *B(x,y)* is 0. The value of function *d(α)* is sum of the pixels classified as the road.

Let's denote that *u* and *v* are coordinates in trajectory coordinate space τ with zero point in the center of the bottom edge of screen. Transformation formula between x,y and u,v:

$$\begin{aligned} y &= height - u \\ x &= \frac{width}{2} + v \\ v &= u \cdot \tan(\alpha) \end{aligned} \tag{6}$$

where height and width are the dimensions of the image. The trajectory is fully described by angle α. Therefore, we formulate the motion planning problem as an optimization problem. We are looking for trajectory angle α that maximizes weight function *w*.

There is countless number of potential trajectories and we can't evaluate them all. Therefore we have pre-computed set of trajectories for different α and final trajectory is obtained as their weighted average.

$$\alpha_{final} = \frac{\sum w_i \alpha_i}{\sum w_i} \tag{7}$$

F. Extraction of reference point

At the end of one loop of the algorithm, the resulting bit mask is placed to original image. The result of multiplication of both images is a new image with valid data for extraction of new reference point. With inversion of bit mask we can acquire non-road reference points. Both points (road/non-road) are stored in database.

Using recognized path from last images gives to algorithm adaptability to continuously changing conditions.

G. Initialization stage

In initialization stage we pre-compute all necessary parameters, trajectories and reference point for path recognition. We assume that robot is standing during initialization on the path, heading towards next waypoint.

To extract the reference point the trapezoid region in front of the robot is used [5]. The image for reference extraction must be static. To exclude moving object in image during initialization we implemented autocorrelation function. If two images in sequence are static, autocorrelation function reach maximum and image is suitable for computing of road reference point.

IV. Experiment

We tested our navigation algorithm on real data gathered by mobile robot Bender II during Robotour 2009 competition. The camera used to acquire the images was Megapixel USB2 Wide Angle Webcam Live WB-6200p.

The images on Figure 4 show pictures randomly selected from the robot track. Each figure is composed of four subframes illustrating the output of individual algorithm steps: top left the original image, top right the image converted to c1c2c3 color space, bottom left the threshold segmentation result and bottom right direction extraction.

First image shows the performance of direction extraction algorithm on border of the path. Algorithm reacts correctly. Next two images show segmentation in difficult lighting conditions. Both results are satisfying, i.e. usable by the planner to keep the robot on the path. Last image shows the drawback of presented algorithm. Because the weight is a sum of all positively classified pixels in tested direction, we can expect, that some obstacles will be ignored. This property was implemented on purpose, because small areas of different terrain type may occur and true obstacles are detected by other means (laser rangefinder). Nevertheless, with further research taking into account the shape(s) of segmented portion of the image such issue can be addressed.

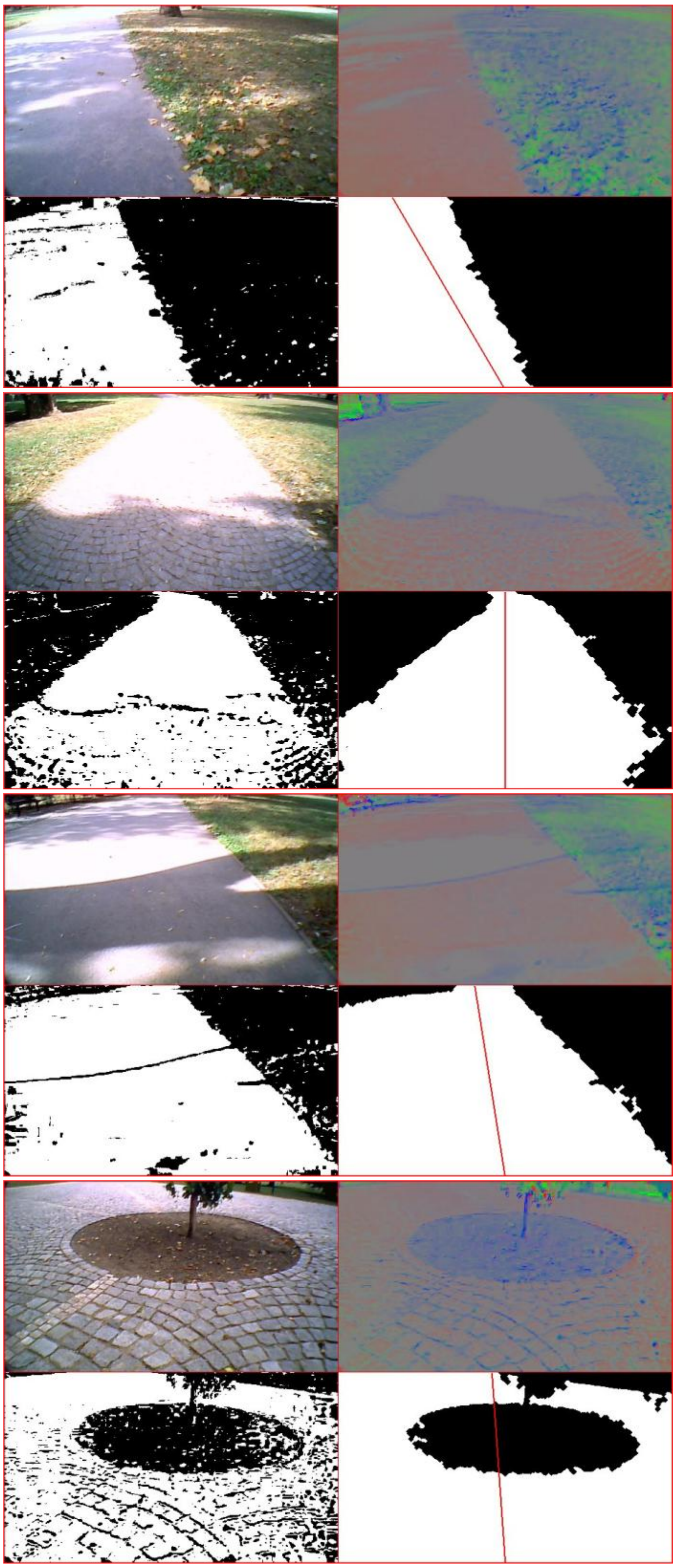


Figure 4. Examples of algorithm performance.

The computation time of the algorithm for a single image of (480x360) pixels without code optimization is around 200ms. The algorithm was implemented in C#. Such speed is sufficient for relatively high velocities of the robot, as the distance traveled during the processing can be kept in reasonable values.

V. CONCLUSION AND FUTURE WORK

In this paper, we report our development of robust vision-based algorithm used for motion planning of autonomous mobile robot. To achieve good performance, data are first filtered to remove the additive noise. We transform regular RGB color coordinates to c1c2c3 color model for better segmentation result as this color model is less sensitive to the changes in lighting conditions. Then threshold segmentation method classifies the image data to road or non-road class with respect to the representative point of the road. Segmented data are stored as bit array and are post-processed. The holes in image are filled and only few continuous clusters are obtained. Bit array is applied on original data and new reference point for segmentation in next frame is stored in the database. Finally, the direction extraction algorithm computes the best angle to achieve desired direction. Such image processing algorithm is not difficult to implement and can be used by other researchers not only in robotics competitions.

Our future work will be focused on reduction of the computation time of complex algorithms via OpenCL for GPU programming, as most of the image processing routines can be easily transferred to semi-parallel versions, ideal for multiple processors used on GPU. With this technology, we will be able to increase processed images frame rate to high values on commonly available hardware.

ACKNOWLEDGMENT

Authors thank for the support of the Academy of Sciences of the Czech Republic under the research plan AV0Z20760514 and for the support of the Operational Programme Education for Competitiveness, project number CZ.1.07/2.3.00/09.0162.

REFERENCES

[1] M. Sotelo, F. Rodriguez, L. Magdalena, L. Bergasa and L. Boquete, “A color vision-based lane tracking system for autonomous driving in unmarked roads” *Autonomous Robots*, vo.16, no. 1, 2004

[2] M. Ekinci, F. W. J. Gibbs, B. T. Thomas, “Konwledge-based navigation for autonomous road vehicles” Turkish Journal of Electrical Engineering & Computer Sciences, 2000

[3] Alvarez, J.M & Lopez, A.M. & Baldrich, R., ”Shadow resitstant road segmentation from a mobile monocular system” *Iberian Conference on Pattern Recognition and Image Analysis*, vol. II, 9-16, 2007

[4] Juan Gabriel Aviña-Cervantes, Michel Devy, “Scene Modeling by ICA and Color Segmentation” MICAI 2004: 574-583

[5] D. Song, H. N. Lee, J. Yi, and A. Levandowski, “Vision-based motion planning for an autonomous motorcycle on ill-structured roads” *Auton Robot*, vol. 23, pp. 197–212, 2007.

[6] Gevers, T., & Smeulders, A.W. M. (1999). “Colour based object recognition” *Pattern Recognition*, 32, 453–464.

[7] Lin, L. Zhou, W., "A Robust and Adaptive Road Following Algorithm for Video Image Sequence", *Lecture Notes in Computer Science*, 2007, NUMB 4681, 1041-1049

[8] Rasmussen, C. “Grouping dominant orientations for illstructured road following” In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, June 2004.

[9] Ohta, Y., “Knowledge-Based Interpretation of Outdoor Natural Color Scenes” *Research Notes in Artificial Intelligence 4*, Pitman Advanced Publishing Program (1985)

[10] Grudic, G., & Mulligan, J., “Outdoor path labeling using polynomial mahalanobis distance” *In Robotics: science and systems*, Philadelphia, PA, June 2006.

Web based remote mobile robot control

Jaroslav Hanzel
Institute of control and industrial informatics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
jaroslav.hanzel@stuba.sk

Abstract—**The paper deals with the internet based robotics. The attention is focused on the proposal and implementation of the experimental web based interface for the remote control of the mobile robot. The proposed system contains visual feedback to assistance the operator for safe navigation of the robot in dynamic environments. The control system utilizes the client - server architecture and is mainly implemented in the platform independent Java programming language.**

Mobile robot; telerobotics; visual feedback control; Human Machine Interface

I. Introduction

With increasing use of the internet, the number of smart devices or systems dedicated to service, safety and entertainment is growing. These are composed of the distributed computer systems with use of the observation cameras, manipulators and mobile robots. As the idea of web robots or web-based robots is relatively new, it draws attention and interest of researchers. In addition to the control in hazardous environments, which are traditional telerobotic operations, internet extends the limits of real robots using robots in the areas known as telemanufacturing, teleeducation, telesurgery as well as a guide to a museum, in traffic control, space research, in the rescue operations during disasters, domestic cleaning or care. Although the internet provides for the teleoperations inexpensive and easily attainable information channel, there are many problems that must be resolved before the successful achievement of its real use. These problems are mainly due to the limited bandwidth and the arbitrarily large transmission delays that significantly affect the performance of telerobotic systems based on the Internet. For these reasons, it is necessary to equip the robot with a high level of autonomous behaviour. An intuitive user interface for operators is required for controlling the robot remotely.

Web based robotics uses a web browser for remote control of the robot and it differs from the traditional teleoperations in several aspects. The delay and throughput of the Internet are highly unpredictable, unlike traditional teleoperations, where the interfaces have known and guaranteed delays. Web based remote controlled robot also needs a high degree of resistance to the loss of the data packets. Web robots are controlled in most cases by people with little expertise and limited experience, unlike traditional tele-robots, which are operated by trained operators, and therefore their behaviour also become an important factor in the system design. Web robots deal with problems of a complex, dynamic environment in terms of the unpredictable delays in the network communication. Therefore their design and execution itself bring many challenges in addressing these problems.

This contribution deals with mobile robot control system via a web interface. The system should include a standard network protocol and interactive Human Machine Interface (HMI). Using a web browser, a remote operator can control a mobile robot with visual feedback over the internet. Using an intuitive user interface allows internet users to control mobile robot and implement useful tasks remotely.

II. System design

Research on remote controlled systems deals with a new generation of network telerobotic systems for real use, such as telemanufacturing [1], teleteaching [8] and telemedicine [7]. These systems combine advanced networking technology with intelligent mobile robots [2], [5], [6]. Modern telerobotic systems should have several properties to enable their efficient and flexible use. Among those there is the requirement of the:

• universal interface for easy integration of different types of robots into the system,

• intuitive user interface and the adequate feedback,

• easy expandability of the system for adding more complex function,

• implementation of the cooperative approaches to solve complex tasks,

• high degree of autonomous robot behaviour and intelligence.

With the rapid growth of the internet, several available communication technologies are implemented in a networked environment. Current internet protocol used by web browsers is the Hypertext Transfer Protocol (HTTP). A Communication Gateway Interface (CGI) is attended to link the external applications with the web server. By means of a Hyper Text

Markup Language (HTML) a requirement from the client to the server to start the process of executing a certain predetermined actions on the server can be specified. Dynamically generated HTML page can return results to the client. On the other hand, CGI has a number of shortcomings such as relatively slow speed of response. It must be also generated a complete HTML page with every client request. So this method of communication is not very suitable for remote control in real time. Contrariwise Java (object oriented programming language) offers the possibility to implement network connections and thus avoid restrictions of the CGI.

The relatively flexible and extensible approach for such tasks is to use a central server architecture [4], as shown in Fig. 1. All clients and servers are connected to a central web server. It is necessary to know the location of the web server and the reciprocal communication with each other through a web server. With this architecture all the video services and robot control services can either be provided for a single computer, or it may be possible to connect multiple computers. It is very easy to add more computers to control the robot and to process the graphical data or for the purpose of the control of more robots.

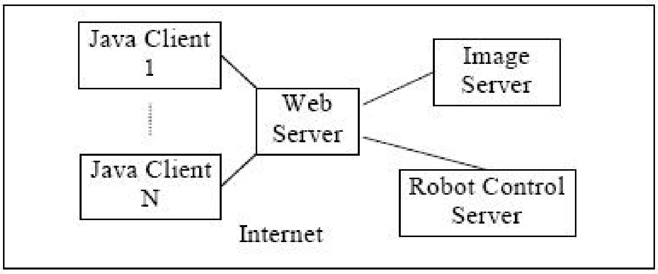


Figure 1. System architecture [4].

III. Hardware and software configuration

When designing the hardware structure of the telerobotic system, it is necessary to consider several factors related to the intended practical use of the system and it is also necessary to take financial possibilities into account. Fig. 2 shows the proposal of a hardware system for the simple remote controlled robot.

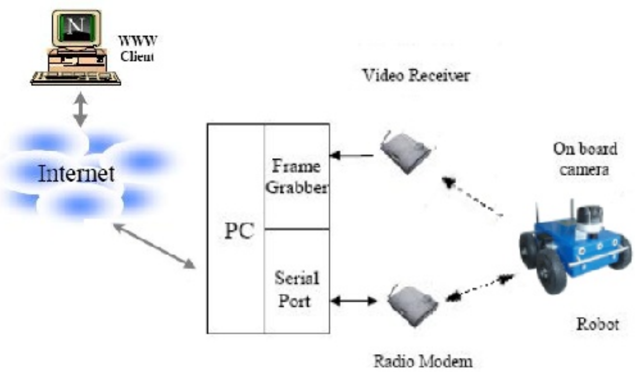


Figure 2. Configuration of the robotic system.

Main host computer communicates with a mobile robot through a radio modem connected to a serial port. The main computer is connected to the network by standard network interface. The front part of the robot is equipped with a camera, that gives the user a clear view of the environment appearing before the robot. The robot can also be equipped with various sensors (eg ultrasonic, laser), which help to provide a more complex sight of the robot working environment. Video signal from the camera located on the robot is captured by the frame grabber of the main computer and it is sent to the client.

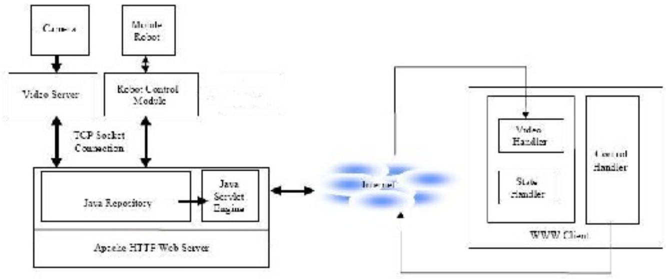


Figure 3. Software architecture.

As a web server the application Apache HTTP web server working on multiple platforms such as MS Windows or Linux is used. The entire software system consists of several independent modules for optional services, each of which contains a server-side program and client-side Java applets. Java servlet in the Apache web server handles the communication between clients and servers, as shown in Fig. 3.

IV. Control and visual module of the system

Operating of the mobile robot is performed by the robot control module. In the primary stage of the implementation of the control module, certain basic functions such as the controls for the movement, change of the speed and stop function are inserted. More intelligent forms of the behaviour are possible to integrate afterwards.

When the system begins to function, the Java program will run and accepts commands sent from the client and controls the movement of the mobile robot by the radio modem connected to the serial port. The robot can be controlled at the same time only by one user and other users have to wait in a queue until the current operation is completed. At the same time the program sends the information from the robot, such as the ultrasonic sensor data and state of the robot, to the clients. In order to reduce transmission time, any information is transmitted in the form of the character strings and sent to all clients connected to the server. These strings are interpreted and displayed on the client side.

A key element of the mobile robot remote control is an image from a camera placed on the robot transmitted to the client side. The image quality and speed of transmission should be sufficient to provide maximum information in real time for the safe and efficient remote robot control. Number of projects dealing with the transfer of images via web are using server push technology. The video is composed from a stream of static images sent by the Java program via sockets to the Java applets. In this system, the images captured from the frame grabber are compressed into JPEG format by the software implemented in C++. Subsequently these images are sent from the image server to the web server. Java program streams these JPEG images to all clients connected to this web server. On the client side Java applet restores the image after its receiving.

V. Web interface

Simple user interface is designed to provide basic information necessary for safe remote control of the mobile robot and it also provides the necessary basic controls. The user interface may consist of several Java applets, as shown in Fig. 4.

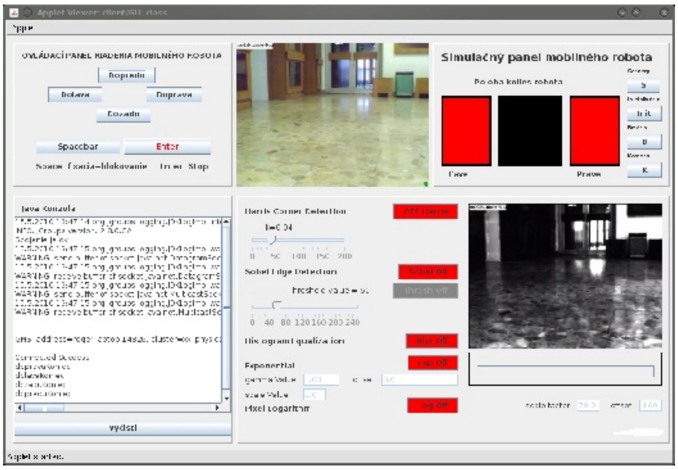


Figure 4. Implementation of the web interface.

On-line instructions for the robot are processed by the control panel, which may be formed in the basic version of the four directional buttons. The user can directly control the mobile robot by clicking on the direction buttons on the control panel, or by use of the keyboard for fast and complex control, such as change or adjust the chosen speed, or eventually by input of the coordinates of the target. The image display applet shows a visual feedback in form of the continuous stream of JPEG images. The virtual environment applet can show some basic information about the mobile robot and workspace, and analyse the feedback information from the mobile robot for example in the form of an environment map. Users can monitor for example the obstacles near the robot, the travelled path and current position and speed of mobile robot. The active user can control the movement of mobile robot through interface with visual feedback. Other users can only track the visual and sensory feedback and they have to wait until the first user logs off the network to control the robot.

VI. Application practice

The experimental remote control system is intended to gain the practical experiences with telerobotic systems and to find effective approach for implementation of systems capable of operation in the environment of public network. The aim of the implementation was to prove the correctness of the proposition. The system is presently actively developed and there is a strong requirement for searching of its optimal structure and experimenting with its final solution. Though, the initial testing experiments in the local network afford opportunity to control the robot in loopback with small enough delays and with sufficiently fast video refresh rate.

Properly designed telerobotic system can be successfully utilized in the education process how it exemplifies the similar telerobotic system [3]. The proposed telerobotic system was primarily dedicated for an inspection robot designated for exploration of unknown environments. Although the system is still in development stage, in the future, after some minor changes, it can be applicable also in the process of education. With the use of this telerobotic system, students can obtain the beneficial experiences with teleoperations and telepresence.

VII. Conclusion

The aim of the paper is to analyse the options and outline a possible structure and implementation for an experimental network telerobotic system for internet users, who can control a mobile robot in dynamic environment remotely from their home. The system allows internet users to control the mobile robot with utilization of the data obtained by the robot sensory system using a web browser. On the client side the obtained information is processed in order to encourage operator to safely control the robot. The visual feedback module provides fast image updates and presents a relatively credible real time visual information for the web users.

Acknowledgment

The work has been supported by grant VMSP-P-0004-09 “Inteligentné riadenie servisného robota”. This support is very gratefully acknowledged.

References

[1] M. J. Bailey, “Tele-Manufacturing: Rapid Prototyping on the Internet,” IEEE Computer Graphics and Applications, vol. 15, no. 6, pp. 20-26, Nov. 1995.

[2] W. Burgard, et. al., “The interactive museum tour-guide robot, Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence,” Madison, Wisconsin, United States, pp. 11 – 18, 1998, ISBN:0-262-51098-7.

[3] P. Petrovič, A. Lúčny, R. Balogh, D. Ďurina, "Remotely-Accessible Robotics Laboratory", Acta Mechanica Slovaca., June, 2006. Vol. 10(2-A), pp. 389-194.

[4] A. Sayouti, F. Qrichi Aniba, H. Medromi, “Control Architecture design for a Mobile Robot via the Internet,” 9th International PhD Workshop on Systems and Control: Young Generation Viewpoint, 1. - 3. October 2008, Izola, Slovenia.

[5] D. Schulz, W. Burgard, A. B. Cremers, “Predictive simulation of autonomous robots for tele-operation systems using the world wide web,” in IEEE/RSJ International Conference on Intelligent Robots and System, Victoria, B.C., Canada, October 1998.

[6] R. Simmons, “Xavier : An autonomous mobile robot on the web,” in International Workshop On Intelligent Robots and Systems (IROS), Victoria, Canada, 1998.

[7] D. Wright, L. Androuchko, “Telemedicine and developing countries,” J Telemed Telecare, 1996, 2(2), pp. 63-70.

[8] S. Yuanchun, et. al, “The smart classroom: merging technologies for seamless tele-education,” Pervasive Computing IEEE, April-June 2003, vol. 2, iss. 2, pp. 47 – 55, ISSN: 1536-1268.

Basic Principles of Design of an Autonomous System

Alexandra Šlezárová and Igor Hantuch
Faculty of Electrical Engineering and Information Slovak University of Technology,
Ilkovičova 3, 821 09
Bratislava, Slovak Republic
{alexandra.slezarova, igor.hantuch}@stuba.sk

Abstract— **This article describes the basic steps to create autopilot. States what action should follow, what inputs and outputs will contain. It describes the basic proposals for the regulator. It also describes the disturbances, that may arise in different situations and how to avoid, or that it should be able to react on these disturbances. As the future we will also regulate the amount - the location, it can select the appropriate device. The article describes the basic of GPS.**

Keywords - autopilot, GPS, regulator, intup, output, pitch, roll and yaw axis

I. Introduction

History [1] of the first RC (radio controlled) model dates back in 1893. RC model was invented by Nikola Tesla. His first model was not an aircraft but a submarine. He controlled it with radio waves. The first successful radio-controlled model was invented in 1917. And during the World War II, Germans controled weapons using radios. Since the first flight in 1917, the radio controlled model aircraft have been developing and few years later were developed some improvements. Aviation, in its early days, required flying pilot`s permanent concentration to flying safely. Long flying required permanent concentration, which leaded to fatigue. Therefore they tried to develop an autopilot to perform certain tasks instead of the pilot.

The first ship, which used the autopilot, was Standard Oil tanker JA Moffet in 1920. The first aircraft with autopilot was invented by Sperry Corporation, managed by Elmer Sperry in 1912. About two years later, his son Lawrence Sperry, convinced the audience so that he flew over them, without the hands controlled the aircraft.

Nowadays, autopilot is mainly used for long lines, when the aircrafts are set to direct flight in a horizontal course, if the situation do not requires close attention of the pilot, and thereby it is reducing pilot`s workload.

II. DESCRIPTION OF THE AUTONOMOUS SYSTEM

The concept of autonomy [2], in the case of robot, is defined as a system, which for a longer period can independently carry out tasks and where no human intervention is necessary. To characterize the robot as autonomous, it should satisfy certain properties:

- collecting and processing information about their surroundings in real time,
- prolonged work without human intervention,
- shifting ability in space,
- also avoid situations, objects, which could lead to human injury, and damage of buildings or the robot itself.

As we already know, the concept of autonomous pilot - an autopilot can be characterized as a process that responds to certain changes in the environment. Application of autopilot can be used for different devices. Devices are divided into two categories: stable and unstable. How the stable passive devices, we could characterize the robots, which are firmly on the ground. They are such a small ground mobile robots. Unstable category includes aircraft, helicopters and all devices that do not have a stable base. Autopilots are used in the case of flight, when it is set the exact route and position.

III. TRENDS IN THE USE OF AUTONOMOUS SYSTEMS

This process begins to be use increasingly in more sectors. Not only in industrial (robots, aircraft), but also in other institutions. They are used for example in economics, households, health care and in others. It is used primarily to facilitate and speed up work in industry and households. The main reason is that autonomous systems are used in environments, which are unhealthy for human bodies or in places, where human access is not possible.

IV. THE USE OF AUTONOMOUS SYSTEM

They are mostly used in the military sphere, where the equipment is programmed with an autonomous system for small mobile robots. In the military, we observe the greatest long-term progress of autonomous systems development and subsequent use in various military systems. These robots are used for retrieval, disposal, information gathering, transfer and delivery of materials and so on. Furthermore, it is used in the aircraft as autopilot. Currently, we are experimenting with the autopilot in the RC (Radio Controlled) models. Autopilot can control parameters such as height, speed, direction, location and more.

V. BASIC PRINCIPLES OF FLIGHT

The most fundamental principle is the principle of flight method [3]. Principle of flight includes two main concepts, and those are the aerodynamics and flight mechanics. Aerodynamics deals with the movement of gases, their effects on the bodies when they are floating through. A flight mechanics of aircraft includes the laws of motion. It is very important because there are many types of flight, where the various flying forces effect on devices.

For example:

- aerodynamic forces,
- physical strength,
- inertia and centrifugal forces,
- tensile strength.

This is all we should know and deliberate, when we are going to design the autopilot.

VI. DISORDERS OF INPUTS

In our paper we will now pay attention on the autonomous system of aircraft. In respect of fault inputs operating on our system, as we mentioned in the introduction, fault input can be weather conditions, especially wind flow as an impact on the system, but it may also be other objects which have to be identify by the system and if it is necessary they should change and pre-programmed the route. The deflection of route is the result of bad weather conditions or failure of communication between devices. Failure of communication evocates information dropout. These information plane uses to identify and monitor all the information necessary for safe flight. To prevent from hazards, we are trying to design quality autopilots. For example an autopilot, which is specifies on the high compensation. Furthermore, the system must be designed in the way to be able to self-control the aircraft without an accident also in the case, when is the communication lost.

VII. IDENTIFICATION SYSTEMS

To enable us to design an autopilot [3] for the aircraft we should consider all the details of aircraft. It means, we should record inputs and outputs, whic are needed to obtain the model. Seeing that it flies in the three dimensional environment, identification is difficult. Thus, there appear to us the x, y and z-axis. So the movement of aircraft in the space we are describing by 3 - ch axes, acting on the side upright. The centre of these axes is the centre of aircraft centroid. Through them we can examine each movement. Axes:

- x is the longitudinal axis of aircraft – it is located in the symmetry plane and it has suitably chosen direction
- y is the lateral axis of aircraft - is perpendicular to the symmetry plane
- z axis is perpendicular axis of plane – it lies in the symmetry plane perpendicular to the x axis

X-axis (longitudinal) identifies the flight speed, namely:

- longitudinal vx,
- lateral vy,
- perpendicular vz.

Y axis defines lateral breakaway and z axis defines ascending and descending. Rotating motion around the longitudinal x axis is called the pitching x, around the side of the y-axis, it is called luffing and around the perpendicular axis, it is called cornering.

Each of these axes has its specific properties. The best would be to design their own controller for each of them, so we can control their properties.

Quality regulator can be design using various methods. We can choose any method according to parameters that we know. At first, we should identify the model. Identification is a process by which we can initiate the experiment on the basis of the input-output information or and we get a dynamic process. Processes can be controlled by the control signal.

Kind of Methods:

Analytical methods or Experimetal methods

Analytical: Naslin`s Method
Method of Placing Poles

Experimental: Ziegler-Nichols`s Method
Strejc`s Method
Method of Direct Synthesis
Cohen-Coon`s Method
Method IMC
Haalman`s Method
Chien-n-Hrones
Reswicka`s Method
Smith-Murrill`s Method

For example: if we want to determine a PID regulator by an experimental method for example for measuring the height, we can use the method of Ziegler-Nichols. We can do this, when we activate in the system the greatest instability.

VIII. PRACTICAL DESIGN OF EQUIPMENT (SYSTEM)

The aircraft consists from the forequarters of aircraft [3], where the cabin is located, the centre of aircraft - fuselage and from the end of aircraft. Further are the wings, propeller, and landing gear. In the front of aircraft is located cabin as we mentioned before, where is located AC motor. Then there are servos (actuators) to control the elevator, rudder and flaps. The chassis is fully given. The apparatus from which we obtain data and which contains the aircraft are: accelerometer, gyro, altimeter, planimeter, speedometer, processor, camera controller, receiver, transmitter. All of these devices include avionics.

The accelerometer is a device that measures the non-gravitational acceleration. Further to this are used inertial properties of material objects. For the overall acceleration we need at least three accelerometers, because accelerometer measures acceleration only in one direction. We can recall that in the Space are accelerometer used to measure the pressure of solar radiation, environmental resistance, thrust engines.

By the gyroscope we keep the momentum conservation law. This means, if we are using it we measure and maintain the same orientation and direction. Flywheel is a major part of gyroscope. The outer frame is created by the flywheel attached on the axle and the axle is attached on the swivel joints. The flywheel body can move in all directions around its axis.

Altimeter may be use in air transport as a device for measuring the height of total Earth's surface (usually is used barometric altimeter based on pressure measurements in atmosphere). We can also use GPS for measuring the height. The best for for measuring height is probably a combination of both methods.

Control is performed by 2.4 GHz transmission of RC (Radio Controlled) station. This includes sending and receiving part. The receiver provides the reception of signal from transmitter and then instructs servomotor. On the transmitter there will be defined by all the channels necessary for the flight. Gradually we will replace it by small radio autonomous computer. At this time, the aircraft will obtain control by microprocessor, where will be the program to regulate the height.

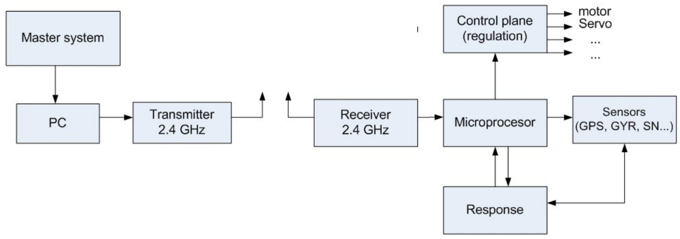


Figure 1. – the schem of the automatic autonomous system

IX. GPS – Global position system

To enable us to measure and then control the high in real-time, we need to choose a device that will be convenient for sending data to the PC. The PC will be recorded the high values and on the basis of these values it will prevent from the inclination from route. There are several devices that are provided these measurements. Of course it also depends on what high we want to measure. We can measure the water table, temperature, pressure, humidity and other amounts. But we need to measure the high somewhere in the Space. These measurements allows us to measure the GPS [4], altimeter, vario, pressure gauge and others.

We should describe GPS that we use to determine the exact location by three dimensions. To determine it, we need at least three satellites, one for each dimension. GPS is built on three basic segments. They are the space, control and user segments. What does it mean?

Space segment includes 21 active satellites and three backup satellites. Each satellite circle arount the Earth every 12 hours. Satellites circle around the orbits. Satellites circle around the orbits in 22.200 km. Orbit contains four satellites. It means, when we want to determine the correct position, four satellites are enough. Signals from the satellites can be received at all locations on the Earth in the frequencies 1575.42 MHz and 1227.6 MHz.

Management segment updates information contained in the satellite data messages. Five monitoring stations that create the management segment are placed around the earth along the equator. Using these stations are calculated ephemeris - orbits.

User segment mainly consists from users and GPS tools. GPS calculates preliminary route determination by received signals: position, velocity and time.

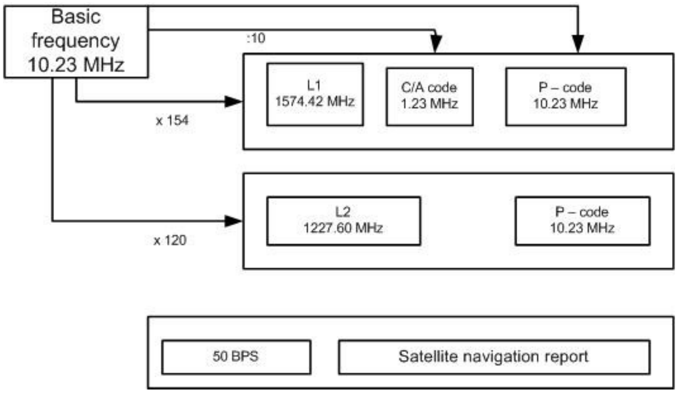


Figure 2. – frequency of satellite signals

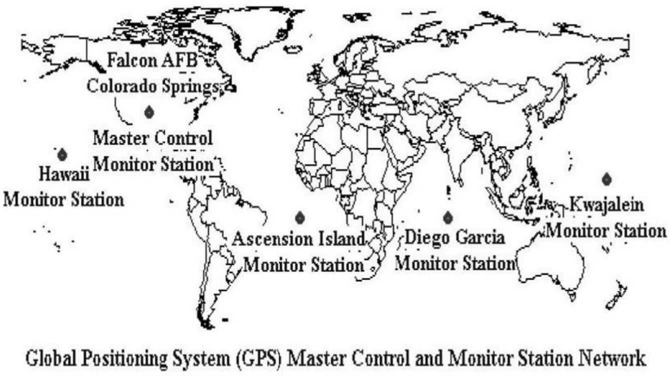


Figure 3. – location of the satellites in the world

X. Assessment

The paper is aimed on the independent control systems. Following a brief introduction we went directly to the centre of our research and that is the independent management and design process management system for aircraft. Nowadays, we have well defined the structure of system and partial system proposals (for example plane construction) are done. We will continue in our research with the proposals of optimum control system with microprocessor.

REFERENCES

[1] Infortmation about history of aircraft. [online]. Wikipedia [cit: 2010-06-15]. Online on internet. http://en.wikipedia.org/wiki/AutopilotJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] JONES, J.L., SEIGER, B.A., FLYNN, A. M. – Mobile robots, ISBN:1-56881-097-0, 1999K. Elissa, "Title of paper if known," unpublished.

[3] KELLER LADISLAV, Ing. – Učebnice pilota 2006 - Principy letu, 2006, ISBN – 80 – 85280 – 89 - 2Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[4] Steiner, I., Černý, J. - GPS od A do Z, 4. aktual.vyd. - Praha : eNav, 2006 ; obr., tab., fotogr., lit. - ISBN 80-239-7516-1.

Using Pololu‘s 3pi robot in the education process

Branislav Thurský
Alexander Dubcek University of Trenčín
Faculty of mechatronics
Trenčín, Slovakia
thursky@tnuni.sk

Gabriel Gašpar
Alexander Dubcek University of Trenčín
Faculty of mechatronics
Trenčín, Slovakia
gasparg@gmail.com

Abstract— **The paper presents a sample of usage of the Pololu‘s mobile robot 3pi at the interactive teaching of robotics. Since this is a mobile robot with a circular structure and two controlled wheels, it is mainly suitable for the role of following the line, movement in the maze and possibly after adjusting for robotic soccer. In our presented example of it's usage we are dealing with following lines forming a maze. Thinking of the robot can be demonstrated using described algorithm of finding a way in the maze. The described algorithm allowed primary school students in the hobby group better understand the challenges and problems associated with programming a mobile robot and to develop creative thinking of future robotic systems programmers.**

Keywords - component; formatting; robot, hardware, software, algorithm

I. INTRODUCTION

At the beginning we are introducing the design and parameters of 3pi robot. 3pi robot is designed so that it can be used in the tasks of following the line and finding the way out of maze. The maze is represented by perpendicular lines with a labeled target point. 3pi robot is a small size (diameter 9.5 cm, weight 83 g) and is powered by four AAA batteries, from which we obtain by using a unique energy system "boost-converter " voltage 9.25 V (for the engine used to power motors of the robot), regardless of the level of battery charge. Regulated voltage of 3pi robot allows to achieve speeds of up to 100 cm / sec, while achieving accurate revolutions because of constant voltage while discharging the battery.

3pi robot as mentioned has a circular shape, and is powered by two independently controlled propulsion units, consisting of DC motor, gear box and a wheel. In its heart is placed Atmel ATmega168 processor resp. ATmega328P which also provides for the management of motor units and thus allows the movement of 3pi robot. The key features of used processor Atmel ATmega328P are clock frequency 20 MHz, 32k of flash memory, 2 kilobytes of RAM and 1 kilobyte serial EEPROM memory, see Fig.1. [1]

Atmel corporation offers their Atmel AVR Studio development environment with integrated free GNU C/C++ compiler freely available for use with their micro-controllers on the internet. For the very 3pi robot is by the manufacturer prepared an extensive set of libraries designed to communicate with all the integrated hardware. This creates an excellent platform for people with experience in programming in C language for learning about the mysteries and problems solved in robotics. Or it provides a funny way to learn programming in C. In our case it was not different in our mechatronics hobby group at the Elementary School na Dolinách.

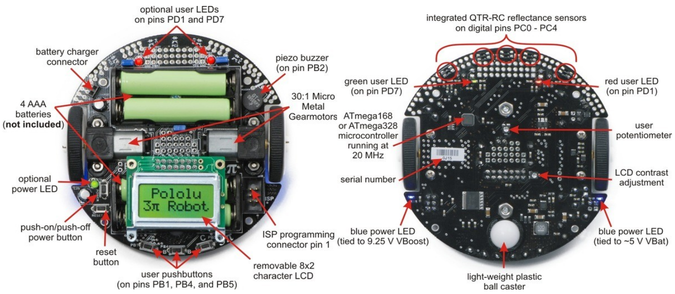


Fig.1 Hardware platform of 3pi robot

Although the hobby group bears the name „Programming of the Atmel 8051 microcontroller", but since it is the older predecessor of processor ATmega328P and as a programming language we used the C language, it was not a great problem to switch to this platform.

Another fact that made our work on this platform easier was the mentioned set of libraries and prepared sample programs. At one such example, I would like to draw attention in this article. This example deals with finding path out of the maze which is represented by a black line. The example due to its simplicity allowed students to dive into the secrets of robotics and by a playful way it delivered the problems to be overcome.

II. FINDING THE WAY HOME

To start, one could define the question: What is a maze? Maze can be: a place where it is easy to get lost, figuratively confusion, opaque place, building or garden built so that one can get easily lost.

In our case, since the construction of the robot is suited to follow the line, the maze will be represented by intersecting lines on a perpendicular angle (i.e. will be developed on a regular grid), creating a maze with its start and target. For simplicity maze must be designed so that there is only one path from start to target and also to avoid creating loops in which the robot might get stuck.

This simplification of the maze allows us to use a simple strategy to get through the maze and to find a way out. In the next stage we can remove the simplifications and find a strategy that can deal with this type of maze. For better illustration, see Fig. 2.

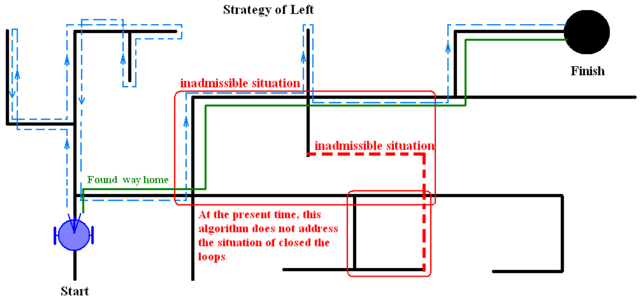


Fig.2 Maze and strategy of finding the target using turn right priority.

The maze is made for a better resolution by a black line representing the path on a white background. The chosen strategy must allow a search of the entire maze and then find the shortest path from start to target. In addition to finding an appropriate search strategy it is necessary to solve a task to remember the mapped part of maze and then search for the shortest possible distance from start to target (or to the point of it's current location, which is our case) [2].

III. Strategy

It is possible to develop more strategies to search the maze. All of them will achieve different results depending on the type of maze. It can not therefore be said, that it is possible to find a single strategy that would universally, for different types of mazes, led as quickly as possible to find the target.

We chose a strategy in which there is a priority to turn left or turn right and after the gradual maze search we can simply exclude the places in the maze where there is no target. What does it mean to have priority on the left? This means that if it is possible to turn left, we turn left. In case that it is not possible to turn left, we prefer to proceed straight over to the right, and if it is not possible to turn right, then we turn around and continue with this strategy. Similarly, it is the case of the priority right. In this case, eight situations can occur in the maze: left, right, left or right, intersection left-straight-right, intersection left-straight, intersection straight-right, end of the road and end of the maze. Fig.3

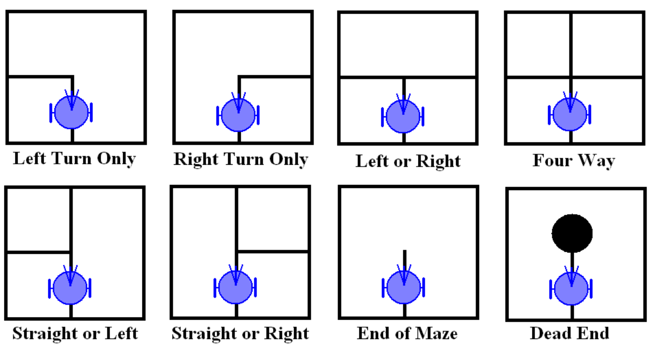


Fig.3 Situations which may occur in a maze

In order to be able to recognize situations, 3pi robot is equipped with infrared sensors that are able to detect the given situation. If the sensor is on the line, its value equals 1, otherwise its value is equal to 0. Sensors are placed on the forehead of the robot in such distance, so that we are able to detect the line and intersections in our maze. Their location is shown in Fig.1. By the combination of switching individual infrared sensors, we can evaluate the current situation of the robot in the maze and depending on this, to issue a control signal. Some situations are illustrated in Fig.4.

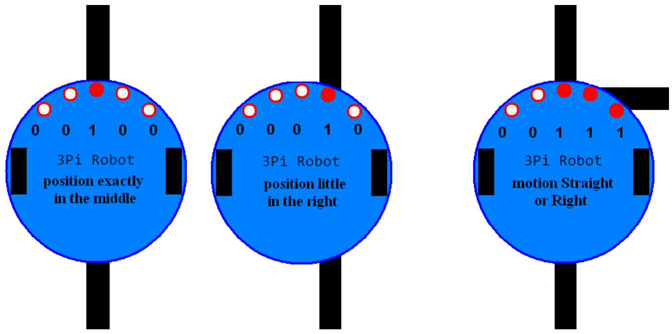


Fig.4 Infrared sensors state in different situations in the maze.

By the signals received from sensors, the robot knows what the situation is and according to our chosen strategy decides how to continue in the given situation. At the T-type intersection the robot would get 1's from all sensors. As can be seen in the figures our robot is equipped with five infrared sensors, and thus one can get 2^5 which is 32 possible combinations, some of which for our type of the maze are unlikely (for example, unacceptable are combinations such as 10101, 10001, 11101, etc.). Our control and strategy is based on the following arguments:

1. In case the state from infrared sensors is 00100, go straight at the full speed.
2. If you reach state 00110 turn slightly left to state 00100.
3. If you reach intersection (00111 „R“, 11100 „L“or 11111 „T“) the chosen strategy decides about the movement.
4. If the state is 00000 you are at the end of line. Turn around to 00100 and continue straight.
5. If you get into state 11111, you are at the target.

IV. Finding the Shortest Way to the Target

To explain how to find the shortest path, we define some basic concepts that we will use. One is the intersection that represents the place in which the robot has more than one choice of direction. Fig.3 shows eight situations, four of which can be regarded as intersections. The first two situations can not be intersections, because the robot has no choice but to turn. In the last picture is shown the situation where the robot has to turn around because of finding the end of the road. The picture therefore shows only four intersections to be detected in the process of searching the maze. It remains to be clarified how the robot determines the type of intersection. For example, if the robot moves on the line the state of its infrared sensors will be 00100 and at the transition to 00111 we could conclude that it is a turn right or an intersection right and straight. It is necessary to make one more step, in which we confirm the given situation we are at. In the case we get sensors state 00100 at the step forward, it is a right-straight intersection, if the state 00000, it was a turn right. Similar situations occur also at the left-straight intersection and the "T" and "+" intersection. As soon as we get information on the type of situation where we are at, the strategy decides about the behavior in the given situation.

To represent the behavior of the robot at the given situation in its memory, we represent every situation by appropriate symbol. If the robot based on the strategy will go straight we store „S-straight“ into the memory, if it turns right „R-right“ and if it turns left „L-left“ and eventually when it turns around we will store "U-turn into the memory. If there is a situation where the robot performs on the basis of the strategy decision on next move, it stores its decision into memory as an appropriate symbol. Sequence of symbols creates information about the passed path in the memory. Once we know how the robot moves in the maze, we can remember the way that the robot passed. We can proceed to excluding the paths which do not lead to the target. In this case, the most interesting situation is when the robot must turn around ("U-turn") due to the impossibility of further progress in its way. This also means that the previous decision was incorrect and has to be changed. So if we find right-straight intersection and we go straight based on the strategy left, (we store symbol "S") and then we have to turn around "U", in the memory will be stored "SU". When we return to the previous decision we make a change (we do not go straight, but in this case we turn to the left "L" because of the priority in the strategy to the left). By doing this we get a situation "SUL" in the memory, which means that at the intersection we should not continue straight, but we should turned right. The situation in the memory is to be resolved by replacing the combination of "SUL" by symbol "R" and we continue until we find the next intersection, at which we will make decision again and in case of bad decision we correct it again. The situation when the robot has no choice but to turn is not considered, as stated as an intersection. Therefore this was not written into memory. After hitting the end of path and turning around, the robot goes back to the place of decision (intersection) where the following situations may occur leading to a correction of the earlier decision. In the case of a combination „LUL“we obtain „S“, see Fig.5.

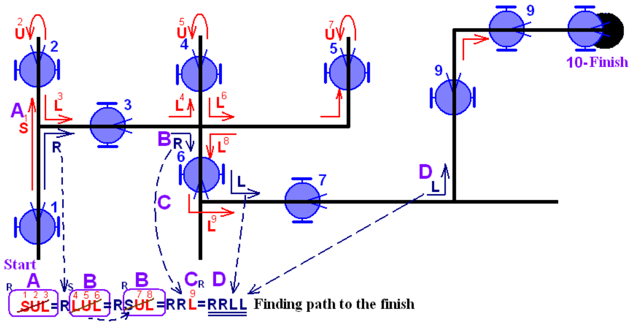


Fig.5Applying the path search using strategy with priority to the left.

As shown in Fig.5, during maze searching it is also possible using simple rules find the way to the target. It must be noted that our maze is simplified and we are not dealing with finding the shortest path in any way. Our maze may not have loops and thus has only one path to the target to be found. It is therefore important to find such search mechanism, which allows searching of the entire maze and find its way to the target. This search method is easy to implement into the platform of 3pi robot and it can be practically tested in real conditions. This example is a part of the sample examples demonstrating the properties of 3pi robot.

In the group were students aged 12-15 years working in couples. The main idea of the group was to introduce ways of creating control algorithms designed for specific hardware to students. Student had to understand the controlled hardware for which the control algorithm had to be developed. The goal of this group was not to develop their own solutions, but to teach students to acquire and use the necessary knowledge to build custom robotic systems. The group meetings were held once a week and lasted for three hours during school-year. In this time students managed to acquire software and hardware skills in way they learned to test sample programs, modify them and add their own ideas. Based on experience in teaching this group could be said, that the given platform of mobile robot equipped with these software equipment is fully suitable for use in the education in robotics. It develops theoretical analytical thinking as well as practical aspects in the construction of a mobile robotic system. The system is fully proved in our group sessions and fulfilled our expectations. We strongly recommend it for the education in robotics.

Conclusion

This example illustrates how easily we are able to solve seemingly complex problems such as path finding in a maze. Students in the hobby group not only were able to understand and operationalize the maze search algorithm, but to modify and expand it with their knowledge. After mastering this algorithm they expressed interest in the new challenging tasks. 3pi robot platform allows extension of the universal printed circuit board, thus giving the possibility of extending the custom peripheral expansion. This creates the possibility of further extension of the hardware or sensor systems that enable solutions to a variety of other tasks. Some students began to think of their own design of robots that allows them to better solve their proposed tasks.

References

[1] *Pololu 3pi Robot User's Guide*. ©2001–2009 Pololu Corporation. Available at: < http://www.pololu.com/docs/0J21>.

[2] VANNOY R. T. *Line Maze Algorithm*. Available at: < http://www.pololu.com/file/0J195/ line-maze-algorithm.pdf>.

Proceedings of the 1st International Conference on Robotics in Education, Bratislava, Slovakia, Sept. 16 - 17, 2010

How to attract children to mechatronics

The experiences with mechatronic after school classes

Ing. Renata Janošcová, PhD.
Alexander Dubček university of Trenčín
Trenčín, Slovakia
janoscova@tnuni.sk

Abstract - **The tool of technical disciplines popularization for children can be free-time activity – after school classes for schoolchild of primary schools from 5th to 9th class. Schoolchild ponders the choice of secondary school although they have not concrete opinion on their future work. Suitable support of their interest to technical disciplines by playful form in after school classes can distinctly influence their following drift. If the schoolchild will chose technical oriented secondary school so then will chose technical faculty of university probably. Motivation of schoolchild to technical fields is one of the primary goal neither of this conference nor complete activity of mechatronics after school classes that are realized in Trenčín and its vicinity from 2003.**

Key words - Mechatronic; MC - Mechatronic after school classes; building set Lego MINDSTORMS; programming language NQC - Not Quite C, ADuT – Alexander Dubček university of Trenčín, LMS – Learning Management System.

I. Motive

Mechatronics is a new field of science and technology. She deals with the development and use of machines and systems with computer control [20]. It is based on knowledge of mechanics, electronics and microprocessor technology, information and computer control . The word *"mechatronics"* is for amateurs unknown. The faculty of mechatronics as part of ADuT exists from 1997. In spite of it children of primary schools don't know what this term means, someone thought that means sometimes prehistoric. Nobody was near to right answer. At this time the Slovak technical faculties feel interest decrease of study. This reality brought us to belief that right time is for change [4].

II. Primal impuls

No activity can run without primal impulse and enthusiasm of people. This is our case too. Others possibilities were positive too. One of them were no-useable LEGO MINDSTORMS building sets in The primary school at Na dolinách street in Trenčín – Zlatovce and active parents council that had demand on technically oriented out-school activities for their children [6].

The 2nd one positive factor were the students of mechatronic engineering who suggested leading of groups and processing of school-year-work in Lego MINDSTORMS models programming in NQC theme.

And still one factor is here – expert teachers of faculty who had courage to do collaboration of type: university – primary schools. When the right people meet right people new quality can be bored. If impulse on lower organizational level arose it doesn't disappear when it is supported by leading management level. The rector was on the ADuT side, director and his assistant were on the side of primary school. After two meetings the agreement of collaboration was arose. Then co-ordinator for all group activities was named. Then LMS support for groups was appointed together with period and periodicity. For each group the suggestion for teaching documentation was created with graduate profile, teaching plan, organization of study, teaching points and time-thematic plan [2] (the demonstration of teaching documentation part for group "Programming in NQC language" - The mechatronics basis II., in Tab.1).

III. Power of reciprocity

In September of 2003 the children of two primary schools were acquainted that new "*Mechatronic after school classes*" are opened, oriented on mechatronics popularization through robotic building set LM programming. Parents of children were acquainted. The final models attracted children the models were created by students of The faculty of mechatronics. The interest of children was enormous [7].

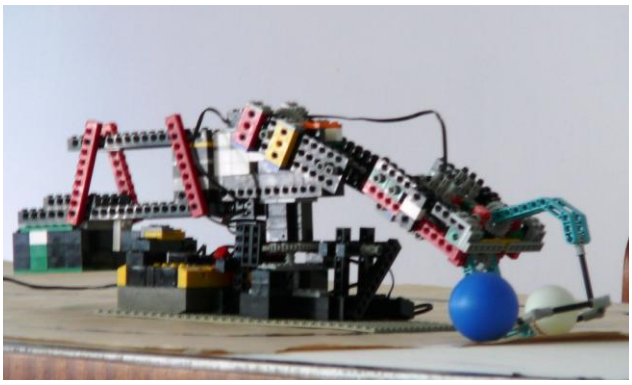


Figure 1. Manipulator model [1]

Paper was prepared with the support of the project KEGA 3/6411/08

TABLE I. EXAMPLE OF TIME-THEMATIC PLAN

n	T.	Theme title	Teaching method	Teaching aids and didactic technology
1	1.	Safety instrucions, Acquainting with Learning Management System	Frontal lecture	Lego MINDSTORMS Dacta, RoboLab, PC, Internet – http://elearning.tnuni.sk/moodle/course/category.php?id=11
2	2.	Acquaint with RoboLab , Lego MINDSTORMS Dacta, own building	Frontal lecture, discussion	Building set LEGO MINDSTORMS Dacta
3	3.	Lego MINDSTORMS Dacta, own building	discussion	Building set LEGO MINDSTORMS Dacta
4	3.	Programming environment RoboLab, basis of work with RCX a motors	Frontal lecture, discussion	PC, programming environment RoboLab
5	4.	Programming environment RoboLab, basis of work with sensors	Frontal lecture, discussion	PC, programming environment RoboLab
6	5. – 6.	Programming environment BrixCC, jazyk NQC, basis of work with motors and sensors	Frontal lecture, discussion	Building set LEGO MINDSTORMS Dacta RoboLab, PC, prog. environment BrixCC
7	7.	Robot that can stay on table (photically sensor using)	Frontal lecture, discussion	Building set LEGO MINDSTORMS Dacta RoboLab, PC, prog. environment BrixCC
8	8.	Preparation for *Robotic day ofTrenčín*	Team work	Building set LEGO MINDSTORMS Dacta RoboLab, PC, prog. environment BrixCC
9	9.	Robot that can stay on table (tactile sensor using)	Frontal verification of knowledge	PC, prog. environment BrixCC
10	10. – 12.	Preparation for *ISTROBOT* match	Team work	PC, prog. environment BrixCC
11	13.	Robot that can know colours (photically sensor using)	Frontal lecture, discussion	Building set LEGO MINDSTORMS Dacta RoboLab, PC, prog. environment BrixCC
12	14.	Robot that can play melody (photically sensor using)	Frontal lecture, discussion	Building set LEGO MINDSTORMS Dacta RoboLab, PC, prog. environment BrixCC
13	15.	Using of RCX and positional sensor	Frontal lecture, discussion	Building set LEGO MINDSTORMS Dacta RoboLab, PC, prog. environment BrixCC
14	16.	Robot on remote control	Frontal lecture, discussion	Building set RoboLab, PC, prog. environment BrixCC
15	17. – 19.	Self-reliant project of group no. 1	Team work	Building set LEGO MINDSTORMS Dacta, PC, prog. environment BrixCC
16	20.	Self-reliant project of group no. 2	Team work	Building set LEGO MINDSTORMS Dacta, PC, prog. environment BrixCC
17	21.	Self-reliant project of group no. 3	Team work	Building set LEGO MINDSTORMS Dacta, PC, prog. environment BrixCC
18	22.	Terminatively repeating – presentations preparing	Discussion	PC, prog. environment BrixCC
19	23.	*Presentation of mechatronic after school classes*	Presenation	Building set LEGO MINDSTORMS Dacta Dataprojektor
20	24. – 26.	*Open university* conference preparing	Team work	Building set LEGO MINDSTORMS Dacta RoboLab, PC,

IV. COURSE

In academic year 2003/2004 the 1st two mechatronic after school classes were realized, these were focused on work with building set system Lego MINDSTORM, *RoboLab* system and *NQC* language in *Bricxcc* environment. The groups were leaded by students of FM. The children met their teachers from 2 to 4 hours per week during winter and summer semester. Their were interested in mechatronics basics by way of building set Lego MINDSTORM, RoboLab system and NQC in accordance with applicative time-thematic plan from paced pedagogic material. Single groups were divided into three categories in accordance with sophistication:

The basics of mechatronics I., II., III. For example when some student wanted go to group that related under The basics of mechatronics II., firstly he must graduate course I. of these thematic category. The two of teachers paced and edited the guide that was the suitable aid for next teachers [19].

In the following content is demonstration of the code that children learned during 5th and 6th week of teaching (TABLE I. :

```
// Basis of work with motor,  bulbs, audio, forward and reverse

task main()                      // basic task
{
 while(true)                     // non-ending cycle
  {
  start Blikac_Vpred;            // calling task Blikac_Vpred…
  start Motor_Vpred;
  start Hudba_Vpred;
  Wait(600);                     // stops running the program for 6 second
  stop Blikac_Vpred;             // stops task Blikac_Vpred…
  stop Motor_Vpred;
  stop Hudba_Vpred;
  Off(OUT_A);                    // output off (A – motor)
  Off(OUT_B);                    // output off (B – bulb)
  Wait(250);                     // stops running the program for 2,5 second
  start Blikac_Vzad;             // calling task Blikac_Vzad…
  start Motor_Vzad;
  start Hudba_Vzad;
  Wait(600);                     // stops running the program for 6 second
  stop Blikac_Vzad;              // stops task Blikac_Vzad…
  stop Motor_Vzad;
  stop Hudba_Vzad;
  Off(OUT_A);                    // output off
  Off(OUT_B);
  Wait(250);                     // stops running the program for 2,5 second
  }
}
/*-------------------------------------------------------------------------*/
task Blikac_Vpred()               // flashing lamps B
{
 SetPower(OUT_B,7);              // setting output  B the maximum
 while(true)                     // non-ending cycle
  {
  OnFwd(OUT_B);                  // lamp on
  Wait(5);                       // pause 0,05 second
  Off(OUT_B);                    // lamp off
  Wait(5);                       // pause 0,05 second
  }
}
```

```
/*------------------------------------------------------------------------------------------*/
task Blikac_Vzad()                  // flashing lamps C - slow
{
 SetPower(OUT_C,7);                 // setting output  C the maximum
 while(true)                        // non-ending cycle
  {
  OnFwd(OUT_C);                     // lamp on
  Wait(50);                         // pause 0,5 second
  Off(OUT_C);                       // lamp off
  Wait(50);                         // pause 0,5 second
  }
}
/*---------------------------------------------------------------------------------------------*/
task Motor_Vpred()                  // move the engine forward
{
 SetPower(OUT_A,7);                 // setting output  A the maximum
 OnFwd(OUT_A);                      // forward movement
}
/*----------------------------------------------------------------------------*/
task Motor_Vzad()                   // move the engine rearwards
{
 SetPower(OUT_A,1);                 // setting output  A the 1
 OnRev(OUT_A);                      // backward
}
/*----------------------------------------------------------------------------*/
task Hudba_Vpred()                  // plays a melody when moving forward
{
 while(true)                        // non-ending cycle
  {
  PlayTone(262,40);  Wait(50);      // tone plays 0,4 seconds and waiting 0,5 sec.
  PlayTone(294,40);  Wait(50);
  PlayTone(330,40);  Wait(50);
  PlayTone(294,40);  Wait(50);
  }
}
/*----------------------------------------------------------------------------*/
task Hudba_Vzad()                   // plays a tone when moving backward
{
 while(true)                        // non-ending cycle
  {
  PlayTone(1000,40);  Wait(100);    // tone plays 10 seconds  and waiting 1 sec.
  }
}
/*----------------------------------------------------------------------------*/
```

At the end of school year the public *„Presentation of members activities of mechatronic after school classes“* was realized.

Each group presented its all-year work. Children showed the practical illustrations of their models, they informed about models creation and programming [8].

After successful presentation the superfine students received certificate, which encompass information of successful Mechatronic after school classes graduating in range of *The basics of mechatronics I., II., III.*, duration in hours (minimum is 72 hours), the list of teaching themes, the name of teacher. Certificate was signed by the rector and dean of The faculty of mechatronics and children obtained them from hands that the big experience was for them. The holders of certificates received bonus – the entry to the university computer laboratories under view of faculty teachers. This was used by children mostly for access on internet during all academic year and mainly during holidays. Similarly these activities run during others years 2004-2009.

At the beginning of school years 2007, 2008 and 2009 *OPEN UNIVERSITY [10]* conference was organized (Fig.2).

Figure 2. Receptive auditorium of Open university conference 2009

The goal of this conference based on students interest encouraging for science and technics, mechatronics popularization, learned knowledge and skills presentation and robotic equipment exposition united with match. There, on conference, the contributions of popular mechatronics were presented. Other participants presented their experiences with groups leading oriented to technical fields. Selected mechatronics groups members presented their contribution, sometimes in English too.

Figure 3. Conference organizational board – mechatronic after school classes members

Children even were in workshop board (Fig.3).

V. OUTCOMES

Between other activities of mechatronic qroups - except periodical meetings – we can integrate active presence on activities:

- *Robotic day of Trenčín* (racing exposition of robots, organized by secondary school Stredná odborná škola Trenčín, in February[1]);
- *ISTROBOT* (robotic match that is organized by Faculty of electrical and computer engineering of Slovak technical university in Bratislava, in April[2]);
- *Presentation of mechatronic after school classes activities* (it is presented that what children have learned during school year, organizer is co-ordinator of mechatronics groups activities in co-operation with The faculty of mechatronics ADuT, in June[3]) [5];
- *Open university* conference – forum on academic domain ADuT about youth motivation to technical sciences, about experiences with work in this area; organizer is co-ordinator of mechatronics groups activities in co-operation with Faculty of mechatronics ADuT [4], in September [4];
- *Work-shop LMS [9]* (teaching for teachers with the most modern education methods using and e-learnig; organizer is co-ordinator mechatronics groups activities; in November[5]) [3].

At the present information about mechatronics groups are publish on portal http://www.mk.tnuni.sk, which contains rich photo-documentation of activity in section Galéria (Gallery).

VI. OVERALL VIEW

From 2003 to 2009 were groups realized which were oriented to:

1. The mechatronics basis I.
   - Models building with using Lego MINDSTORMS [18];
   - RoboLab;
2. The mechatronics basis II.
   - Programming in NQC language [19];
   - Basis of programming in C language;
   - Basis of programming in C++ language [12];
   - Basis of programming in OOP Visual Studio.NET – C#;
   - Basis of programming in OS LINUX;
   - Basis of programming – OOP language PYTHON [13];
3. The mechatronics basis III.
   - Progamming of robotic systems in Assembler [17];
   - Control system SIEMENS LOGO! [14].

Until 2009, 114 certificates were handed about mechatronic after school classes graduating. Certificates were nominated by teachers in compliance of attendance and doings. Some children stopped their groups attending but their number was from 0 to 2 in every group.

Fig.4 illustrates number organizations connected to mechatronic after school classes activities according to types.

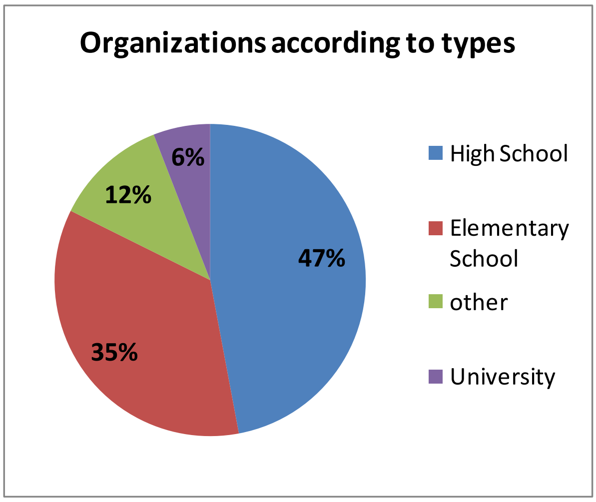


Figure 4. The number organizations according to types

Until 2009, 17 schools and organizations were connected in activities in active or passive approach.

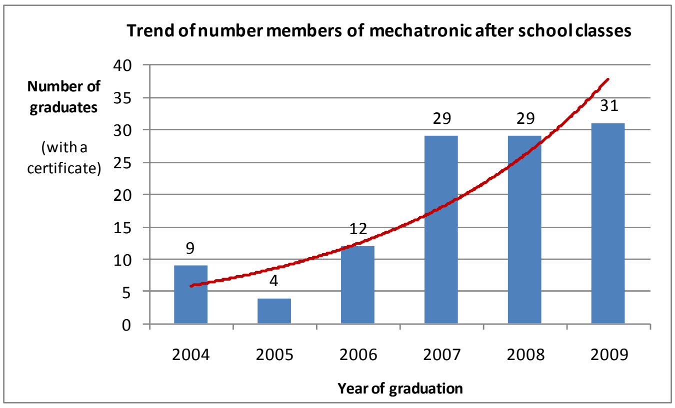


Figure 5. Trend of number graduates

Fig.5 illustrates increase number members of mechatronic after school classes which were nominated by their teachers for certificate receiving.

[1] Information: http://www.trencianskyrobotickyden.sk/
[2] Information: http://www.robotika.sk/maine.php
[3] Photos: http://mk.tnuni.sk/?page_id=54&album=1&gallery=9
[4] Information: http://mk.tnuni.sk/?page_id=16
[5] Photos http://mk.tnuni.sk/?page_id=54&album=1&gallery=14

VII. Feedback

At the present the first graduates are students of technical faculties of Slovak University of Technology in Bratislava, Masaryk university in Brno, ADuT and other universities.

Here are some of the observations and views of former members of mini - questionnaire MC - "mechatronic after school classes":

Jakub Káčer, completed MC: NQC programming language I. and II. Student Central Technical College in Nové Mesto nad Váhom.

Jaroslav Dzurilla, completed MC: Introduction to Programming (ANSI C) Object - Oriented Programming Visual Studio. NET - C #, Programming in Linux. The faculty of mechatronics Alexander Dubček university of Trenčín.

Michal Bystrický, completed MC: Introduction to Programming (ANSI C) Object - Oriented Programming Visual Studio. NET - C #, Programming in LINUX OS. Student Faculty of Informatics and Information (FIIT) Technologies Slovak University of Technology in Bratislava (STU).

1st How many years have you been a member of "mechatronic after school classes" and where did you study?

.. Two years. (Jakub Káčer)

.. Three years. I am a student at university in the field of mechatronics - 3rd grade. (Jaroslav Dzurilla)

.. Three years. I still FIIT, I'm going into the third grade (Michal Bystrický)

2nd To what extent has it affected you in deciding where the high schools, universities? *(Rate scale of 0-5, 0-no effect on me, 5 – completely influenced me)*

.. I would give three points. Total impact on me. (Jakub Káčer)

.. Partly influenced me. So I put three points. (Jaroslav Dzurilla)

.. I always knew that I would study IT, therefore I will give 0 points. (Michal Bystrický)3. Pomohlo Ti niečo z krúžku pri štúdiu na SŠ, VŠ ?

3rd Did something of "mechatronic after school classes" help to you in the study of the high schools, universities?

.. helpful and fairly. (Jakub Káčer)

.. I went through the MC in the "image" and I learned interesting things for me which I would otherwise not receive in high school. (Jaroslav Dzurilla)

.. yes definitely, I'd probably pick a "sense of programming" (programming thinking). (Michal Bystrický)

4th What was for you to "mechatronic after school classes" too difficult, and what you did not understand what you were bored?

.. nothing has bored, everything was great .. (Jakub Káčer)

.. hm ... I do not know at what level it is there now ... but I liked it ... (Jaroslav Dzurilla)

.. Rather, I enjoyed procedural programming. Object - oriented programming power of explanation by elementary school students. I remember I was struggling with the ring Constructors and destructors. (Michal Bystrický)

5th What change would you on the "mechatronic after school classes"?

.. Of course I would be more happy if they were available and modern versions of Lego dacta ... (Jakub Káčer)

.. I think that the results depend on the MC as an individual approach to students and teacher. (Jaroslav Dzurilla)

.. missing my leader, which showed us the way - work, work and more time to work on yourself. I think it must change to build better quality people from schools. Set a goal and go for it and actually implement it. To teach people to look in the API and Google. Regarding hardware, I would certainly start to work on the principle of what and how it works. Not Lego, but the board, and mini-solder the solder components.

I am studying the embedded systems and missing me a lot of the principles, I got behind the electronics, electrical engineering, logic circuits, and I do not know everything, but I would not know the real solder and construct a 486, and it is a mistake.

And this greeting students of Faculty of Electrical Engineering and Information Technology (FEI) STU, who have this on examination ☺). (Michal Bystrický)

VIII. Conclusion

We can write that the word "mechatronics" has become well-known in Trenčín and its vicinity. Our experiences from groups realizing confirmed that schoolchild motivation was rose and positive attitude was created to technical sciences, primarily to mechatronics.

It can seem our goal is complete and we can end – but not! After short pause we must receive new power, new inspiration, find new enthusiasm and continue in this and similar activities. Although the graduates of mechatronics groups will be not students of ADuT, with the highest probability they will be students of some of technical faculties, not only on Slovakia and this is our primary goal.

This activity is very hard from time and organization point of view. In spite of this costs that are put into it will be turned back in form of potential students of universities technical faculties. In the future we will greet integration this kind of activity into boring town conception for work with children and youth.

We want to thank to Faculty of electrical and computer engineering of Slovak technical university in Bratislava for *ISTROBOT* match realizing. It was for us big inspiration and motivation, it supported us in thought that our steps walk on right way. We thank to members of Tatran Team too who lead groups at the present for schoolchildren in laboratories of secondary school directly. And thank for this conference too that is taking statute of *Open university* conference in the right time.

REFERENCES

[1] Balucha, D. 2008. *WALKER 08*. In *Otvorená univerzita 2008* : Proceedings of the 2nd conference. Trenčín: ADuT, 2008.

[2] Janošcová, R. 2005. *Mechatronics after school classes*. In *Mechatronika 2005* : Proceedings of 8th International symposium on mechatronics. Trenčín: ADuT, 2005. ISBN 80-8075-058-0. pp. 54-60.

[3] Janošcová, R. 2006. *Skúsenosti s využívaním E-vzdelávania v predmete Manažérske informačné systémy.* In Schola 2006 : *Kvalita výchovy a vzdelávania.* 7th International Scientific Conference KIPP. Bratislava: STU, 2006. ISBN 80-227-2389-4. pp. 135-140.

[4] Janošcová, R. 2007. *Popularizácia vedy a výskumu formou mechatronických krúžkov*. In *Otvorená univerzita 2007* : Proceedings of the first conference. Trenčín: ADuT, 2007. ISBN 978-80-8075-248-4. pp. 32-35.

[5] Janošcová, R. 2007. *Využitie voľnočasových aktivít na propagovanie mechatroniky.* In *Využitie IKT vo vyučovacích predmetoch.* Proceedings of the first videoconference FIT. Trenčín: MPC v Trenčíne, 2007. ISBN 978-80-89319-01-5.

[6] Janošcová, R. 2007. *Mechatronické krúžky ako motivačný nástroj*. Mechatronické krúžky © 2007, BIOSDESIGN.EU [PDF]. Dostupné na **http://www.mk.tnuni.sk/files/ako_sme_zacinali.pdf**.

[7] Janošcová, R. 2008. *Päť rokov činnosti mechatronických krúžkov 2003 – 2008*. In *Tnu – Trendy*. Trenčín: ADuT, 2008.

[8] Janošcová, R. 2008. *Predstavenie mechatronických krúžkov*. In *SEPEM 2008* : 4th workshop teaching mechatronics. ISBN 978-80-8075-266-8. pp. 55-63.

[9] Janošcová, R. 2008. *Činnosť mechatronických krúžkov a využitie LMS*, In *Otvorená univerzita 2008* : Proceedings of the 2nd conference. Trenčín: ADuT, 2008.

[10] Janošcová, R. 2009. *O čom bola konferencia Otvorená univerzita*. In *Tnu – Trendy*. Trenčín: ADuT, 2009.

[11] Majerík, M., Ešše, M. 2008. *Process Model for conceptual Design of mechatronic Product.*. In *AiM 2008 Proceedinngs of 3rd International Conference*. Trenčín: ADuT, 2008. ISBN 978-80-8075-358-0. pp. 77-79.

[12] Šimovič, V. 2007. *Krúžok programovania v C++ pre žiakov ZŠ*. In *Využitie IKT vo vyučovacích predmetoch*. Proceedings of the first videoconference FIT. Trenčín: MPC v Trenčíne, 2007. ISBN 978-80-89319-01-5.

[13] Šimovič, V. 2008. *Krúžok základy programovania II.: Programovanie v jazyku Python.* In *Otvorená univerzita 2008*: Proceedings of the 2nd conference. Trenčín: ADuT, 2008.

[14] Thurský, B. 2006.*Teaching Logo! Siemens at Primary School*. In *AiM 2006 Ist International Conference Advances in Mechatronics*. Trenčín: ADuT, 2006. ISBN 80-8075-112-9. pp. 274-275.

[15] Thurský, B. 2007. *Mechatronical approach by making constructions of walking Robots* [online]. In: *Cybernetics letter: Informatics, Cybernetics and Robotics.* ISSN 1802-3525.

[16] Thurský, B. - Sabo, P. 2007. *Činnosť mechatronického krúžku LOGO! a jeho dosiahnuté výsledky.* In *Otvorená univerzita 2007 :* Proceedings of the first conference. Trenčín: ADuT, 2007. ISBN 978-80-8075-248-4. pp. 40-44.

[17] Thurský, B. 2008. *Ako vzbudiť záujem detí o riadiace systémy*. In *Otvorená univerzita 2008* : Proceedings of the 2nd conference. Trenčín: ADuT, 2008.

[18] Urbančok, J. *Krúžok RoboLab v ZŠ Na Dolinách*. In *Otvorená univerzita 2007 :* Proceedings of the first conference. Trenčín: ADuT, 2007. ISBN 978-80-8075-248-4. pp. 45-47.

[19] Tunega, T. - Koptík, M. 2008. *Základy programovania mechatronických systémov : Príručka pre prácu s MSL RoboLab - jazyk NQC*. Trenčín: ADuT, 2008. ISBN 978-80-8075-349-8. p. 78.

[20] Zgodavová, K. Majerík, M. 2009. *Mechatronic Product Proportionality and Inter-Changeability Management: Mechanical Components.* In *Annal of DAAAM for 2009&Proceedings of the 20th International DAAAM Symposium Intelligent Manufacturing & Automation: Focus on Theory, Practice and Education : World Symposium.* Vienna: Vienna University of Technology, 2009. ISBN 978-3-901509-70-4. pp. 1933-1934.

Sophisticated Measurement of Non-Electrical Parameters Using Image Analysis

Libor Hargaš, Dušan Koniar, Viktor Bobek, Stanislav Štofan, Miroslav Hrianka
Department of Mechatronics and Electronics
University of Žilina, Faculty of Electrical Engineering
Žilina, Slovakia
{Libor.Hargas, Dusan.Koniar, Viktor.Bobek, Stanislav.Stofan, Miroslav.Hrianka}@fel.uniza.sk

***Abstract*—Paper focuses on main topics in frequency measurement with high speed video camera. The system is designed as phantom measurement. Phantom is realized with linear motor and controlled by DSP processor. The motor frequency is measured using image acquisition. This acquisition is done by high speed video camera. The sequences of captured images are processing with image analysis. Image analysis and other algorithms are done by virtual instrumentation using LabVIEW. The parameters of measured objects give relevant information about frequency and trajectory. This system can be used in sophisticated measurements in many educational, research and industrial applications where moving objects of investigation can't be equipped with sensors of kinematic parameters.**

Keywords-image processing, virtual instrumentation, dimming, regulation

I. INTRODUCTION

Objects investigated by microscopy usually can't be highlighted using reflection marks or equipped with sensors of kinematic parameters. In this case we often use advantages of image analysis and processing. Some methods for frequency measurement (using photodiode and photomultiplier) of biomechanical or microscopic objects can't do the correct analysis of structure pathologies. The most progressive method is high speed digital video method, which brings relatively good results in formation of mathematical and mechanical model of structure movement [1].

As example of moving biomechanical system we can consider cilium of respiratory epithelium cell. Each ciliated cell of respiratory epithelium contains ca. 200 cilias (6 μm long) beatings with frequency of 1000/min. Cilias are synchronized with metachronal waves propagated in periciliar liquid. From the basic position cilium folds down to the epithelium cell (recovery stroke – 75% of beating cycle) and then rapidly darts up to move mucus with its tip (effective stroke) [2].

II. DESIGN OF ACQUISITION SYSTEM

Important parameter in measurement process of biomechanical systems or moving structures is object beating frequency (OBF). In the case of respiratory epithelium this parameter has specific name: CBF (ciliary beating frequency). The value of CBF is normally in range 18-30 Hz. Image processing and FFT-based method require high-speed video acquisition system with optimal frame rate up from 400 fps [3], [4], [5].

Measurement method designed by our team is based on frequency analysis of intensity variance curve. This curve is obtained from video sequence by capturing intensity variation in selected region of interest (ROI). Curve is then analyzed with FFT algorithm, measurement is verified using curve thresholding and envelope analysis. Whole algorithm is shown in Fig. 1.

Measurement is done in graphical development system NI LabVIEW as virtual instrument and results are written as Microsoft Excel XLS file. This component helps to integrate results of investigation to laboratory or clinic information systems. Next advantage of LabVIEW virtual instrument is called Web Publishing Tool. Using this tool we can provide control of whole application through Ethernet or Internet connection.

Quality and method accuracy depends on quality of acquired video sequence and used acquisition system. The main aim of this analysis was removal of recording intensity variance curve with hardware photosensitive devices and usage of signal processing tools like autocorrelation, FFT or PSD (Power Spectral Density) for calculating object beating frequency. We can split measurement into a few solution steps, where algorithm calibration and debugging is done on phantoms:

- phantom acquisition with defined beating frequency;
- sequence preprocessing before intensity variance curve recording;
- processing of variance curve (spectral methods);
- applying on real videosequences.

To generate an accurate object beating frequency, we used DSP controlled stepping motor with reflection mark on the vane of its propeller (Fig. 2). Phantoms with defined parameters are common and useful parts in design process of new diagnostic or measuring method.

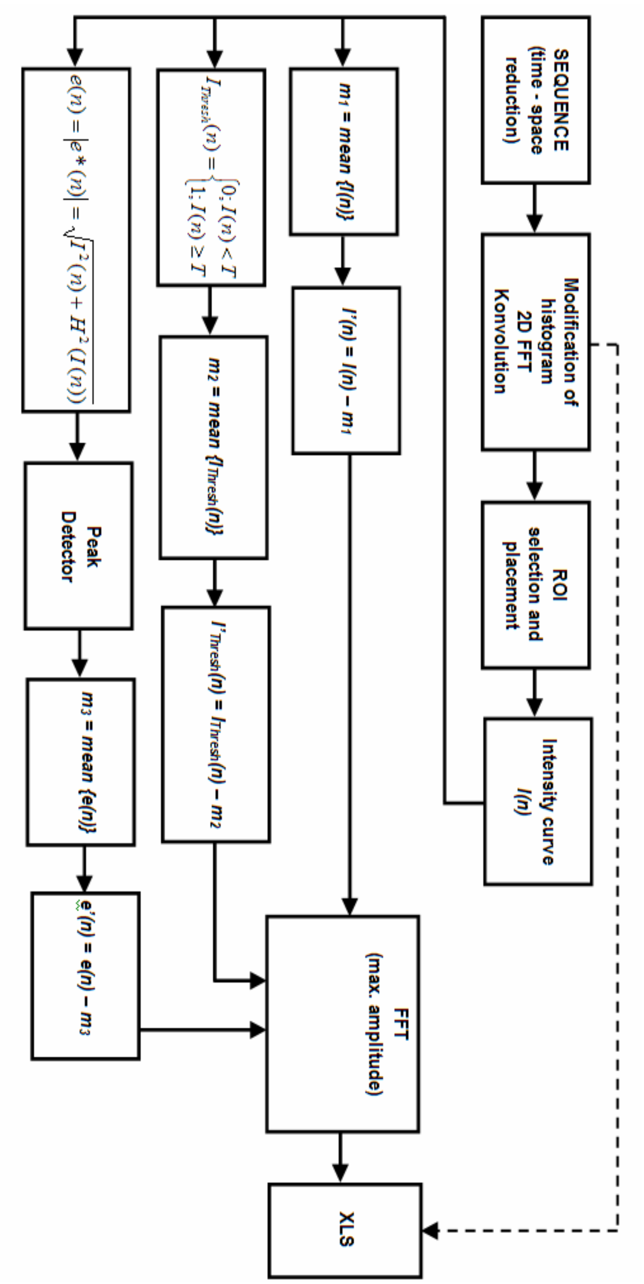


Figure 1: Frequency measurement algorithm

The first real measurements (in Clinic of pathological physiology, Jessenius Faculty of Medicine, Martin, Slovakia) were taken after algorithm debugging on phantoms. Because the ciliary beating frequency 'in vitro' goes down from ca. 18 Hz to a half value, primary we used acquisition system with slower camera. AVT Marlin F-046B camera was connected to inverse biological light microscope MODEL IM 1C via C-mount adaptor. Sequences from camera were stored on acquisition computer through IEEE 1394 (FireWire) as uncompressed sequences with parameters: 8 BPP / 640 x 480 pxl / 60 fps.

In the case of usage high speed camera system (Basler), microscope illumination is very important. We have changed microscope condenser light source and made some measurements of intensity by Lutron LX-1102 luxmeter.

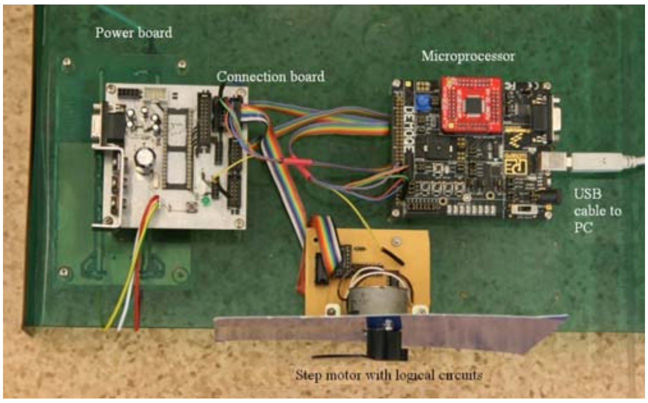


Figure 2: DSP controlled stepping motor phantom

In case of ultra high frame ratio of camera we can meet these essential problems: if the illumination of specimen is too low, frames in video sequence are underexposed and dark; if the illumination of specimen is too high, frames are overexposed and too bright (Fig. 3); high frame ratio causes growth of data for storage, so we must consider optimal connection between camera and acquisition computer.

Original maximal value of illumination (measured between condenser optics and specimen) changed from 8,6 klx to ca. 80 klx after replacing 20W halogen lamp for 120W halogen lamp. Heat from condenser must have been removed using active CPU cooler mounted onto microscope or using intelligent dimming tool.

For automatic setting of some parts of acquisition system we created regulation feedbacks controlled by LabVIEW virtual instrument (Fig. 7, Fig. 8, Fig. 9). Block diagram of mutual connections and feedbacks of whole system is in Fig. 10.

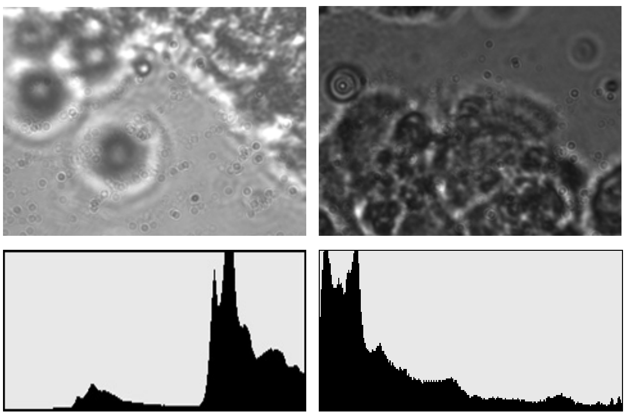


Figure 3: Overexposed (left) and underexposed (right) sequence frame and their histograms with greyscale distributions

In process of sequence acquisition, a ROI placed into image extracts important image feature: average intensity and histogram distribution. Overexposed image has its histogram concentrated to high intensity values and underexposed image to low values (Fig. 5). Histogram distribution is used as regulation parameter for setting up the PWM for halogen lamp dimmer. Lamp dimmer communicates with LabVIEW PCI card NI PCI-6221 (Fig. 11).

On the basis of simulations was designed wiring dimmer. The power supply is made directly from the mains supply 230V via input transformer. Dimming feature is provided through one PWM channel, which generates pulses at gate on switching transistor.

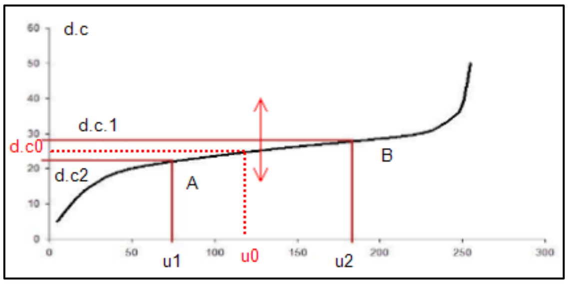


Figure 4: Image histogram mean dependence on dimmer switching pulses duty cycle

The measurement card is generating impulses to circuit. The protection and electrical separation of measurement card from the power circuit is guaranteed through the opto coupler and driver.

We can approximate the curve between points A and B (Fig. 4) as linear line:

$$y = ax + b \tag{1}$$

$$A = \lfloor \mu_1, d \cdot c_1 \rfloor; \; B = \lfloor \mu_2, d \cdot c_2 \rfloor \tag{2}$$

$$a = \frac{d \cdot c_2 - d \cdot c_1}{\mu_2 - \mu_1} \tag{3}$$

$$b = d \cdot c_1 - \frac{d \cdot c_2 - d \cdot c_1}{\mu_2 - \mu_1} \mu_1 \tag{4}$$

$$y = \frac{d \cdot c_2 - d \cdot c_1}{\mu_2 - \mu_1} x + d \cdot c_1 - \frac{d \cdot c_2 - d \cdot c_1}{\mu_2 - \mu_1} \mu_1 \tag{5}$$

$$\mu_i \langle \mu_0 \Rightarrow d \cdot c \uparrow \tag{6}$$

$$\mu_i \rangle \mu_0 \Rightarrow d \cdot c \downarrow \tag{7}$$

$$d \cdot c_i - d \cdot c_0 = \pm \Delta d \cdot c \tag{8}$$

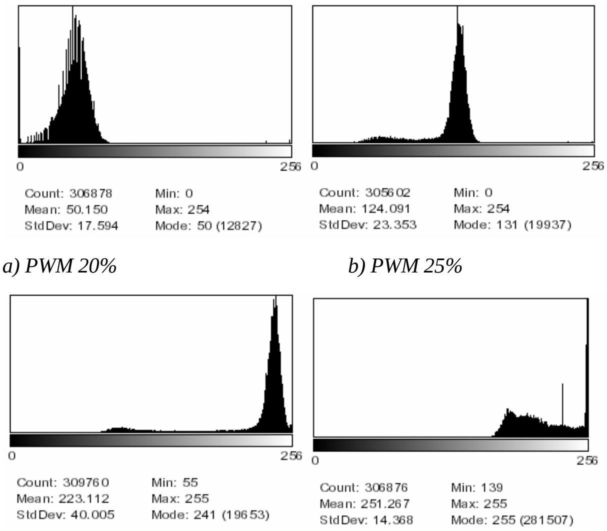
where d.c – duty cycle value
μ – gray level value

The dimmer control performance is realized through virtual instrumentation.

The proposed virtual instrumentation consist a connection of high speed video camera applications and light microscope regulatory loop. This regulation is implemented by setting the optimal switching pulse width (duty cycle). The standard frequency pulse from the measurement card is about 25 kHz to 50 kHz.

The optimal duty cycle choice is realized by the ongoing analysis of statistical parameters of the histogram of image acquisition.

The mean position of the image histogram is an important indicator of proper lighting (Fig. 5). The optimal mean value is the case in the mid-range gray level (128 at 8 bpp).



a) PWM 20% *b) PWM 25%*

c) PWM 30% *d) PWM 40%*

Figure 5: PWM image histograms

The measured characteristic is the value of the high speed video camera frame 60 fps. In this frequency was used for the halogen lamp set optimal duty cycle range 20-30% chart. The histogram mean was about 124, in duty cycle 25% and a value close to the optimum value.

While increasing the frequency is a horizontal shift towards higher value of duty cycle. Application of this regulation is using the approximation of the curve in Fig. 6.

Approximation is done with a 1% resolution of duty cycle and tries appearing mean to value near 128. For different fps is necessary to create approximations of the corresponding curves.

For halogen lamp dimming a 'switch-mode' controller circuit will be used. This type is highly efficient because very small waste of heat is generated. Nowadays there exist many different varieties depending on what type of switching system is used.

A basic Pulse-Width Modulation (PWM) switching unit suits this application. The regulator should be able to control lamp brightness from 0% to 100% (in dependency on duty cycle of the high-frequency PWM channel). Only drawback of this type of unit is the generation of high frequency 'noise' or emissions. Most of them will probably not cause any concern but they should be shielded or filtered. Another important thing is protection criterion, whereby such kind of regulator has to be fully protected against intermittent output short-circuits, input over-voltage and under-voltage conditions [6], [7], [8], [9], [10], [11].

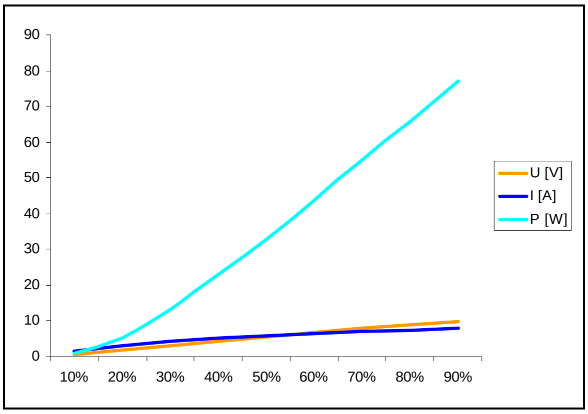


Figure 6: Dimmer output power dependencies

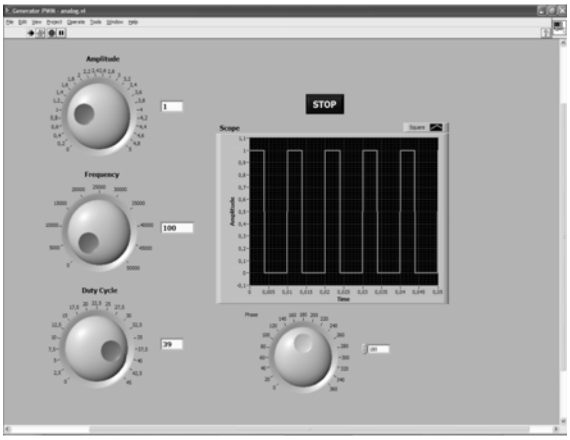


Figure 7: LabVIEW PWM regulation Front Panel for dimming application

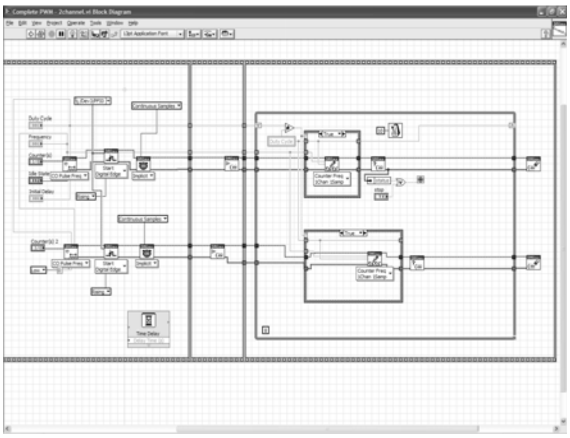


Figure 8: Part of LabVIEW Block Diagram for PWM dimming regulation & voltage / current measurements

Figure 9: FFT-based measurement Front Panel

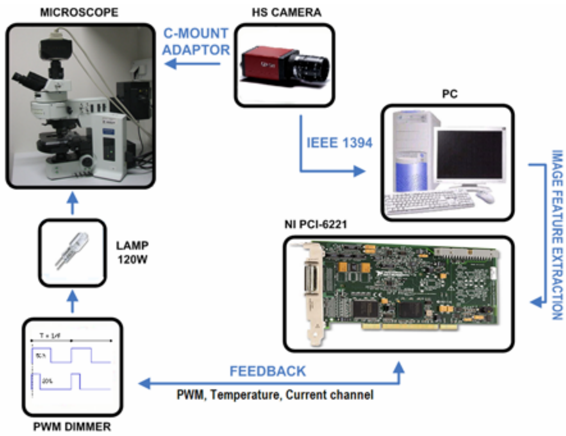


Figure 10: Scheme of acquisition system with devices, connections and feedbacks

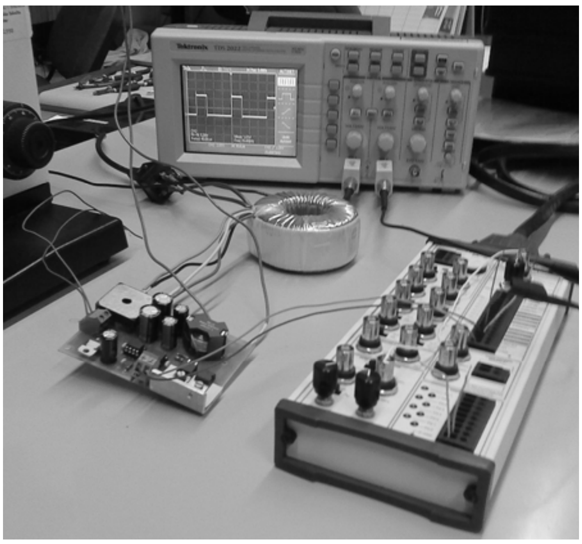


Figure 11: Testing mutual communication between microscope dimmer and LabVIEW measurement card

III. CONCLUSION

Designed solution for measuring object beating frequency from video sequence using tools of image analysis and spectral analysis simplifies present used methods and reduces usage of hardware devices. Using some development environment (e.g. NI LabVIEW) we can create fully automated application with interactive inputting of some parameters.

In the first approach, algorithms were tested on phantoms with defined frequency. Intensity variance curve analysis can be used in many other applications dedicated to frequency measurement not only in biological environment. Designed hardware acquisition system can be used with or without microscope in applications, where placement of kinematic parameters sensors is not able. Intelligent regulation of condenser illumination through image features extraction and histogram analysis enables fully automated approach to video sequence acquisition.

ACKNOWLEDGMENT

Results of these experiments were made with support of grant agency VEGA, project No. 1/0704/08 and No.1/0087/09, they are parts of project "Centre of Experimental and Clinical Respirology", IMTS code: 26220120004, "Centre of Experimental and Clinical Respirology II", "Measurement of Respiratory Epithelium Cilium Kinematics" and R&D operational program „Centrum of excellence of power electronics systems and materials for their components", No.OPVaV-2008/2.1/01-SORO, ITMS26220120003 funded by European Community, also for financial support to Slovak Research

REFERENCES

[1] Eyman Shebani: Ultrastructural Studies of the Airway Epithelium in Airway Diseases (Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Medicine 123), Acta Universitatis Upsaliensis, 2006, Uppsala, ISBN 91-554-6492-0

[2] Javorka, K. et al.: Medical Physiology, Martin, Osveta, 2001, ISBN 80-8063-023-2 (in Slovak)

[3] Mark A. Chilvers, Christopher O'Callaghan: Analysis of ciliary beat pattern and beat frequency using digital high speed imaging: comparision with the photomultiplier and photodiode methods, BMJ Publishing Group Ltd & British Thoracic Society, Thorax 2000, 55:314-37(April)

[4] Zhang, L., Sanderson, M. J.: Oscillations in ciliary beat frequency and intracellular calcium concentration in rabbit tracheal epithelial cells induced by ATP. In The Journal of Physiology (J Physiol) [online] February 2003, Volume 546, Number 3

[5] Gheber, L., Priel, Z. Extraction of cilium beat parameters by the combined application of photoelectric measurements and computer simulation. Biophysical Journal [online] Január, 1997, Volume 72, Issue 1, ISSN 1542-0086

[6] www.stmicroelectronics.com [online, February 1st, 2010]

[7] Frívaldský M., Drgoňa P., Príkopová A.: Design and modeling of 200kHz 1,5kW LLC power semiconductor resonant converter, IEEE International conference on applied electronics, Pilsen 2009 (September)

[8] Drgoňa P. - Bobek V. - Dobrucký B. - Frívaldský M.: Performance analysis of equation computation for real time model of PMSM for VHFIM control of PMSM position drive, EDPE 2009 (October), International Conference on ELECTRICAL DRIVES and POWER ELECTRONICS

[9] Drgoňa P. - Bobek, V. - Príkopová A. - Boggarpu N. K.: Metódy návrhu diskrétnych regulátorov pre zvyošvací menič (in Slovak). In:Sborník príspevkov konference ZVŮLE 2008, IEEE Workshop ZVŮLE 2008, Zvůle, Česká republika, 25.-27. August 2008, ISBN 978-80-214-3709-8

[10] VITKO, A., JURIŠICA, L., MURÁR, R., KĽÚČIK, M., DUCHOŇ, F.: Prediction of Imminent Faults in Mechatronic Systems. In: Process Control 2008 : Proceedings of 8th International Scientific-Technical Conference. Kouty nad Desnou, Czech Republic, 9.-12.6.2008. - Pardubice : University of Pardubice, 2008. - ISBN 978-80-7395-077-4

[11] VITKO, A., JURIŠICA, L., KĽÚČIK, M., MURÁR, R., DUCHOŇ, F.: Sensor Integration and Context Detection in Mechatronic Systems. In: Mechatronika 2008 : Proceedings of 11th International Conference on Mechatronics. Trenčianske Teplice, Slovak Republic, 4.-6.6.2008. - Trenčín : Trenčianska univerzita A.Dubčeka, 2008. - ISBN 978-80-8075-305-4. - p. 49-53.

„We supported the research activities in Slovakia/Project is co-financed from EU sources"

List of Authors