RiE 2011

2nd International Conference on

Robotics in Education

15-16 September, 2011
Vienna, Austria

Roland Stelzer and Karim Jafarmadar (editors)

RiE 2011

2nd International Conference on

Robotics in Education

Conference Proceedings

Co-hosted by
INNOC – Austrian Socitey for Innovative Computer Sciences and
Austrian Federal Ministry of Science and Research

Roland Stelzer and Karim Jafarmadar (Editors)

15–16 August 2011
Vienna, Austria

www.rie2011.org

ISBN: 978-3-200-02273-7

Contents

Introduction

Scope of the conference

The International Conference on Robotics in Education (RiE) brings researchers, teachers, practising engineers, as well as industry experts from all over the world onto a common platform.

The conference presents new trends, practical experiences, and the latest innovations and advances in the area of Robotics in Education and covers in particular the following topics:

- Robotics in school
- Robotics curricula
- International trends in educational robotics
- Hardware and software of robotic kits
- Laboratory experiments for teaching robotics
- Teaching and training for robotics
- Project-based learning and robotics
- Didactic approaches and materials
- Exemplary robotics projects in classes
- Robotics competitions
- Evaluation and pilot studies
- Web-based robotics and simulation
- Evaluation and assessment of robotic-enhanced class activities

Organising committee

The conference is jointly organised by INNOC - Austrian Society for Innovative Computer Sciences and the Austrian Federal Ministry of Science and Research and is technically sponsored by IEEE Austria Section.

General co-chairs

- Roland Stelzer, INNOC, AT
- Karim Jafarmadar, INNOC, AT
- Richard Balogh, Slovak University of Technology, SK

International program committee

- Richard Balogh, Slovak University of Technolgy in Bratislava, SK
- Simon Coupland, De Montfort University in Leicester, UK
- Hugo Ferreira, Instituto Superior de Engenharia do Porto, PT
- Gerhard Gruhler, Heilbronn University, DE
- Peter Hubinský, Slovak University of Technolgy in Bratislava, SK
- Karim Jafarmadar, INNOC - Austrian Society for Innovative Computer Sciences, AT
- Robert John, De Montfort University in Leicester, UK
- Dariusz Marchewka, AGH University of Science and Technology in Krakw, PL
- Martin Mellado, Universitat Politècnica de Valncia, ES
- Mark Neal, University of Aberystwyth, UK
- David Obdržálek, Charles University Prague, CZ
- Alexander Schlaefer, Universitt Lbeck, Deutschland, DE
- Fritz Schmöllebeck, University of Applied Scienced Technikum Wien, AT
- František Šolc, Brno Universtiy of Technolgy, CZ
- Gerald Steinbauer, Graz University of Technology, AT
- Roland Stelzer, INNOC - Austrian Society for Innovative Computer Sciences, AT
- Francis Wyffels, Ghent University, BE

Robotic Team Projects at FEI STU

Richard Balogh and Jozef Škultéty
Slovak University of Technology in Bratislava
Ilkovičova 3, 812 19 Bratislava, Slovakia
Email: richard.balogh@stuba.sk

***Abstract*—We will describe organization of team projects at the Faculty of Electrical Engineering and Information Technology of Slovak University of Technology in Bratislava. We would like to share and discuss our experiences with the project-based learning. As an example, the robot J2MP, winner of contest Robotchallenge is presented.**

***Index Terms*—project-based learning, team work, robot contest**

I. Introduction

It was demonstrated many times (e.g. [1], [2], [3]), that challenges in mobile robotics can become rich design experiences for engineering students as a part of their curriculum. Students are exposed to real problems and have to grasp engineering concepts in a very practical way. Then they can become more active, with a teacher serving as consultant. As written by Resnick [3], we can focus on themes and projects that cut across disciplines, taking the advantage of rich connections among different domains of knowledge.

Robotics is especially appropriate for team skills training as there are usually involved more application areas (hardware design, software design, mechanical design, modelling, signal processing, etc.) – it offers good possibilities to distribute team roles. Some of them are in their nature more experimental, some more theoretical.

Examples of the project-based courses can be found all over the world: MIT has *2.017J Design of Electromechanical Robotic Systems* [4], similar course *227-0080-00L PPS im Basisjahr* (valued 3 ECTS) is offered at ETH Zurich [5], both for undergraduate students. Their focus is on practical exercises and skills in design of complex systems, circuit design, instrumentation, microprocessors, programming and they also offer the possibilities of free experimentation and creation.

Of course, this type of education has also some drawbacks. Some authors mention overhead to manage large projects [6], or funding problems [2]. It requires significant efforts in coordination, management, orientation and commitment by all, but returns are amazing: students got to develop specialized skills in a real, hands-on, interdisciplinary environment; research in robotics progressed from innovations reported from this work; and the experience inspired enthusiasm [1].

We would like to share our experiences with robotic contest oriented projects. The text is organized as follows: In the first section we describe an organization of the *Team Project* at FEI STU in Bratislava. Then, we describe rules of the Puck Collect category, which was chosen as an appropriate and attractive topic for the project. Later we describe the construction of the robot constructed during the team work which won the competition in 2010 and 2011. We will also mention the software architecture and its implementation. In the last section we will mention some pros and cons of such educational style.

II. Team projects at the FEI STU

The Faculty of Electrical Engineering and Information Technology is one of seven faculties of the Slovak University of Technology (STU), the oldest and the largest university of technology in Slovakia. It offers accredited programmes within a complex bachelors, masters and PhD study system.

Course *35031_3I Team project* is a standard part of master courses in the study branch *Applied informatics* at the Faculty. Its value is 4 credits in the ECTS system. The course lasts 2 semesters. During the first semester, the goal is to prepare students for teamwork on large-scale projects, to demonstrate their ability to communicate with others, to share tasks reliably, to design a product (or its part) which is understandable and modifiable by others. During the second semester, it is focused on creation of an integrated product and its presentation.

Students learn to work in team, communicate between team members and with a customer, distribute tasks and responsibilities between team members, create a product and document all steps in such manner that anybody is able to continue with the product. Each team also presents its results,

Fig. 1. J2MP Team – winners of the Robotchallenge 2010.

advocate it against critical comments and clearly formulate team ideas.

The evaluation consists of the proposal evaluation, plan and methodology specification, project proposal and evaluation of a prototype. In the end, students present the project and advocate it in front of the commission consisting of teachers and concurrent teams.

The focus is on design of large software projects. As there is one special study branch *Information Technologies for Control and Automation*, we try to broaden the focus also on design, construction and testing of robotic systems. Within the project, each student should be responsible for a specific subsystem. Lectures on project management and tools for group work in engineering practice are included.

Recommended literature for students is available both in Slovak [7] and English languages [8], [9], [10].

Following table summarizes some of problems solved by teams within previous years (some team web pages listed below):

2011 *Freescale Race Challenge 2011*
http://scalectic.cstudios.sk/
Application development for iPhone
http://www.posterus.sk/iFly/
WebKit project I + II
http://petrex.yweb.sk/TP/
http://ayo.yweb.sk

2010 *Development of an open-source apps. A + B*
http://nailen.yweb.sk/
Freescale Race Challenge 2010

2009 *Robotchallenge*
http://ap.urpi.fei.stuba.sk/robotchallenge/
Internal Information System A + B

When it is possible, we try to establish two concurrent teams working on the same (or similar) topic. Then they can compete and also evaluate the opponents concepts at the end. Sometimes, this approach is not appropriate (e. g. for financial reasons). Then there is a good alternative: to participate in an open contest. When the contest is international, then it is even more appropriate to evaluate results of the project on an independent base.

A. Course milestones

During the team creation phase students split into teams and choose an appropriate topics from a list offered to them. Each team represents a small company struggling for a commision contract with a customer (represented by the teacher). Teams are to create two different project proposals (offers), one of them labelled as preferred.

The consumer's offer contains information about the team (members, experiences, recommendations etc.), motivation for the selected topic, and a rough version of the project solution. Also an actual time schedule for each member of a team should be attached to confirm that the team is able to have regular meetings.

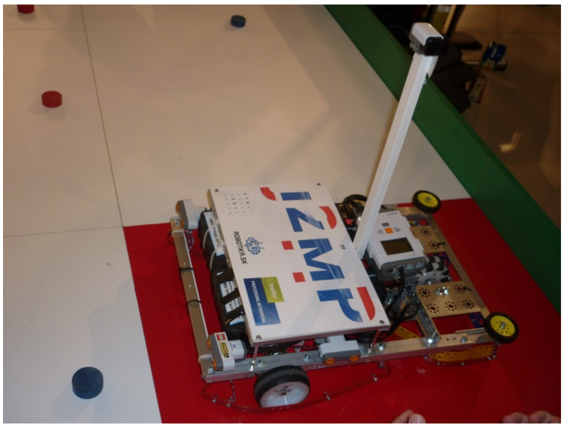


Fig. 2. Robotchallenge 2011: J2MP Robot in its homebase.

Quality of all proposals (including their formal side) is evaluated by teachers which can then choose their teams – there exists also a "competition" between teachers to acquire the best students.

After establishment of customer – company relationships students immediately start to distribute roles, tasks and responsibilities. They should decide who will be the team leader, who will be responsible for presentation issues including their own web page. They also elaborate their own time schedule.

Each team has to provide an actualized web page with project advancements. Web pages should contain actual information about the state of their projects, and all documents related to the project should be accessible from there.

Then the design phase starts. Design process should follow the Concurrent Engineering Methodology phases [1]:

1) Requirement analysis (customers requirements, constraints)
2) Functional analysis (translation of requirements into functional terms)
3) System design (analyse general concepts addressing identified functions)
4) Preliminary design (analyse specific concepts for different subsystems)
5) Detailed design (for each subsystem, calculations, drawings, schematics)
6) Integration and validation (assemblage all parts, and testing)

Teams have to organize regular meetings and provide teachers with minutes of each meeting. Minutes are a part of the documentation created during the project realization. It should contain basic information about the meeting (place, date, participants) and detailed information about the state of tasks from previous meetings, actual solved problems, times and persons responsible for the tasks.

At the end of the first semester, teams have to provide a documentation together with a prototype (where appropriate) to let a teacher tailor the right direction.

Then, during the second semester, they have to work on the project realization, keep actual web page of their ”company” and prepare a detailed technical documentation of the project.

At the end of the second semester, each team receives a critical expert's report. Expert's opinion is created by members of another (competitive) team. It should contain the evaluation of the project (analysis, design, description) including its formal side. They are expected to emphasize both positive and negative qualities of the project.

The course is finished by the public presentation of results. It should involve all members of the team and it must not last longer than 15 minutes. Then students have to answer comments from the expert's report and various questions from the public.

Everything is evaluated by commission of teachers which award each team with an appropriate number of points. Evaluation is pertained for the team as a whole, detailed distribution of points is left on the team alone. Surprisingly, they usually distribute points equally, even the evident unbalanced load during the work.

B. Components of the Course

Basic components of the course can be summarized as follows (exclamations mark the most problematic parts):

- Role and task distribution
- Organizing regular meetings, writing minutes
- Maintaining the web page
- Writing technical reports
- Evaluation of the results
- Document sharing and changes tracking
- Presentation skills
- Communications (team, consumer)
- Responsibility (!)
- Time management (!)

A brief description of a recently developed robotic system provides an example of a successful project. Its purpose was to build a robot for the international robotic contest Robotchallenge in Wien. It may serve as an insight into the possibilities of this type of education and for inspiration.

III. ROBOTCHALLENGE 2011 - PUCK COLLECT

RobotChallenge is one of the world's biggest competition for self-made autonomous mobile robots. Since 2004, more than 1000 robots from all over the world took part in the competition [12]. More than 250 robots from 16 countries from all over the world participated in the last event in March 2011 in Vienna. RobotChallenge is an event hosted by the Austrian Society for innovative Computer Science (InnoC). It is an independent research centre founded in 2005 and located in Vienna, Austria.

One of Robotchallenge disciplines is *Puck Collect*. This competition calls for special interaction between sensor technology, mechanics and artificial intelligence. Eobots have to collect small discs in the field according to colour [13].

In the contest two robots compete. Their dimensions are limited to 50×50 cm, their height and weight are not limited. Ten red and ten blue pucks are spread randomly in the field

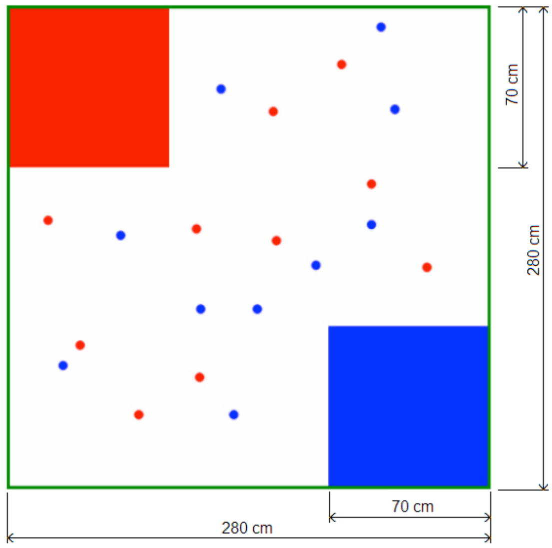


Fig. 3. Puck collect game field

which is 280×280 cm large and bordered by 10 cm high boards. Pucks are wooden disks with 4 cm diameter and 1.7 cm height.

The aim is to collect all pucks of the assigned colour and carry them to own home base. Home bases are 70×70 cm large, red and blue, and they are positioned in opposite corners of the field. The remaining part of the field (the neutral zone) is white.

A. Structure of the Game

Each of two robots gets an assigned colour (red or blue). In the beginning of the match each robot is placed on its home base. When a judge announces the start of the round, teams start their robots. The aim of this competition is to collect all pucks of the assigned colour. A puck is counted as collected, if it is situated anywhere above the home base of the same colour. Therefore pucks need not be unloaded.

The robot which first collects all its pucks, wins the game. If no robot manages to collect all its pucks within three minutes, a time-out occurs, and the game is stopped and the robot which collected more pucks wins.

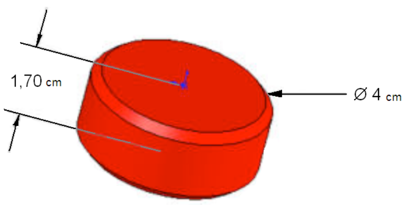


Fig. 4. Dimensions of the puck.

There are also two special rules – if a robot puts one of its pucks onto the foreign home base, the puck will be taken out of the game by a referee. If a robot puts a foreign puck onto its own home base, the referee moves this puck into the opposite home base. This case is called ”own goal”.

More detailed rules can be found at the Robotchallenge website [13].

In 2010, 23 teams were registered there, but only 13 really participated. In 2011, the number of participated teams increased to 15. As this competition is very challenging, it was a clear choice for the *Team Project*.

IV. J2MP Robot

In 2009, the team J2MP consisting of four students (Jozef Škultéty, Ján Maláč, Michal Beňo and Peter Mihál) began to work on its robot for the Robotchallenge.

A. Hardware

First construction was based on LEGO Mindstorms [14] and LEGO Technic parts. Students enjoyed the ease of construction and an opportunity to test various configurations and approaches. Unfortunately, the resulted construction was not strong and robust enough, so we start to combine mechanical parts with some Merkur (Meccano equivalent) components. Later it was changed to a rigid metal chassis built from the standard aluminium profiles completed with various components from Meccano and LEGO Mindstorms kits.

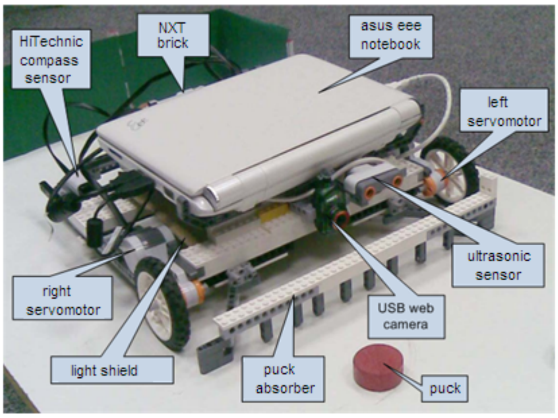


Fig. 5. Abandoned version of the J2MP Robot.

Controller design evolved from the Gumstix Overo Earth processor [15] with the CMUCam3 programmable camera [16] to the hierarchical control based on the low-level control with the NXT processor brick (LEGO Mindstorms) for motors and sensors, and the high-level control provided by an Asus EE notebook with an USB web-cam for image processing.

Later, the image processing approach was completely abandoned and we rely purely on the colour sensor and only the NXT controller alone. That makes the robot lighter and also more reliable due to the smaller number of components.

During initial experiments with the robot we grabbed images from the USB web-cam and we applied the Gauss blur filter on it to remove a noise. Then our own colour filter based on HSV components was applied. Resulted image was processed by the Sobel edge filter together with a Hough Transformation [22] to detect the position of the closest puck of selected colour.

Unfortunately, the algorithm was very sensitive to changing light conditions. Together with a narrow area scanned by the camera we definitely abandoned the idea of image processing as an inappropriate for this purpose. Instead of this, the colour sensor was used for the puck recognition.

Light conditions significantly influenced the success of the colour recognition. It was necessary to shield the sensor area and we also used the additional source of light (white LEDs). Under the almost constant light conditions the colour sorting using the HiTechnic colour sensor worked as supposed. During the contest, we *never* encounter the wrong colour decision.

Following list summarizes all sensors used on the robot:

- Ultrasonic Distance sensor 4x
- HiTechnic Color sensor [17]
- HiTechnic Compass sensor [18]

B. Colour recognition

For safe navigation over the field, we used two ultrasonic distance sensors in the front of the robot. Although it was sufficient in 2010, in the next year we added also sensors on the left and right sides of the robot to improve safe movements without collisions with opponents or walls.

Compass sensor used for the orientation of the robot was very sensitive to interferences from the motors. Even if we placed the sensor on a plastic pole and thus removed own interferences, some competitors produced really large electromagnetic disturbances.

As the total numbers of sensors exceed the number of the NXT inputs, we used the HiTechnic Sensor multiplexer [19] to connect them to the controller.

The initial design started with platform driven by two NXT motors. As the complexity of the system increased, it was clear that their power is insufficient, but there was no alternative at the time of the contest in 2010. The improved design for the 2011 contest used the HiTechnic DC motors from the Tetrix [20] kit. We also replaced the ball caster with a Tetrix omnidirectional wheel. Now, the robot is faster and can easily deal with stuck situations.

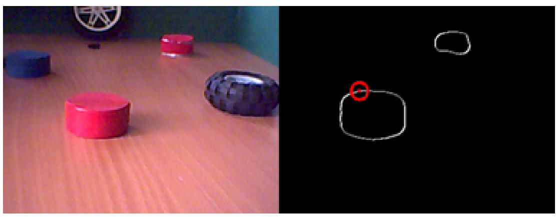


Fig. 6. An example of the camera image processing.

Total number of actuators used is

- 2x Tetrix motors for fast platform movement
- 1x NXT motor for the puck manipulation

The robot was powered from the 12V NiMH accumulator pack and the NXT brick uses its own power supply from the six AA batteries. The huge NiMH battery also provided the necessary thrust to move faster.

C. Strategy

The robot task was to regularly scan the competition area, manipulate the sorting cross to select pucks of different colours and deal with exceptions (opponent robot hits, loss of the direction etc.).

Two different and equally important independent tasks were identified:

- Puck recognition
- Systematic field searching

Puck recognition starts when the original white background colour changed to either red or blue. For our colour, the puck is moved using the sorting cross to the bin, otherwise it is moved out. As we mention above, the puck recognition was based on the colour sensor information.

For **navigation and field searching**, many complex solutions are available. Our experiments showed, that even if the robot has no information about its actual position on the field, it was able to solve the task effectively. The robot had only an information about the actual direction regarding to home base based on the compass sensor information. The direction information was periodically corrected when the edge of the field was reached.

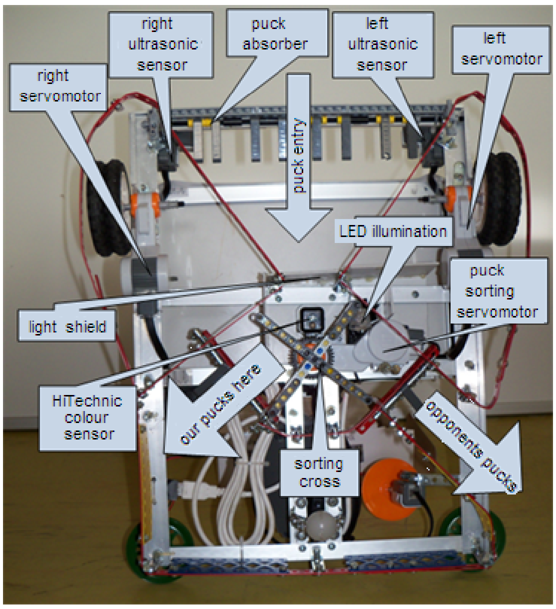


Fig. 7. J2MP Robot – bottom view.

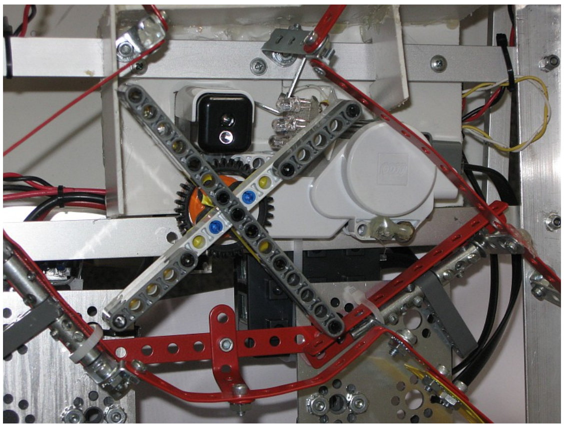


Fig. 8. J2MP Robot – detail of the upgraded sorting mechanism.

Another important feature we implemented was an opponent avoiding. We learned from previous contests that many teams lost valuable time jamming with the opponent robot, often with fatal results for robot components.

As the robot should return home safely, separate part of the algorithm was responsible for automatic home navigation with some back-up time to solve unexpected accidents (e.g. collision with the opponent robot).

D. Software

The program is written in Java language. We used the leJOS – a tiny Java Virtual Machine ported to the LEGO NXT brick [21]. Unfortunately, there were no classes for the Tetrix components available, so we need to write our own. Thanks to the HiTechnic company, we were provided with the source code for their RobotC library and we successfully ported the code for leJOS.

The main program consists of few threads with following classes:

- `MainThread.java` – class only for inheritance (parent of the thread classes),
- `RobotThreadControl.java` – control algorithm,
- `SensorsThread.java` – periodic actualization of the ultrasonic sensors values,
- `TetrixMotorThread.java` – periodic sending the last control command to the DC Motor Controller (we have to send the command each 3 seconds, otherwise the motor stops, there was no time to investigate it further),
- `VypisThread.java` – for debugging purposes, it sends the sensor data to a console,
- `WatchDogOfTimeThread.java` – this thread is for timekeeping; after the certain time elapses, it immediately starts the `returnHome` algorithm.

Not only that this algorithm is very simple, it brings also the advantage of its stability and reliability. We have seen many teams with overcomplicated algorithms, which were unstable and full of errors.

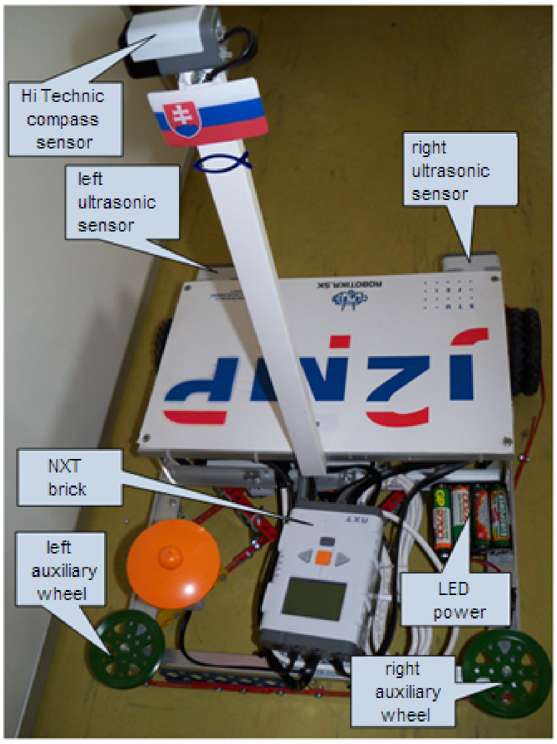


Fig. 9. J2MP Robot – final version for 2010.

Work on the robot takes a lot of time (much more than assumed for the *Team project* course), and we spent many hours by testing. The result was more than satisfactory – our robot won the *Puck Collect* category both in 2010 and 2011.

V. Conclusions

There are two main aspects of *team projects*: first is, of course, a technical challenge that *should be* solved. Usually we assume already existing knowledge and experiences from previous courses; new topics are self-studied by students themselves.

The second, formal side of the project is represented by the project management. We try to show students some theoretical background about dividing of roles, team leading, task distribution, document sharing and time management. Also the communication aspect plays an important role. Sometimes it is difficult to focus their interest also to the formal side, especially when they are just researching a technical side of the project. Our experience shows that it used to be done in a very beginning of the project when they just start to search ideas. We tried to focus them also on some more or less complex tools to help them with the project management (see e.g. a good overview in [11]). Anyway, we feel that formal side of the project management should be improved.

We observed that students are curious, they want to know everything, so sometimes is distribution of tasks very unclear (fuzzy), as they want to participate (or at least to see) in everything. Also, they don't know each other well enough, so they are probably uncertain to rely on the other team members. Sometimes they are simply unsatisfied with results of others…

On the other hand, it is important to concern that limiting students to work with only one aspect prevents them from experiencing other activities [6].

We also learned that participation in the (international) contest is a very good alternative to two competitive teams. The contest is very good motivation for a team to make the job as good as possible. They usually spent much more time than is "officially" required few weeks before the event. At the end, the atmosphere of the contest, discussions with other teams, and new contacts left them many positive impressions.

Moreover, contests are usually very attractive for media and thus helps us to popularize our study branches in the public. This is very important, if we face the fact the number of young people wishing to study technical disciplines is decreasing.

From a technical standpoint, the project has greatly benefited students. There is a number of engineering topics which students encounter while working on the robot. In terms of *control systems* experience and exposure, students have gained experience with developing image processing algorithms, system modelling and control algorithms. In terms of *software*, they had to face embedded systems limitations and code porting issues. In terms of *hardware*, the project required mechanical and modular design, interfacing issues and power analysis.

Together with design, lot of testing and evaluation of results was required. In addition to the development of technical skills, students also were exposed to issues faced by real-life situations (travelling, financing etc.).

We consider the Puck Collect competition as a very inspiring challenge and we would like to encourage other teams from universities to participate. We hope that our description of the J2MP robot construction will be a good starting point for others.

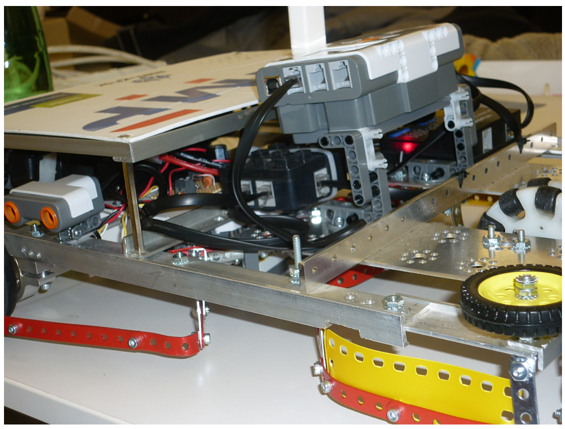


Fig. 10. J2MP Robot version 2011. Combination of parts from the LEGO Mindstorms, Meccano and Tetrix together with custom parts.

ACKNOWLEDGMENT

Publication of this paper was supported through the Scientific Grant Agency of the Ministry of Education, science, research and sport of Slovak Republic with the grant VEGA 1/0690/09.

REFERENCES

[1] François Michaud: *Engineering education and the design of intelligent mobile robots for real use.* International Journal of Intelligent Automation and Soft Computing, Special Issue on Global Look at Robotics Education, 2005. Available on-line: http://www.gel.usherb.ca/laborius/papers/IJIASC2007.pdf

[2] C. Kitts: *Surf, turf, and above the earth [robotics education].* Robotics Automation Magazine, IEEE, 10 (3) : 30 – 36, sept. 2003. Available on-line: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1233555

[3] Mitchel Resnick: *Rethinking learning in the digital age.* In: G. Kirkman (Ed.), The global information technology report: Readiness for the networked word. Oxford, UK: Oxford University Press. Available on-line: http://llk.media.mit.edu/papers/mres-wef.pdf

[4] Franz Hover and Harrison Chin: *2.017J Design of Electromechanical Robotic Systems*, Fall 2009. (Massachusetts Institute of Technology: MIT OpenCourseWare). Available on-line: http://ocw.mit.edu/courses/mechanical-engineering/2-017j-design-of-electromechanical-robotic-systems-fall-2009/

[5] PPS im Basisjahr – LEGO MindStorms Website des Instituts für Technische Informatik und Kommunikationsnetze (TIK) der ETH Zürich. Available on-line: http://www.tik.ee.ethz.ch/education/lectures/PPS/mindstorms/

[6] David Coppit and Jennifer M. Haddox-Schatz: *Large team projects in software engineering courses.* SIGCSE Bull., 37:137–141, February 2005. Available online: http://portal.acm.org/citation.cfm?doid=1047124.1047400

[7] Mária Bieliková: *Ako úspešne vyriešit projekt.* [In Slovak] STU in Bratislava, 2000. 158 s. ISBN 80-227-1329-5.

[8] Christian W Dawson: *The essence of computing projects a student's guide.* Harlow: Prentice Hall, 2000. 176 s. ISBN 0-13-021972-X.

[9] Curtis, B., Hefley, E., Miller, S.: *Overview of the people capability maturity model.* CMU-SEI-95-MM-01, September 1995.

[10] Watts S. Humprey: *Managing Technical People: Innoviation, Teamwork and the Software Process.* Addison-Wesley, 1997.

[11] Rudolf Mühlbauer, Thomas Klambauer, Walter Ruprechtsberger, Josef Schaitl: *Seminar Projektorganisation Comparison of Projectmanagement-Software.* Leitung: Mag. Dr. Elisabeth Kapsammer. Johannes Kepler Universität Linz Institut für Bioinformatik, Linz. 2007. Available on-line: http://klambauer.info/pms.pdf

[12] InnoC.at: *RobotChallenge 2011 Home Page*. Available on-line: http://www.robotchallenge.org/

[13] InnoC.at: *RobotChallenge 2011 - Puck Collect Rules.* Available on-line: http://www.robotchallenge.org/reglement/puck-collect

[14] Lego Mindstorms NXT programmable robotics construction kit. Available on-line: http://mindstorms.lego.com/

[15] Gumstix Overo Earth. Tiny Linux computer-on-module. Technical documentation. Available on-line:http://gumstix.com/store/catalog/product_info.php?products_id=211

[16] Embedded Vision Processor CMUcam3. Documentation. Carnegie Mellon University. Available on-line: http://www.cmucam.org/wiki/Documentation.

[17] NXT Color Sensor V2 (NCO1038). Available on-line: http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NCO1038

[18] NXT Compass Sensor (NMC1034). Available on-line: http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NMC1034

[19] HiTechnic Sensor Multiplexer (NSX2020). Available on-line: http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NSX2020

[20] TETRIX heavy-duty aluminum design system compatible with Lego Mindstorms. Pitsco, Inc. Available on-line: http://www.tetrixrobotics.com/

[21] LEJOS. Java for LEGO Mindstorms. Available on-line: http://lejos.sourceforge.net/

[22] Jamie Hutton and Ben Dowling: *Computer Vision Demonstration Website.* Electronics and Computer Science University of Southampton. Available on-line: http://users.ecs.soton.ac.uk/msn/book/new_demo/

Use of an Infocenter to Improve the Management and Understanding of Project-Based Learning Robotics

Kathia Pittí[1], Belén Curto[2], Joaquín García[1], Vidal Moreno[2], Román Ontiyuelo[3]

[1]*Dept. of Theory and History of Education, University of Salamanca*
Paseo de Canalejas, 169 - 37008, Salamanca, Spain
kathia_pitti@usal.es
carrasco@usal.es

[2]*Dept. of Computing and Automation, University of Salamanca*
Plaza de los Caídos, s/n - 37008, Salamanca, Spain
bcurto@usal.es
vmoreno@usal.es

[3]*International Centre for Advanced Technologies*
Calle Nuestra Señora, 65 - 37300, Peñaranda de Bracamonte, Spain
tallernxt@yahoo.es

***Abstract*— Robotics allows the implementation of a variety of interesting experiments in conjunction with the methodology of Project-Based Learning (PBL). It has the potential to become an ideal tool in the teaching of a wide variety of scientific and technological disciplines. Furthermore, the student is brought closer to the reality of the professional world through the completion of a project. However, when applying this methodology in an extra-curricular robotics workshop, it is important to bear in mind the following question: how can we ensure that the participants are fully aware of the stages involved in a project and have a realistic experience of project-management? In this article the use of a tool inspired by the world of business will be explained. We have decided to call it an '*infocenter*'. It is a strategy designed to enhance the effective management of the project and also the feeling of forming part of a team which is central to carrying out a project. To allow an evaluation of its usefulness the experimental results from the implementation of this strategy during the NXT Baby Sumo workshop are presented. Through the evaluation of the participants and the instructor, it is shown how the infocenter can be adjusted to suit this purpose.**

***Keywords*— project-based learning, robotics, project-management, infocenter, NXT workshop**

I. Introduction

For the 21st century generation, robotics, due to its polyvalent and multidisciplinary character, is an ideal motivational teaching method, and together with the use of project-based learning (PBL) it provides an extensive range of opportunities for learning which can then be applied in various contexts: in the classroom, as an extra-curricular activity or through competitions with robots.

The combination of these two concepts (robotics and PBL) means that robotics projects provide an excellent educational platform from which the student can develop and apply invaluable skills and knowledge. These are put into practice in a realistic working environment, as is indicated by several investigations and authors [1]-[3]. They also promote the development of skills which are essential for success in the modern world.

The basic idea is to use robotics and PBL to motivate the learners from a very young age and introduce them to real world experiments. Working within a project requires a variety of skills: a team must be formed, tasks must be planned and prioritised, roles assigned, a consensus reached etc. Of course, through merely introducing this approach alone the desired objectives will not be reached if the same strategies as before are still used.

For this reason we have opted for the design of an infocenter as a project-management tool that facilitates the understanding of PBL robotics, providing all the participants involved in the project with a roadmap for the successful completion of the project. The evaluation and tests of this design are realised within the NXT Baby Sumo Workshop.

In this article a few of the key ideas about the potential of PBL and educational robotics are examined (section I); after which the basic characteristics of PBL and PBL with robots are outlined briefly (sections II and III); then a didactic description of the infocenter is presented (section IV); and finally, there is an evaluation of the infocenter by the students (section V), which allows us to formulate some initial conclusions. At the foot of the article there is an acknowledgement.

II. Project-Based Learning (PBL)

Project-based learning (PBL) can be defined as a teaching method in which the learners complete a project within a set period of time to solve a problem or complete a task through the planning, design and completion of a series of activities.

All of this is based upon the application of acquired knowledge and the effective use of resources [4].

It is also an action-based style of learning. It does not consist simply in learning ‘about’ something (as happens in problem-based learning), but ‘to do’ something. To complete a project one needs to combine the knowledge of various areas as well as using the appropriate materials, thus avoiding a fragmented learning process.

Another benefit is that PBL is a method based on learning from experience [5], [6] and in reflective learning [7]. The investigation of a topic, with the aim of resolving complex problems, is central. As the project is student-centred and promotes self-motivation the learners acquire new knowledge and develop new skills by coming up with group solutions to problems. We will see in the next section how PBL can be used when designing robots.

III. PBL AND THE USE OF EDUCATIONAL ROBOTS

In light of these benefits PBL has been chosen as one of the methods used in teaching with educational robots [8]. Due to the multidisciplinary and collaborative character of this pedagogical technique it provides students with the opportunity to work in teams. ‘The organisations of the Knowledge Society will undergo constant change. They are organizations where people will work in teams and for teams [9].’

PBL is the fundamental work methodology employed in NXT Workshops [2], [10]. When the children construct the robots or experiment with them, they experience, at first hand, the complex process of creating ideas, solving problems and overcoming difficulties. The project work, organized within the context of the NXT Workshops, is aimed at promoting collaborative learning; this style of learning [11] is one of the most powerful tools for the development of the skill of learning to learn.

A. Advantages of PBL Robotics

The use of PBL in robotics-based learning will allow us to achieve objectives such as:

- *Solving new problems.* As science and technology advance, it is impossible to impart the knowledge and skills that an individual will need in the future. For this reason, rather than teaching them what to think, we must teach them *to think,* rather than giving them information we must teach them to *seek information.*
- *Group work.* This facilitates the student’s personal development. The learners acquire useful experience as well as a sense of the value of group work, besides the skills which result from group work and which are essential in all professions.

It is possible that PBL in combination with robotics produces other benefits, which are described in the literature that has been consulted [1]-[3], [12].

- It provides a practical way in which to learn to use technology.
- It allows the students to see and then make connections between the various disciplines.
- It presents a problem in a context similar to that in which the learners will encounter in their professional life, thus connecting the learning process with reality.
- It enhances their communicative and social skills.
- It encourages creativity and curiosity.
- The students learn to take their own decisions and to work independently.
- It strengthens their self-confidence.
- Being founded on experience it increases their motivation to learn and encourages the conception of objectives based on the task.
- It allows them to apply their knowledge, their skills and the new outlook they have acquired to specific situations, along with an improvement in the corresponding skills.

B. The Phases of PBL

In general, PBL is characterized by five [13] processes: (a) engagement, (b) exploration, (c) investigation, (d) creation, and (e) sharing. Underlying these five processes is an interactive analytical evaluation of the students' problem-solving approaches and solutions.

The TERECoP project [8] has identified several similar stages: the engagement stage, the exploration stage, the investigation stage, the production/creation stage and the evaluation stage.

The variations in the stages of PBL are numerous. This is due to the fact that each of the stages can vary from one educational facility to another. In the following section a visual tool will be presented which allows for the improvement of PBL as applied in the NXT workshops.

IV. INFOCENTER – DIDACTIC APROACH

As part of the NXT Workshops [14], more than a dozen activities have been organized since 2006. This has meant that the most recent workshops have been composed of a mixture of novices and experts. This enriches the activity significantly but also requires a greater commitment, not just from the instructor but also from the participants, to ensure that the tasks are completed within the stipulated period of time.

The continual desire to improve have led us to look for a simple project-management tool, from the professional world, and we aspire to introduce into the robotics workshops tools in order to work in a more efficient way and increase speed and flexibility during the execution of a project.

With this aim in mind, we investigated what is being used in the professional environment and thus introduced a tool which will be a genuine aid to them in their professional future. From this the infocenter emerges, inspired by the tool developed by the Scrum and Kanban processes [15] which help us work more effectively, to some degree, defining what must be done, how and by whom. Furthermore, both tools use a board as a visual representation to show the sequence of tasks and activities which are carried out during the project.

It is necessary to point out that as a visual management tool, apart from the control that visualization gives, it promotes ‘the ability to see the potential to transform’ and in this case transformation also includes the mental processing and interpretation of what has been seen. Moreover, a visual environment is very powerful as it is expressed in a language that the human brain is particularly efficient at processing: visual language. People absorb visual instructions more quickly, as it is the shorter learning curve.

The infocenter we have designed can be defined as a visual tool whose purpose is to manage, control and constantly update the information supplied to those participating.

In all of the previous workshops, during the final session before the competition or exhibition, each team has presented the robot that they designed and constructed together for their friends, family and classmates. This is accompanied by a small investigation into the main theme of the workshop. This activity is the only one in which we can appreciate the final product. This project-management tool would allow the instructor, the participants and even people who are not involved in the workshop (parents, friends etc.) to appreciate, in a visual way, the magnitude of the work entailed in each session leading up to the final challenge.

A. Basic Concepts of Scrum and Kanban

Experts in Scrum and Kanban [15] recommend avoiding a limitation to one tool, suggesting instead a combination of them to best meet the needs of each team. For example, many Kanban teams have daily meetings (a practice taken from Scrum). They urge us to be aware of the limitations of each tool and to experiment until we find something which works. Kanban and Scrum are not the objective. Continual learning is the objective.

In this sense, Scrum and Kanban are empiricists. It is expected that we experiment with the process and personalise it to suit our own environment. In fact, we must experiment. Neither Scrum nor Kanban give us all the answers – they merely set out a series of basic guidelines which direct our own process of improvement.

A few important concepts in which Scrum and Kanban [15] resemble one another are:

- Both are *Lean* and *Agile*.
- Both establish work in progress limits.
- In both the visibility of the process is the basis for its improvement.
- Both work with self-organized groups.
- Both require the division of work into modules.

The main differences appear in table 1:

TABLE I
DIFFERENCES BETWEEN SCRUM and KANBAN

Scrum	Kanban
The team takes on a work commitment for each task.	This responsibility is optional.
The teams must have multiple functions.	The teams are specialized.
Several teams or persons share the same board.	The board belongs to one specific team.
3 roles are assigned.	No roles are assigned.
The tasks must be prioritised.	Prioritisation is optional.

Both Scrum and Kanban can cover the entire system of the production of a product, including the work in progress limit, the capacity, the duration of the cycle, the quality and the changes in predictability, among other factors. Our objective is not as broad nor does it involve as many variables as in the case of the business world. Instead, it concentrates on improving the management of the teams, giving the children a visual guide of the tasks to be completed during a robotic workshop and the opportunity to experiment with a model that is very similar to a real world project.

Fig. 1 Design of the infocenter for Baby Sumo NXT

B. Design of the Infocenter

The infocenter consists of a visual board inspired by that used by Scrum and Kanban [15]. It possesses the advantages of these boards, which are as follows:

- Each person (team) chooses the task to be carried out. In other words, responsibility is assumed, not assigned.
- It is a light and valuable tool, which makes the work flow clearly visible.
- It is easy to respond to: Where are we?
- It focuses the team
- Bottlenecks are quickly observed.
- It is easy and cheap.
- The correct task is carried out at the exact moment in which it is possible.

Therefore, the infocenter is composed of two sections; the section on the upper left is used to show everyone the number of tasks to be done and the time available. It is important that the tasks accurately represent the complete process involved in the realisation of the project. They must be specific and easily comprehensible to everyone. As an extra guide for the students it is also helpful to organise the tasks in order of priority.

This board eliminates the necessity for group leaders. Everyone, the instructor in particular, can examine the board and observe, for instance, that a team is falling behind and offer them a little help.

The next decision we had to make was whether to use defined roles (Scrum) or shared responsibility (Kanban). We decided to try shared responsibility. One fundamental reason behind this decision was the level of experience of the students, who ranged from beginners to almost experts.

After selecting and distributing the sections that appear on the board it is put in the classroom of the workshop in a place that is easily visible by all of the teams.

Fig. 2 Placement of the infocenter in the classroom of the workshop

The second part of the design involves generating awareness of the steps to be followed in each session, both by the instructor and by the students. With this purpose in mind the sequence for the robotics workshops is described as a guide to students.

In the first session, the instructor explains to the students what this new visual tool is composed of, its purpose, and how it should be used (next section).

They are then told that the next sessions will begin with a stand-up meeting. This is a characteristic of Scrum and the objective of this meeting is to facilitate the exchange of information and collaboration between the team members so as to increase their productivity, while also indicating areas in which they can help one another.

To achieve this, each team must answer the following questions within a maximum time of 15 minutes.

- What have I done since the last session?
- Was I able to do all I had planned to do?
- What was the problem?
- What am I going to do next?
- What obstacles do I face or am I going to face in order to meet the requirements of this task?

It is worth pointing out that the purpose of the meeting about the state and the synchronisation of the team is not to resolve problems, the problems are resolved after the meeting.

The experts recommend:

- To carry out this meeting while standing, so that the team members don't relax nor spend too long speaking about superfluous details.
- To carry out the team collaboration meeting directly after the end of this meeting.

It is evident that these visual boards in isolation do nothing; rather it is the participants that do the work. The process will become evident through the interaction among the participants which of course will be different with each team and each project. Once the team members understand the aim of these meetings and they get used to concentrating and exercising discipline the meetings become effective. This eventually becomes another group work habit which helps them meet their responsibilities. Moreover, it facilitates learning, as they can see how their fellow team members work and react to circumstances.

C. Description of the Use of the Infocenter during the NXT Baby Sumo Workshop

As we have already mentioned previously, the infocenter can be defined as a visualisation of the project on a board in which the participants regularly enter information in the relevant sections. The advantage of this is that the work to be done and ongoing tasks are always present. This ensures that no one is without work at any time and that all the important tasks are carried out in the correct order.

First we had to select appropriate section headings in the 'NXT Baby Sumo Infocenter' which were tailored to our objectives. These were: the title of the project, a detailed plan of the tasks to be completed, the duration of the project (divided into 6 sessions), the names of the teams and the list of those taking part.

Fig. 3 Detailed plan of the tasks with the duration of the project

Some additional sections were also included such as: comments, incidents, risks and milestones. *Who are we? How do I feel today? How do we have fun?* All of this allows us to record each stage of the project and to feel part of the team throughout the exercise.

Fig. 4 Additional section: How do we have fun? of the infocenter

In the first session the students are offered some general information: what an infocenter is, what its purpose is and the way in which it is used. Each team is asked to place a post-it in the appropriate box in the progress monitoring board each time they complete a task or wish to remark on a section of the infocenter. This post-it must include the state (complete or ongoing) and the name of the team. This makes it possible for all the team members to see exactly how the team is advancing with the project, in an impressive example of visual control.

One initial design flaw arose as a result of using the same colour of post-its for all the teams. As the rapid identification of the post-its is of crucial importance in an infocenter of large dimensions and with many teams it is important not to waste time. Therefore, as a result of this first test, we suggested the use of post-its of different colours as a solution, assigning a specific colour to each team and including special post-its depending on the type and priority level of each task. Here are some examples: a post-it in the form of a rhombus indicates that a decision has been taken, a green post-it is used for improvements, a yellow one for project tasks and red for errors. In addition to this, the post-its must include the name of the team member who is carrying out the respective task as well as the date of entry in each quadrant to allow the observation of each team's evolution throughout the project.

How does the infocenter support continuous development? With the stand-up meetings that are carried out at the beginning of each session of the workshop. This is the point at which each team inspects the progress of the tasks which are currently ongoing using the detailed questions supplied in the previous section. At the end the meeting can continue and/or the necessary modifications can be made which allows the fulfilment of the joint objective which the team undertook. During the meeting, the instructor takes note of any hitches and ensures that the team members stay focused.

On the one hand, this new initiative serves as a vehicle for the participants to acquire a new perspective on project management. On the other hand, it gives them the opportunity to put into practice important skills which they will require in their future professional careers. In the following section we will examine the results of several surveys carried out with the students in order to test the effectiveness of PBL with robots and of the infocenter in project management.

V. EVALUATION OF INFOCENTER BY THE STUDENTS

This baby sumo NXT workshop involved 20 participants (aged between 8 and 15 years old), three of whom were girls. They were separated into 7 teams, who attended six Saturday sessions of 3 hours each.

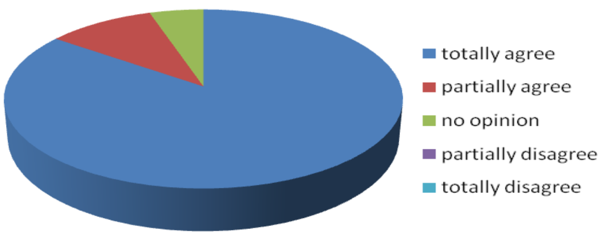


Fig. 5 *Results of the evaluation of the infocenter by the students*

The introduction of the infocenter has been evaluated periodically. We took into account the fact that the participants showed a growing interest in it. They began interacting with the visual board more and more, adding more annotations and even asking if they could include photos of their progress.

In the final session of the Baby Sumo NXT workshop they were asked if the info centre is a good tool for visualising their advancement during the project; 85 per cent of the participants totally agreed with this assessment.

The instructor also decided that this visual tool- which they described as an excellent resource for project management- will be a part of all subsequent workshops.

VI. CONCLUSIONS

In this article the incorporation of an infocenter as a didactic resource, for the first time, in the context of the NXT robotics Workshops has been presented. PBL is the fundamental work methodology employed in NXT Workshops. The aim of infocenter is the improvement of the project management.

The infocenter is an innovative tool within PBL with robots. Its low cost and cheap design allows for easy implementation. In addition to this, thanks to its special design, the infocenter can be used in a wide spectrum of pedagogical activities, unrelated to robotics.

The participants in the Baby Sumo workshop have shown their satisfaction with this new tool. It affords them quick access to the tasks that they need to complete and also to see the continuous progress they are making towards the culmination of the project. The instructor also noted its usefulness in the management of the teams during the workshop, simplifying their job as a monitor.

The change that takes place in the participants won't only have a significant impact on their conception of science and technology, but also on their social relations and their future professional development.

Finally, as part of the process of continuous improvement of the robotics activities, it is possible to investigate whether a digital version of the infocenter would generate even greater benefits than the current model that has just been tested.

ACKNOWLEDGMENT

Kathia Pittí is the recipient of a grant courtesy of The National Secretary of Science, Technology and Innovation (SENACYT) and the Institute for Training and Human Resources Development (IFARHU), of The Republic of Panama.

REFERENCES

[1] H. Goh, and B. Aris, "Using robotics in education: lessons learned and learning experiences," in 1st International Malaysian Educational Technology Convention, 2007, p. 1156-1163.

[2] K. Pittí, B. Curto, J. García, and V. Moreno, "NXT Workshops: Constructionist Learning Experiences in Rural Areas," in Workshop Teaching Robotics, Teaching with Robotics Proceedings of SIMPAR'10, 2010, p. 504-513.

[3] A. Acuña. (2006) *Robótica: espacios creativos para el desarrollo de habilidades de diseño para niños, niñas y jóvenes en América Latina* on FOD. [Online]. Available: http://www.fod.ac.cr/robotica/descargas/roboteca/articulos/2007/frida_robotica_desarrollo_articulo.pdf

[4] S. Boss, and J. Krauss, *Reinventing project-based learning:Your field guide to real-world projects in the digital age*. Eugene, USA: ISTE, 2007.

[5] S. Papert, *Mindstorms, children, computers and powerful ideas*. New York, USA: Basic Books Inc., 1980.

[6] D. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[7] J. Dewey, *How we think*. Lexington, MA: D.C. Heath, 1933.

[8] (2006) TERECoP website - Project Teacher Education in Robotics-enhanced Constructivist Pedagogical methods- [Online]. Available: http://www.terecop.eu

[9] J. López, and I. Leal, *Aprender a planificar la formación*. Barcelona, España: Paidós, 2002.

[10] K. Pittí, B. Curto, and V. Moreno, "Experiencias construccionistas con robótica educativa en el Centro Internacional de Tecnologías Avanzadas," *Revista Electrónica Teoría de la Educación: Educación y Cultura en la Sociedad de la Información* – Universidad de Salamanca, vol. 11, pp. 310-329, Feb. 2010.

[11] E. Martín, and A. Moreno, *Competencia para aprender a aprender*. Madrid, España: Alianza Editorial, 2007.

[12] E. Ruiz-Velasco, *Educatrónica: innovación en el aprendizaje de las ciencias y la tecnología*, Buenos Aires, Argentina: Editorial Díaz de Santos, S.A., 2007.

[13] M. Carbonaro, M. Rex, and J. Chambers, "Using LEGO Robotics in a Project-Based Learning Enviroment," *The Interactive Multimedia Electronic Journal of Computer-Enhanced Learning* – Wake Forest University, vol. 6, number 1, Jun. 2004.

[14] (2009) Robótica Educativa NXT. [Online]. Available: http://robotica.citafgsr.org/index.html

[15] H. Kniberg, and M. Skarin, *Kanban y Scrum – obteniendo lo mejor de ambos*. USA: C4Media Inc., 2010.

Learning Lab - Programming Interaction with Humanoid Robots for Pupils

Alicia Weirich, Carola Haumann and Jochen Steil
Research Institute for Cognition and Robotics,
Bielefeld University
Germany, Bielefeld 33615
Phone: +49 521 106-67108
Fax: +49 521 106-89042
Email: aweirich@cor-lab.uni-bielefeld.de;
chaumann@cor-lab.uni-bielefeld.de;
jsteil@cor-lab.uni-bielefeld.de

Susanne Schüler
University of Applied Science Bielefeld,
Faculty of Engineering and Mathematics
Germany, Bielefeld 33609
Phone: +49 521 106-7471
Email: susanne.schueler@fh-bielefeld.de

***Abstract*—At Bielefeld University pupils of German secondary schools get the opportunity to work with state-of-the-art robots and programming tools in the *teutolab*-robotik. They get in touch with the research field of robot learning in the workshop 'Learning Lab' by working with the humanoid robot Nao in an out-of-school and hands-on one-afternoon course. Its goal is to excite pupils for robotics, to enhance their knowledge and to provide teachers with new stimuli for their school teaching. This is challenging because teaching hands-on experiences in robotics exceeds and expands the standard school curricula, which is a main goal of *teutolab*-robotik.**

I. Introduction

Intelligent technology plays an increasingly important role in our everyday life and will soon include robotic systems for assistive functions. These will provide a large range of supportive functions to humans, for instance in more flexible industrial production or in households to allow for a longer autonomous life of the elderly. Robots must therefor be able to communicate smoothly and on semantic levels with humans. They also need social competences to make them acceptable as assistants. A key feature of such robots will be the ability to learn from humans, which is focus of the lab.

Enable in a technology with such capabilities is a main goal of the focus area 'intelligent systems' at Bielefeld University, which is represented by two high-profile research institutions: The Research Institute for Cognition and Robotics (CoR-Lab, [2]) and the Center of Excellence Cognitive Interaction Technology (CITEC, [1]). Both institutions collaborate for cutting-edge research in the highly competitive field of robot cognition and are dedicated to make cognitive interaction between humans and machines a reality. The Bielefeld researchers aim to make technical systems reaching, from everyday devices to complex robots, more intuitive and user-friendly. To this aim, CITEC offers a broad interdisciplinary research environment including biology, psychology, linguistics, engineering and computer science, while CoR-Lab has strong ties with Honda Research Institute Europe and focuses on research on communication and interaction with humanoid robots. Additional goals are to pursue industry transfer (R&D-projects), public

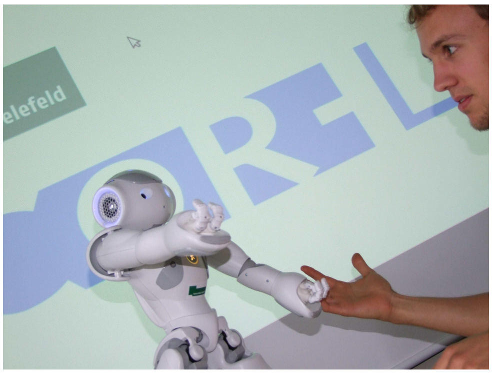


Fig. 1. Interaction scenario between a student and the humanoid robot Nao

understanding of science (industry fairs) and education of young researchers in this high-technology research field. This is realized by educating students in the graduate schools of both CoR-Lab and CITEC, and in the context of *teutolab*-robotik, the hands-on experiences laboratory for pupils.

The *teutolab*-robotik is a joint endeavor of CoR-Lab and CITEC. The research group Cognitive Robotics and Learning headed by Professor Steil at CoR-Lab has developed age-adequate courses reflecting the exciting research questions in human-machine interaction. Cognitive robotics is a highly interdisciplinary field and combines elements from different school subjects like mathematics, computer science, physics, and even biology. Robotics as a single school subject is not on offer, though an increasing number of German schools have robotic project teams. In this sense, *teutolab*-robotik is an exciting complement to the standard school curriculum. Integrating robotics in the curricula of the secondary schools in the longer term is thereto a special challenge and has to be anchored in computer science related subjects.

A further challenge is the required technology. Going beyond very simple toy-like robot kits, most robot platforms are prohibitively expensive and difficult to maintain for schools because of their complexity. Many robot systems need specific knowledge for handling and would require specific qualifications of teachers for their employment in schools. Therefore, pupils have the only chance to get in direct contact with state-of-the-art technology at the out-of-school laboratory at the Bielefeld University. At *teutolab*-robotik that becomes reality, because Bielefeld's researchers are in command of a large number of robotic platforms including the Honda humanoid research robot, the child-like humanoid robot iCub, anthropomorphic heads, hands and a number of smaller biomorphic platforms. Consequently, several skilled researchers, student assistants and staff members are available to choose the eligible platform for the specific tasks and support *teutolab*-robotik in that way. Being closely connected to the actual research, the *teutolab*-robotik stands in the best tradition of the other Bielefeld *teutolabs*, which started in 1999 at the faculty for chemistry [3] and expanded with laboratories for physics, mathematics and most recently biotechnology.

In the long term, our main interest in teaching robotics is the training of highly qualified personnel for fostering innovation in the economy. The topic of learning thereby encompasses a broad empowerment with respect to science and technology. Thereby we want to mediate the relevance of *teutolab*-robotik's learning intents to pupils' everyday life. This includes to develop their interests in technology, confidences in working with technology, and encouragements of future careers in science and technology. We expect as a result an increased self-identification with science and technology of the students. Finally, we want to motivate the *teutolab*-robotik's participants for a study at one of the involved departments at the Bielefeld University later on.

The present paper expands on an previously published general overview on *teutolab*-robotik [4]. The overview featured the concept of *teutolab*-robotik's two workshops for pupils of secondary schools: 'Die Roboterakademie' for youths between 12 and 15 years of age and 'Das Lernlabor' (the 'Learning Lab') for the seniors. In this paper, we focus on the workshop for the senior grades of secondary schools in detail: The 'Learning Lab'. We elaborate in particular on the questions, why the humanoid robot Nao is the ideal platform for this workshop and how we teach pupils why learning is essential for Nao to robustly behave in the real world.

II. THE 'LEARNING LAB'

teutolab-robotik's workshop 'Learning Lab' - in German 'Das Lernlabor' - focuses on the complex research field of robot learning and caters to youths from the age of 16 years up (the senior grades of secondary schools). During the workshop the young people slip into the role of young researchers for one afternoon. The participants are acquainted with research questions in learning robots and approach the topics of this interdisciplinary research field. In small groups they reason, discuss, program, and try to jointly accomplish the tasks assigned in the workshop, where it was a particular challenge to design age-appropriate contents. Finally, the pupils perform experiments in simulation and with real robots. We want to overcome negative prejudices like 'programming is too difficult', and create positive associations with robotics by experiencing in practice how to control robots. While working on topics exceeding curricula and with robots that are usually not available at schools, the pupils learn how to design simple program architectures. Through supplementary character to school lessons, participants strengthen their skills in both problem-solving and teamwork. In order to produce a sustainable effect, we encourage the pupils to engage in the workshops topics beyond their visit at *teutolab*-robotik.

A. The humanoid robot platform Nao

While in the workshop 'Die Roboterakademie' for the younger pupils we field the toy-like robot dinosaur Pleo and the robot dog Aibo, in the 'Learning Lab' we use the humanoid robot Nao (shown in Fig. 1). Nao is a state-of-the-art humanoid platform that is also used in many research projects in human-robot interaction at Bielefeld University, for example the European project HUMAVIPS [1] [5]. What does Nao offer for pupils' first contact with robots in the workshop 'Learning Lab'? - Nao is a humanoid robot developed by Aldebaran Robotics SA in France with a height of 58 cm and a weight of 4.3 kg. It is endowed with touch sensors on its head and feet, with stereo microphones, loudspeakers and two color cameras, with sonar, distance and acceleration sensors, as well as with a gyrometer. All in all, Nao is equipped with 27 degrees of freedom (DOF). Nao comes in two different designs: RoboCup and Academic version. At Bielefeld University we use the academic edition of Nao in research projects and in education at the *teutolab*-robotik also. As a result, in April of this year Aldebaran Robotics announced the Educational Partnership Program (EPP) with Bielefeld University by presentation of an EPP-award. The graphical user interface (GUI) of the programming software Choregraphe is intuitively operable and delivered with NAO Academics Edition as a standard feature. It has interfaces to programming languages like C, C++, and URBI that also allow for a more flexible scientific use. Nao's embodied sensors and actuators and its software platform thus provide a very powerful environment for the *teutolab*-robotik workshop. Finally, Nao's visual design is very appealing, which is very important for motivating pupils to work through the afternoon.

B. Course goals

teutolab-robotik's courses present a survey of robotics with different platforms, robot behavior, perception, navigation and teleoperation. The participants are introduced to the complexity and the difficulties of robotics, but also of the fascination, the variety and the potential of this topic. Pupils get practical appreciation for their capabilities to control robots. Although robots are already very complex machines, their sensory and motor systems are highly constrained compared to that of a human. These differences are constituting for the conceptual layout of the *teutolab*-robotik's courses, because the core

[1] HUMAVIPS - Humanoids with Auditory and Visual Abilities In Populated Spaces

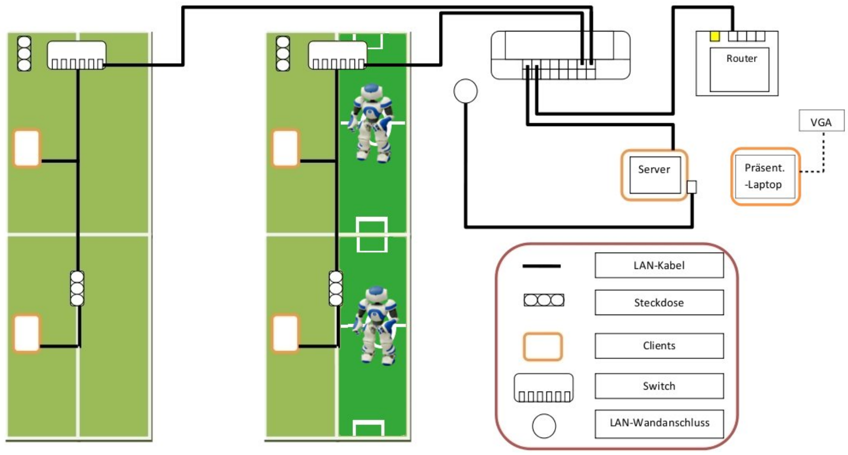


Fig. 2. Set-up of the workshop. For simulation and programming work, there are four groups, for Nao operation two with one instructor each.

theme of the workshop is to convey why robots need to be capable of learning to achieve flexible behavior similar to that of the human. In this way, the participants gain insight into the research field of robot learning at the 'Learning Lab'.

In the workshop, pupils are motivated to analyze their own human behavior. They ponder how they themselves learn or how they perceive their surroundings. Afterwards, they try to apply their knowledge about the human behavior to robots. That is, they follow an important principle of CoR-Lab's and CITEC's researchers: To create robots which adapt to humans and not vice versa.

There is no previous experience required to participate in the workshops. Groups have different previous knowledge and come from courses of all levels in computer science and partially from not subject-specific courses like biology. They have respectively varying expectations for the workshop. We comply to these different educational attainments of the participants by modifying the degree of difficulty of the workshop. If someone is well grounded in the programming system, which is used in the course, the instructors react and assign appropriate tasks of different difficulty.

C. *Course structure*

The course is planned for three hours and consists of four main modules comprising 8 submodules (shown in TABLE I). The workshop is held by two course instructors, mostly students at the Faculty of Technology at Bielefeld University, who work as student assistants at *teutolab*-robotik. They are continuously trained in didactically and technical issues. Due to their own experiences, they can answer questions about computer science and robotics and can also tell the participants more about studies and life at the university in informal conversation after the workshop or in the break. In the 'Learning Lab', the maximum attendance is limited to eight participants. They start in four groups of two each for the programming and preparatory parts, organized in the workshop set-up as illustrated in Fig. 2. For the hands-on part with Nao, participants are re-organized into two groups, each equipped with one Nao. This ensures a quite individual supervision with one course instructor as contact person per group.

TABLE I
OVERVIEW ON THE RESPECTIVE 'LEARNING LAB' COURSE MODULES.

Section	**Module (submodule no.)**
Introduction	General Introduction (1) Pawn chess (2)
How to program a robot?	Introduction to Choregraphe and Webots (3) Practice Choregraphe/simulation (4) Practice Choregraphe/reality (5)
Programming assignments	Rock, Paper, Scissors (reality) (6) Rock, Paper, Scissors (internal simulation) (7)
Conclusion and feedback	Discussion and feedback (8)

At the beginning of the workshop, the participants sit in two rows at the tables looking forwards to the presentation wall in front of the seminar room (left side in Fig. 2). Only during the first submodule 'General Introduction', they have the situation of teacher-centered teaching. This module lasts about twenty minutes. Afterwards, the participants work independently, sitting in pairs at their four working places.

During the first main module the attendees familiarize themselves with the workshop's topic by discussing and playing games. They have to get a feeling for the specific challenges and potential applications, because the topic of the *teutolab*-robotik's course is not part of their everyday life so far. In the second main module the participants learn and practice how to program a robot. To introduce even participants without any previous knowledge in programming languages to robotics, solely graphical user interfaces (GUIs) are used. In the third main module the attendees solve programming assignments in teams. They have to combine their new knowledge about the workshop's topic and their just gained experiences in controlling the robots. Finally, participants review what they have investigated to deepen their understanding. In addition, in the fourth main module the participants are requested to

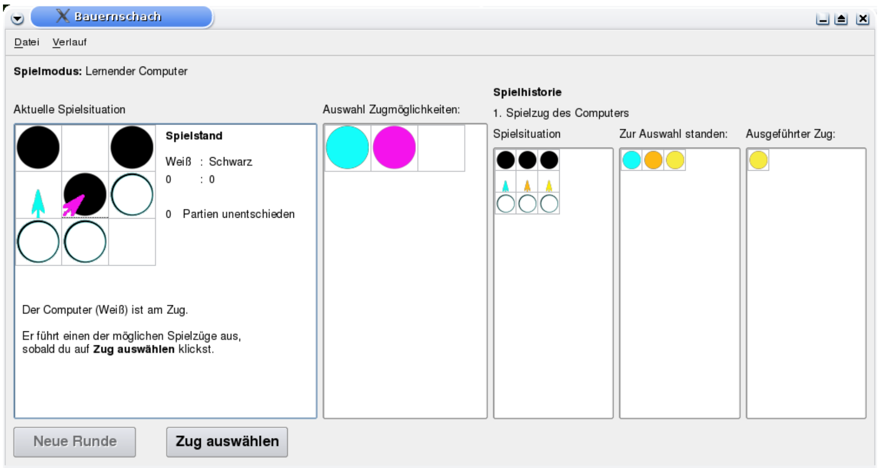


Fig. 3. Illustration and visualization of the computer's learning process when playing the simple board game 'pawn chess'.

give feedback about their experiences during the workshop for evaluation purposes. Exploring the humanoid robot Nao, they comprehend the importance of providing robots with the ability to learn. Using the example of a game strategy that is transfered to the robot, they discover the principle how robots can learn. In teamwork they program the learning strategies to test these learning paradigms on Nao. The respective submodules of 'Learning Lab' cover the following contents in detail:

1) General introduction: At the beginning of the workshop, the participants receive general information about the *teutolab-robotik* and the hosting institutions CoR-Lab and CITEC. The course instructors introduce Bielefeld's robot 'family', in other words the variety of platforms, and explain some of the different robots and their roles in our research. The attendees are familiarized with learning robots in general. They learn about the recent state of research with examples from Bielefeld and are requested to think about some exemplary applications.

2) Pawn chess: In this submodule the pupils play the game 'pawn chess' at the computer. Goal of playing this simple board game is to get an idea how machines like computers or robots can learn. The underlying principle of this game is called 'matchbox computer'. The chess board layout is pictured in Fig. 3, on left side. At first, the participants play some rounds against each other in two by two at the computer to familiarize with the rules: Each player has three figures ('pawns') and must try to go forward to the other side. Moving in alternation, the players can advance forward, if there is spare, or can throw an opponent by removing the opponent's figure diagonally in the front of the player. The winner is, who reaches the opposite side first [6].

In the next step, the pupils play against the computer in three different modes: 1) 'computer without strategy', 2) 'learning computer' and 3) 'computer with strategy'. In the first mode both partners of each team play some rounds against the computer. They recognize very fast that the computer looses almost always. In the second mode the computer learns while playing the game. The computer's learning state is displayed in Fig. 3. The left field shows the current game situation. Possible moves for the players, who's turn is it, are displayed by means of colored arrows. The second field shows the selection of possible moves with color coded circles. The three smaller fields at the right hand side document the history of the game: the game situation, the possible choices and the selected move. Without help of the instructors the pupils should recognize how the computer learns based on the history: If the machine wins with a specific selected move, the pool of moves is extended by this winner move. If the move leads to loosing the game, this playing option is removed from the pool. In the case of stalemate the pool remains unchanged. In the last playing mode, all participants play some rounds against the 'computer with strategy'. The computer chooses its moves according to the principle of contingency and players ascertain that they can not win as easily as before, when the machine had not learned yet. Concluding this part, the course instructors explain the process of the computer's random selection of moves to the participants.

3) Introduction to Choregraphe and Webots: The course instructors introduce how to use Aldebaran's proprietary interface Choregraphe to control Nao (Fig. 4) and the simulation tool Webots. Choregraphe lets NAO's users create and edit different movements and interactive behaviors in a very simple way [8]. It comes with a library with different movement units that comprise the pupils' tools for the following four submodules. A drag-and-drop mechanism allows to place chosen units from the library in the Choregraphe desktop. Control programs are created for connecting their inputs and outputs. To this aim, all units have specific inputs and outputs to organize the flow of the data and control signals. For execution of units, the 'PLAY'-input of the first unit has to connect with the programming start node at the left side of the workspace. The same applies to the output of some unit,

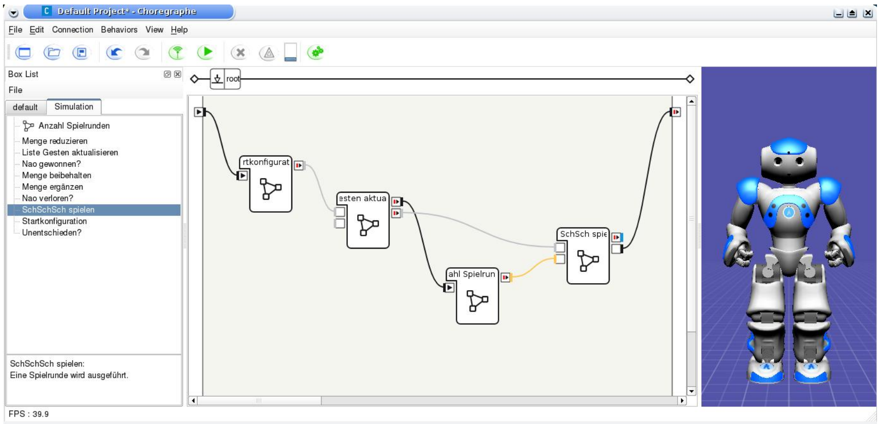


Fig. 4. The graphical user interface Choregraphe by Aldebaran Robotics SA. employed by teutolab-*robotik* to control Nao.

which has to connect to the right workspace's side for program termination. Some of the unit's properties and settings are configurable: This can be done by opening the editor directly with clicking on the unit's surface, or on the tool symbol at the unit's left bottom, if there is one. The simulation of Nao's movements in Choregraphe shows only a visualization of the kinematics, that is the geometric configuration of the robot, but does not simulate the whole physics. For a better illustration, we have connected the robots to the physics based simulation Webots [7], using the low-level interface NAOqi. Nao is shown in this virtual world environment and its behavior can be tested in a more realistic way. Still, motions on the real Nao differ from the Webots simulation, because there are differences in the transmission speed, but Nao does for instance fall over quite realistically during football playing. The pupils like that very much. The difference between simulation and real world is pointed out by the instructors, because it is important in the submodule 5, where the pupils get in contact with the real Nao.

4) Practice: Choregraphe/simulation: The participants now try to score goals in a soccer scenario with Nao using the combination of Choregraphe and Webots visualization. This task achieves a better feeling for moving the robot and participants become more sensible for working with the software. It prepares their interaction with the real robot in the next step.

5) Practice: Choregraphe/reality: If the tasks in the simulation module are mastered, the participants can transfer their commands to the real robot. Working with Nao is realized in two groups of four. We require that one pupil takes care of his or her Nao with respect to safety, for example he or she makes sure that it does not plunge from the table. After connecting with the real robot, the soccer scenarios are executed with the Naos. The pupils are made aware of the slight differences between robots actions in simulation and reality and of accounting for safety aspects.

6) Rock, Paper, Scissors (reality): Now the pupils are ready to implement their knowledge from the first sessions about learning robots. Thereto, they play the game 'Rock, Paper, Scissors' (plus the figure fountain) with the humanoid robot Nao. After explaining the game rules in general, the participants are asked to develop a program for learning by using the program Choregraphe and execute it afterwards. The robot is supposed to learn the winning strategy while playing 'Rock, Paper, Scissors' against the participants. Remembering the learning strategy of the pawn chess game the pupils transfer the principle to the Choregraphe workspace, shown in the middle part in Fig. 4. At the left side there is a library with different pre-programmed modules. Especially for this workshop part student assistants programmed the modules for the pupils. The participants can choose between these module, bring those to the desktop per drag and drop, connect the modules, and create a program for Nao's learning process in this way. At this point it becomes apparent, if the participants to appreciated the principle of learning machines. This part also illustrates the big challenge of implementing robust visual perception. Using a pre-programmed gesture recognizer, that we import from our research projects, pupils find out that still a definite positioning of hands is needed to have the robot recognize the gestures properly. For the configuration of Naos' cameras settings (contrast, luminosity, etc.) Telepathe is used. Telepathe is an application which allows to make the set up with robot's feedback. It shows what the robot is seeing. Nao's hand has only one degree of freedom. So it is not so easy for the pupils to recognize its gestures clearly. They must observe Nao's gestures exactly. The gestures of both Nao and humans, are illustrated in Fig. 5. The participants should try to make their own gestures similar to the presented ones. By rotation principle all group members have the chance to play some game rounds against Nao. All in all it should be not more than 40 rounds. After finishing the game, the course instructors ask the participants how the scores are (pupils can read out the scores from their laptop desktops).

7) Rock, Paper, Scissors (internal simulation): Finally, the participants transmit their program for playing 'Rock, Paper, Scissors' in an internal simulation. This makes possible to play a range of simulated matches in a short amount of time.

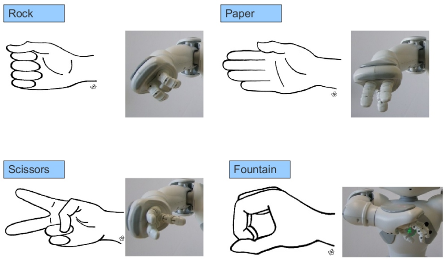


Fig. 5. Guideline for the figures of Rock, Paper, Scissors and Fountain made by human's and Nao's hands

The pupils need to find out how a winning strategy may be structured in this particular case. They write code for the new situation by drag and drop the modules from the library. With the new program they let Nao learn on-line with more game rounds (about 100) than in reality. After finishing this part, the course instructors ask for the new scores. The pupils explain which gestures are winning and which gestures are loosing. The solution is found in cooperation between pupils and instructors. Normally, the game 'Rock, Paper, Scissors is played only with this three figures. The result is, that you can win with all figures exactly once. Because of the extension by the fourth figure, the fountain, the game get an imbalance. In this case you can win with paper and fountain once more than with the other two figures. The winning principle is shown in TABLE II.

8) Discussion and feedback: In the last submodule the attendees summarize what they have grasped during the 'Learning Lab' workshop, reflect their experiences and conclude the main contents. The method that we use aims to make the participant remember the golden thread throughout the workshop: Beginning with the learning computer while playing pawn chess right up to practicing the game strategy of Nao with Rock, Paper, Scissors. The course instructors ask the participants to write down at the white-board their first ideas regarding the workshop and using the letters of the word 'Lernlabor' ('Learning Lab') like in scrabble. Finally, we request the pupils' and also the teachers' feedback about their experiences gained throughout their visit in *teutolab*-robotik. With this feedback we continuously evaluate our workshops.

TABLE II
VALUE OF THE GESTURES TO EACH OTHER.

plays against	**Rock**	**Scissors**	**Paper**	**Fountain**
Rock	o	+	-	-
Scissors	-	o	+	-
Paper	+	-	o	+
Fountain	+	+	-	o

Key: + wins, - loses, o drawn

III. CONCLUSION

Since its opening in June 2009, *teutolab*-robotik had more than 680 visitors in about 70 courses. Nearly 38 percent were female. That proves that robotics can be fascinating for both genders, boys and girls. The workshop 'Learning Lab' also contributes to other special programs for female pupils of senior grades of secondary schools, for example there is the annual 'pea*nuts-Herbsthochschule'. In four days they have the chance to get experiences in three disciplines: Physics, mathematics and computer science/technology. In that program, the *teutolab*-robotik's workshop is part of advertising the study programs at the Faculty of Technology at Bielefeld University. A further program is Bi:tasteMINT (a federal program) [9]. Young women participate in this special program like in a trainee program in companies. Also in this program, the 'Learning Lab' is provides an impression and advertises our research. A further special offer of *teutolab*-robotik is that small groups of pupils can participate in so called 'profession orientating measures', which is a full day visit comprising *teutolab*-robotik, a visit to CoR-Lab and CITEC labs, a meeting with young researchers and getting information about the study program. For this activities the course structure is flexible, so that we can vary it in time and degree of difficulty depending on the participants' necessities. Although it is not necessary to have any previous knowledge for the visit at *teutolab*-robotik, we still inquire the participants' state of knowledge in computer science.

In total, the cooperation projects with local schools create a win-win situation: By involving the teachers, they get information how the *teutolab*-robotik can be implemented in the curricula. In future, we plan to provide teachers with well-elaborated information material for preparation and in their lessons to let them implement topics of *teutolab*-robotik in their school teaching easier. By modifying the workshop contents towards an even more modular and flexible structure, we also want to better adapt to the knowledge standards of each pupil group. Additionally, we interchange with teachers with the goal to make *teutolab*-robotik more sustainable.

It is a special challenge to translate contents of the highly-competitive and complex research field of cognitive robotics into the format of an one-afternoon workshop for pupils. Yet, we think that most participants go home with a general overview about learning robots and human-robot interaction. If they are really interested in robotics research, they may return for a study or a internship.

ACKNOWLEDGMENT

This work has been supported by the Research Institute for Cognition and Robotics (CoR-Lab), the Center of Excellence Cognitive Interaction Technology (CITEC) of Bielefeld University and the Ministry of Innovation, Science and Research of the State North Rhine-Westphalia. We would like to thank M. Johannfunke and M. Götting and many student workers for their substantial effort in providing and programming the experimental environments.

References

[1] Ritter, Helge J., and Sagerer, Gerhard, *Excellence Cluster 'Cognitive Interaction Technology' - Cognition as a Basis for Natural Interaction with Technical Systems*, it - Information Technology, p.112-118, 2009

[2] J. J. Steil, G. Sagerer, H. Ritter, and E. Körner, *Humans and Humanoids - Perspectives on Research in Cognition and Robotics*, KI - Künstliche Intelligenz, no. 4, pp. 33-36, 2008

[3] Kohse-Höinghaus, K., Königer, F., Herbers, R., *Nanochemestry for high school students: Hands-on experiments and informal education in the teutolab at Bielefeld University*. Chemistry today, 25(2), 50-53, 2007

[4] A. Weirich, S. Schüler, C. Haumann, J. J. Steil, *teutolab-robotik - Hands-On Teaching of Human-Robot Interaction*, Proceedings of SIMPAR Workshops 'Teaching robotics and teaching with robots', Darmstadt, pp. 474-483, 2010.

[5] http://humavips.inrialpes.fr, Jun. 2011.

[6] http://fuzzy.cs.uni-magdeburg.de/studium/swprak/ws9900.html, Oct. 2010.

[7] http://www.cyberbotics.com, Jun. 2011.

[8] http://www..aldebaran-robotics.com, Apr. 2011.

[9] http://www.tastemint.de, Jun. 2011.

A Paradox in the Constructive design of Robotic projects in School

Javier Arlegui[1], Alfredo Pina[2], Michele Moro[3]

[1]*Psychology and Pedagogy Department, Public University of Navarra*
Campus Arrosadía, Pamplona, Spain
arleguip@unavarra.es

[2]*Mathematics and Computer Engineering Department, Public University of Navarra*
Campus Arrosadía, Pamplona, Spain
pina@unavarra.es

[3]*Department of Information Engineering, University of Padova*
Via Gradenigo 6/A-B, Padova, Italy
michele.moro@dei.unipd.it

***Abstract*— In the training of teachers for teaching robotics at the primary school level, the methodological aspects of teaching and learning are important. Constructivist methodologies and project-based learning are two "quality" tools that are proposed to the teachers (in training courses) for the design of lesson plans. But, using them we can design constructivist teaching sequences which, although progressively lead to the resolution of real and complex situations, paradoxically may not lead to a parallel progression in the learning of robotic techniques.**

We emphasize here this paradox, showing two paradigmatic examples of constructivist lesson plans for the same theme "No-driver bus ...". Only the second one guarantees parallelism between the increasing semantic complexity of the problems and the positive gradient in the syntactic component of the robot programming.

***Keywords*— Educational robotics; Robotics in school; Teaching and training for robotics; Constructivism and robotics; Project-based learning and robotics; didactic approaches in the teaching of robotics in elementary school**

I. Introduction

Educational robotics is having a significant impact in the development of education at all levels, both primary and secondary. From the methodological point of view very often the literature refers to constructivism/constructionism as a fundamental guideline of 'good practice' in the use of robots in school classes. Particularly at senior secondary level, the literature offers also a wide range of proposals and examples that go beyond the usual applications in the framework of technical vocational schools, proposing experiences of valid multidisciplinary educational content. In this context, the authors were involved in a European project aimed to define robotic-enhanced teacher training actions where the emphasis is given to the robot as a teaching/learning tool with a broad spectrum of application [1].

Scientific education currently lives a critical moment, particularly in Europe, and a huge attention is devoted to enrich curricula to encourage the attraction of scientific subjects by younger generations. Starting from primary school in that direction seems compulsory for the success of these initiatives.

Educational robotics is widely regarded as a powerful engine to promote the interest for science and technology, and therefore a issue of correctly introducing robots in primary school has arisen. The literature shows, particularly from the experiences of the first pioneer teachers using robots in class, that there are two type of problems: one issue is the robotic architecture and the other issue is the robot programming 'philosophy'.

For the first problem, some teachers solve the relative complexity of some robotic kits choosing completely mounted robots and focusing almost exclusively with the strategy to the control the robot (this is for example the case of the well know Bee-Bot). Another solution is to use a flexible kit like LEGO Mindstorms NXT but providing, possibly different, completely or mostly mounted robots to reduce the complexity of the manual construction.

This paper deals only with the second part of the problem at primary school, the programming level, showing a teacher training experience conducted in Spain. This made evident the importance of using the real, live experience of pupils to maintain a fruitful parallelism between the increasing complexity of problems to be solved and the increasing knowledge in the chosen programming language domain.

Programming the tasks that a robot can perform with the use of sensors is an excellent example of the writing of a hypothetical-deductive type of text. Thus, programming robots can help students to build their formal thinking in the "Piagetian" sense, one of the main goals in the last stage of primary education (11-12 years).

The great advantage of programming robots is that it can be organized didactically as an exploratory writing, where the robot's behaviour provides immediate feed back that helps the student to correct the errors of coherence in the program (and correct, thus, their way to think ...).

It is therefore important that the student can use a programming language that has a close correlation between the syntactic expression of the tasks and the sequential behaviour of the robot.

The LEGO's NXT-G iconic language is well suited to the earlier proposal. An icon in this language is a clearly recognizable "block", which corresponds to a robot's behaviour clearly recognizable, whose execution makes a transition between well defined states. In cases in which this correspondence fails, as we shall see later, the NXT-G programming can lead to real cognitive problems for students. And it also causes difficulties for the teacher to imagine alternative structures of programming to restore this syntactic-semantic correspondence.

II. SOCIAL AND EDUCATIONAL CONTEXT OF THE PROJECT

The childhood education and primary school "San Jorge" is located in Pamplona (a city of 200,000 inhabitants, capital of the region of Navarra, northern Spain). This institution is surrounded by the district of "San Jorge", an area of 12,000 inhabitants, built in the industrial outskirts of north-west of the city. It wants to integrate the multiethnic population of the district through an inclusive education of quality for all "... valuing diversity as an enriching element of the teaching-learning process and thus favouring human development ..."

"San Jorge" school has been recently involved in a robotic-enhanced project satisfying the desire to incorporate a science and technology oriented project in order to counteract its identification as a school with a merely humanist and social orientation due to its multicultural characteristic in this disadvantaged area.

The design of the project followed this set of general objectives, both for students and teachers:

• A robotic education for everybody;

• Based on the special skills of robotics to promote the development of formal thought;

• A constructivist teaching and learning;

• A problem based teaching and learning;

• A teacher training program of the center, supervised by professors from the Public University of Navarra

• A project complementary to other projects developed by the school, such as: inclusive teaching and dealing with diversity.

III. TWO PEDAGOGICAL LAWS TO INTRODUCE ROBOTS AT SCHOOL IN THE EARLY STAGES

The objectives abovementioned lead to two basic methodological approaches for the design of robotic-enhanced teaching units:

A. *Designing a process of constructivist teaching and learning, according to the theory of Piaget and Vygotsky.*[2], [3].

This kind of teaching unit is designed as a progressive series of problems: each problem causes an "unbalance" in the student's initial cognitive state, which asks for a cognitive effort of adaptation (assimilation - accommodation) to a new "balance" [4].

To enable the real constructive work of the student, she should be able to use some prior knowledge to solve a problem, i.e. problems should be at least partly recognizable. Thus, if a first problem P1 is of a level A and requires rebalancing to a higher level B, the successive problem P2 should start from a slightly lower level than B (say "B minus") to accomplish a stable cognitive growth (Fig 1). The same for a successive P3 with respect to P2.

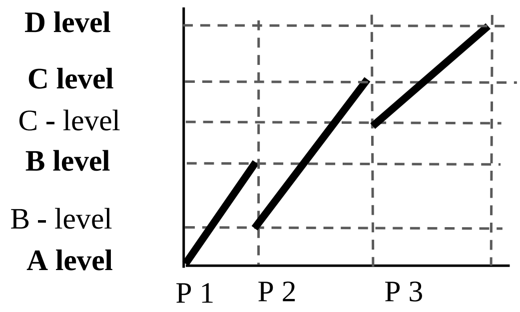


Fig. 1 Constructivist sequence

This approach requires that the constructivist work of the teacher consists in designing and applying each teaching unit regarding a certain theme or context as a sequence of problems P1, P2, P3 ... with these characteristics.

B. *Designing a project-based teaching and learning* [5].

This means that the teaching unit must have an applied nature, and that problems should gradually been formulated and motivated by the "reality" and not merely exercises of a model application. Following this idea, the constructivist work of the teacher in a project-based teaching will consist of designing each instructional unit (theme or context) as a sequence of problems P1, P2, P3 ... of increasing complexity in the real world.

IV. THE FIRST DESIGN OF THE TEACHING UNIT "NO-DRIVER BUS"

Applying the two previous methodological approaches, we have designed for school teachers in "San Jorge" a unit called "No-driver Bus". The chosen robotic architecture was LEGO Mindstorms NXT where the robot simulates a (simplified) bus moving on a linear path. The experimental progression is spread over a number of problems in contexts progressively more and more complex. The common goal can be explained as the designing of the path, and behaviour, of a bus without a driver to perform a passenger service along a highway.

C. *The sequence of problems.*

For this we have proposed a "constructive" sequence of four problems, corresponding to four different cognitive level scenarios, described below:

Problem 1

"*The* bus *must start from point A and travel for 60 cm before stopping at P1, then it must travel for 100 cm to stop at P2 and finally an additional 40 cm to reach the end of the route at point B. Each stop takes 5 seconds*"(Fig. 2)

In this enunciation of the problem, data are formulated in a robot-centered logic: in fact they are given as relative distances and thus they can be easily transformed into angles to be used as parameters in the basic movement command (the so called MOVE block of the iconic language NXT-G).

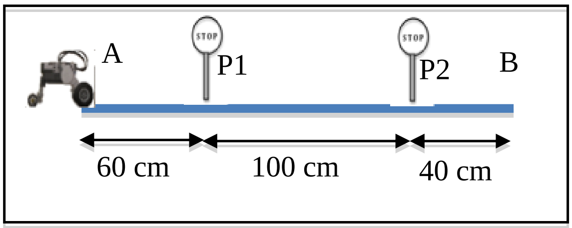


Fig. 2 Problem 1

Problem 2

"*The* bus *must start from point A and stop at P1 and P2 on the way on, then it must go back from B to A, stopping at P3. Each stop takes 5 seconds and its position is given on the chart* (Fig. 3)"

This time the text of the problem provides data in a designer-centered logic, because the given Cartesian coordinate of the stops must be transformed to relative distance in order to be used as 'operative' values in the used programming language.

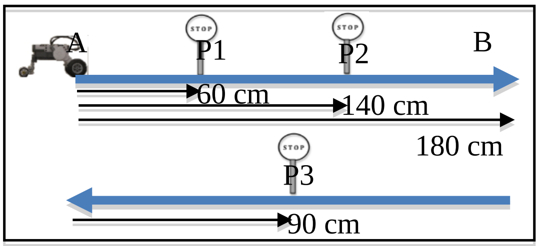


Fig. 3 Problem 2

Problem 3

"*Set up a* bus *to do the tour of the given model"* (Fig. 4).

Now data regarding distances no longer appear in the text of the problem and they should be taken from the model, i.e. a student is requested to measure the appropriate distance on the model. The model is not simply a representation of the experiment but becomes a intermediate representational space between the text (of the preceding problems) and the real world.

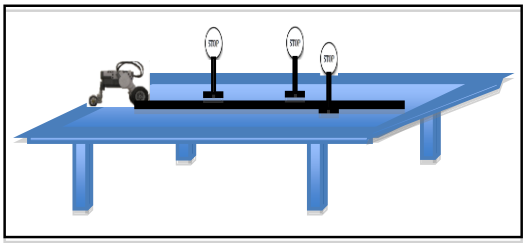


Fig. 4 Problem 3

Problem 4

"*Set up a* bus *to do the tour of San Jorge Street shown on the map*" (Fig. 5).

Now reality is much closer to the exercise: data should be directly taken from the real scenario. Students should go to 'Calle San Jorge', decide where to put the stops and take appropriate actions (the path of the robot, then, is designed on a suitable scale).

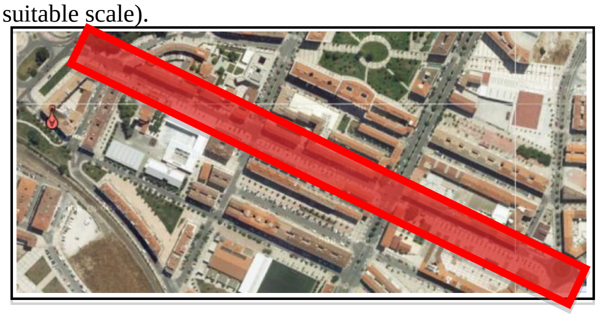


Fig. 5 Problem 4

D. THE PARADOX OF THE PREVIOUS DESIGN

The previous design of the "No-driver Bus" teaching unit performs acceptably when compared with both the teaching laws presented in section III because you can recognize the constructivist progression but also the increasing level of complexity the real world can suggest. Through the presented problems students progress adapting the robot's capabilities to the increasingly realistic conditions of a real urban travel. They finalize the project knowing various aspects of the involved population, of their activities, schedules and travel along the real Calle San Jorge. The final design of the robot bus travel could be a real and complex response to the needs of distribution of citizens in the considered urban area.

But when we see the actually task performed by the robot in the subsequent situations, we see that is essentially the same. The increasing complexity of the problems are always resolved with the same level of elementary programming because in all cases it is reduced to a repeated use of the MOVE and STOP basic blocks.

Therefore we can argue that in this type of design there is the intrinsic paradox that the increasing semantic complexity is not accompanied by a corresponding progression in the formal complexity of the programming task. Whereas students can learn more about transportations within their district, they do not learn anything new about robot programming and thus they do not exploit all the cognitive potential of the used command language.

The result of this reflection is that we need to add a "third teaching law" to the previous two, that could be expressed as follows:

"Designing a teaching and learning process based on the increasing complexity of the robot programming tasks when implementing increasingly complex behaviors of the robot".

V. A SECOND DESIGN OF THE TEACHING UNIT "NO-DRIVER BUS"

Now integrating this third criterion with the previous two, we have designed for the same group of teachers a second unit called "No-driver Bus - 2" through the constructive sequence of five problems described below.

Problem 1 is a transitory one: it has the simple aim to justify the introduction of a sensor as a component on which to make decisions. A student already knows how to control the robot to make it move through given distances and has a first idea of the importance of a sequence of commands; but she has also the direct experience that bus stops can be optional and stopping might be requested by the traveller. So the problem leads the student to relate the stopping of the motor to a condition based on a sensor. Problem 2 is a reformulation of Problem 1 with the adding of a small but logically important detail that produces a solution which corresponds more strictly to the control logic. In this solution a conditional wait is substituted by an 'active' permanent control of the stopping condition which is closer to the student's perception. Problem 3 shows how increasing requirements, such as the approach with reduced speed to the stop, can actually produce a more advanced control program, improving the previous solution.

A. Problem 1

"Designing a bus which stops at the request of a traveller: the request is represented by posing a hand in front of the robot at a distance D <30 cm".

Now the bus stops are no longer in fixed positions. The proposed problem is formulated so that it is necessary to use sensors for the solution: in this case, the student must incorporate and program an ultrasonic sensor. A possible core of the solution in the iconic NXT-G language is given in Fig 6.

The complete solution (fig. 7) must include the stopping for a given time and the repetition of the entire sequence in an undefined loop.

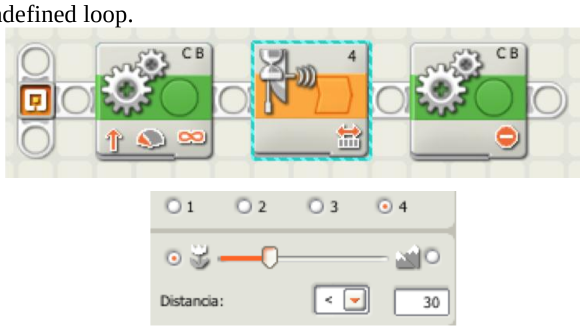


Fig. 6 Problem 1: the core sequence

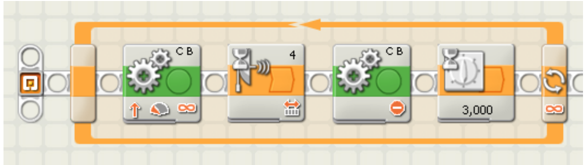


Fig. 7 Problem 1: the complete solution

The fact that the control flow is stopped waiting for a condition regarding a sensor might surprise the student: her personal experience is based on the behaviour of a bus driver who is continuously taking actions and monitoring the situation around the bus. Thus the next step is to suggest a less passive solution.

B. Problem 2

"Designing a bus which stops at the request of a traveller: the request is represented by posing a hand in front of the robot at a distance D <30 cm. Act as a driver who is looking for a travellers' request while moving the bus".

You must consider that the programming instruction "move the robot until ..." implies the use of the MOVE block with a meaning corresponding to a "special" treatment of the NXT interpreter. In fact, when you set as 'indefinite' the time/angle parameter of the motion (see fig. 6), you are not setting an action that corresponds to the transition between two distinct and well defined states S_i and S_j, as it would be in the case of a finite (in time or in space) move command. Actually the interpreter activate a (logically separate) thread indefinitely piloting the motor while the main thread continues to execute the interpreter on the following commands. In this sense the final STOP command acts as the 'killer' of this separate, previously spawn thread. Another state-oriented interpretation could be that, while waiting for the sensor, the state S_i remains unchanged and this corresponds to a (logical) loop insisting on the same state, whereas the transition from S_i and S_j is labelled by the condition when positively verified (D<30) (Fig. 8).

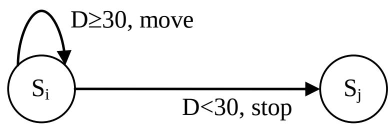


Fig. 8 A relatively complex state diagram

The comprehension of this more complex situation is, in either view, very difficult for a primary student and the solution can 'run away'.

Problem 2 could be solved using an alternative approach closer to the hypothetical behaviour of the bus driver at least in the perception of the student. In this approach the indefinite motion is broken into several micro-movements. So we define a personalized MOVE command (*small forward*) with a very small displacement and executed as an alternative of the STOP command inside an unconditional loop. This small forward corresponds to the action associated with the loop on state Si of fig. 8. The stopping of the bus is signalled by the known requirement (D<30) (Fig. 9).

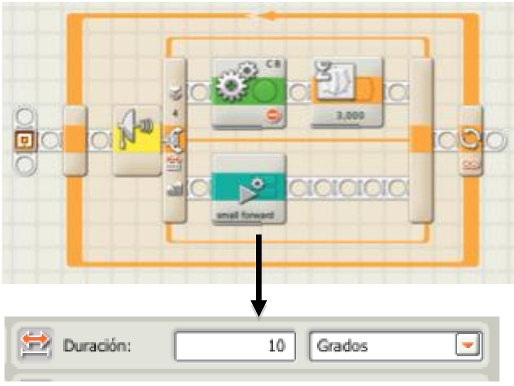


Fig. 9 Problem 2

C. Problem 3

"Designing a bus which stops at the request of a traveller: the request is represented by posing a hand in front of the robot at a distance D <30 cm. Use separate blocks for the sensor and the conditional statement".

NXT-G allows also the independent use of a sensor and a conditional statement, which makes you distinguish more clearly what is the role of the two instructions (the sensor block is of 'operation' type whereas the conditional block is of 'command' type) (Fig 10). Such a separation could be suggested to students as a further improvement, for example saying that the driver not only is aware whether a traveller on the street is requesting the stopping but he can also estimate the distance of the traveller during the approaching phase (observe that the sensor block gives also the distance measure together with the overcoming of the distance threshold).

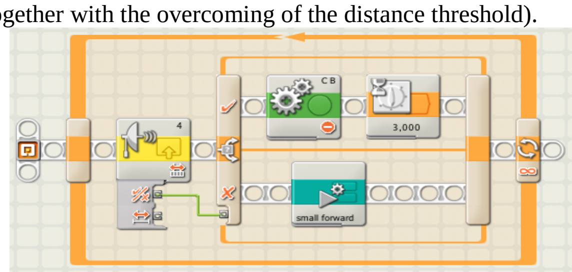


Fig. 10 Problem 3

D. Problem 4

"Designing a bus which acquire the request of stopping from a traveller, and approaches the traveller enough to permit she can get in".

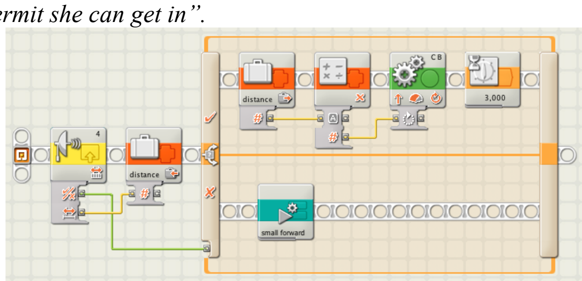


Fig. 11 Problem 4

In this problem, once located the requesting traveller, the bus must go on sufficiently to be close to the person. It is easy to verify that this can (softly) introduce the use of variables. The *distance* variable in Fig. 11 is used to calculate the necessary approaching space, improving the realism of the solution. The example shows also how the sensor output can be used both for logical (comparing with a threshold) and numerical (in absolute term) purposes.

VI. CONCLUSIONS

The sequence of three problems discussed in Section V can be further extended facing other tasks like slowing down when the bus approaches the traveller, turning a brake light on when it reduces the speed, etc. But the previous sequence is enough to demonstrate the great difference between the two constructivist views presented.

We showed that the second one can guarantee the parallelism between the increasing semantic complexity of the problems and the positive gradient in the syntactic component of the robot programming. This is in a nutshell the constructivist teaching model we propose.

This model combines in a dialectical mode a teaching/learning process where robots are "object of knowledge" and a teaching/learning process where robots are "learning tool". The first aspect corresponds to the progression of the formal complexity, the second, the semantic progression. It is possible, and sometimes desirable, to design sequences that focus on one of the two views, but in any case, the teacher should always have clear in mind these two "didactic variables " when designing teaching sequences based on constructivist educational robotics.

ACKNOWLEDGMENT

The authors of this paper wish to thank the primary school "San Jorge" (Pamplona, Spain) and their teachers, especially Juan Jose Aragon and Milagros Lorenzo, for their work and their enthusiasm for carrying forward the innovation project "Robotics education for everybody".

REFERENCES

[1] Alimisis, Arlegui, Fava, Frangou, Ionita, Menegatti, Monfalcon, Moro, Papanikolaou & Pina, "Introducing robotics to teachers and schools: experiences from the TERECoP project", *Proceedings of the Cocstructionism 2010 Conference*, Paris

[2] C. Coll, "Constructivismo y educación escolar: ni hablamos siempre de lo mismo ni lo hacemos siempre desde la misma perspectiva epistemological", *Anuario de Psicología*, vol. 69, pp. 153-178, Facultad de Psicología, Barcelona, 1996.

[3] S. Chaiklin, "The Zone of Proximal Development in Vygotsky's analysis of learning and instruction." in A. Kozulin, B. Gindis, V. Ageyev, & S. Miller (Eds.) *Vygotsky's educational theory and practice in cultural context*, pp. 39-64, Cambridge University, 2003.

[4] M. Bosch, L. Espinoza, J. Gascón, ,"El profesor como director de procesos de estudio: análisis de organizaciones didácticas espontáneas", *Recherches en Didactique des Mathématiques*, vol. 23(1), pp. 79–136, 2003.

[5] J. R. Savery, T. M. Duffy, "Problem Based Learning: An instructional model and its constructivist Framework", Indiana University, Center for Research on Learning and Technology, Tech. Rep. 16-01, June, 2001.

The TiRoLab Concept

Michael Sieb

Robotics Institute Tyrol
Adolf Pichler Weg 23, 6065 Thaur, Austria
sieb@TiRoLab.at

***Abstract*— At "Tiroler Roboter Labor", short TiRoLab, girls, boys, young women and men are introduced to the fascination of robots and, consequently, informatics and mechatronics as a means of building their self-confidence with regard to technical skills. We run a "hands-on lab" and develop robotics workshops for participants from kindergarten to all levels of formal education and beyond. This paper discusses the strategy behind this concept.**

***Keywords*— TiRoLab, robotics, education, kindergarten, school**

I. Introduction

The Austrian state of Tyrol does not differ from many other European regions in that it depends on highly qualified employees if it is to remain a competitive and attractive location for high-tech industries. International studies show that education is a key factor in attracting more young people to technical professions. The Austrian Chamber of Economics suggests offering active vocational training [1].

II. Background

Which profession or course of university study a young person decides to pursue depends on their social and personal environment [2]. A career path is often selected on the basis of stories and experiences related by family and friends.

Technology has acquired a bum rap with the general public: it is seen as a a pursuit for loners and math geniuses [3].

This is due to a general lack of information about technical training opportunities, careers, innovative companies and industry's very real need for skilled workers [4].

The Association of the Austrian Electrical and Electronics Industries in its exploratory study about "Shortage of engineers and choice of study [4]" gave exact recommendations for action (see Table 1).

TABLE 1
Recommendations for Action[4]

Item	Topic
1	Public communication on demand for employees with technical skills
2	Public relations and educational work on technical careers
3	Teachers as a key factor
4	Importance of family members and friends
5	Educational television

The MoMoTech trial [5] reviewed approx. 1.000 technology projects in Germany for their effect and efficacy and then suggested four basic levels of technical involvement (see Table 2).

TABLE 2
Levels of Technical Involvement [5]

Level	Topic
1	Encourage childlike curiosity and technical fascination
2	Convert curiosity into deeper technical interest
3	Taking the initiative to start technical activities
4	Develop a personal preference for a technical career

We decided to use LEGO Mindstorms NXT as our robotics platform for the following reasons:

Nearly everyone, whether child or adult, played with LEGO during their childhood and loved it. Therefore, it is not recognised as an education medium, but is seen as a toy to have fun with. This is a major advantage in bringing kids into contact with technology and arousing their curiosity.

Because it is based on LEGO Technic it is easy to build different robots, even ones that look like animals [6]! There is no need for screwdrivers or other tools, which enhances its appeal for girls.

The NXT kit comes ready to run with three motors and four sensors. They are easy to connect without needing soldering.

If greater mechanical stability or resolution is needed, the NXT kit components can be combined with Tetrix [7], a LEGO Technic compatible system with aluminium elements for construction, metal gears, RC-Servos and DC model craft motors.

A more detailed look at its hardware and software abilities shows the system's enormous flexibility:

Hardware

The NXT kit contains a programmable brick consisting of a 32-bit ARM processor at 48MHz, an 8-bit Amtel AVR coprocessor at 8MHz communicating via SPI, an LCD matrix display with 100*64 pixels, a Bluetooth and a USB2 interface, 8-bit sound supporting sample rates from 2–16kHz, three motor driver ports with encoder inputs and four sensor inputs I2C-capable with two digital IO lines and one analog input.

Input 4 can be used for RS 485 (IEC 61158 Type 4) communication and can run on a rechargeable lithium battery.

Commercial suppliers like Vernier[8], Mindsensors[9], HiTechnic[10], Dexter Industries[11] and Codatex[12] sell high-quality hardware sensors and actuators that can be used out of the box (selected examples are given in Table 3). At the date of publication more than 80 extensions were available. For a current list, please consult the suppliers' websites. NXT-G and RobotC drivers are usually included; some suppliers even support NXC, LeJOS and LabView drivers. With the wide range of available sensors even complex robots and technical experiments can be run.

LEGO provides a free available hardware development kit with all schematics and an open source software development kit of the firmware [13]. This enables users to build their own sensors or actors [14].

TABLE 3
EXAMPLES OF THIRD-PARTY HARDWARE EXTENSIONS FOR NXT

Supplier	Tool
Vernier	25g Accelerometer
Vernier	pH Sensor
Vernier	Turbidity Sensor
Mindsensors	Magic Wand
Mindsensors	3-Axis Acceleration Sensor
Mindsensors	8-Channel Servo Controller
HiTechnic	Compass Sensor
HiTechnic	EOPD Sensor
HiTechnic	Solderless Prototype Board
Dexter	NXTBee (Xbee)
Dexter	dGPS
Dexter	dSolar (2W and 4W Solar Panel, CapBank)
Codatex	RF ID Sensor

Software

The original programming software supplied with the commercial NXT set (or purchased separately for the educational NXT set) is NXT-G, a graphical programming IDE based on National Instruments LabView. Since the set was released in 2006 other open source and commercial programming tools became available. They can be divided into two main groups: those for which programmes are downloaded to the programmable brick (examples see Table 3) and those where the user programme runs on a PC or PDA (examples see Table 4) and communicates with NXT via USB or Bluetooth. PC-based programmes can draw on the PC's full power. The overview is based on an updated review by team hassenplug [15]. Non-programmable PC- or PDA-based remote-control tools were not included, but can be easily located in a webseach.

TABLE 4
EXAMPLES OF NXT PROGRAMMING LANGUAGES BRICK-BASED

Name	Type	Licence	Debug
NXT-G [16]	Graphical	comm..	No
LabView [17]	Graphical	comm..	No
NXC [18]	C-like	OS	No
RobotC [19]	C	comm.	Yes
leJOS [20]	Java	OS	No
nxtOSEK [21]	C, C++	OS	Yes
IAR [22]	C, C++, Graphical	comm.	Yes
NXTGCC[23]	C	OS	Yes

Licence: comm.: commercial; OS: open source

TABLE 5
EXAMPLES OF NXT PROGRAMMING LANGUAGES PC-BASED

Name	Type	Licence	Debug
LabView[17]	Graphical	comm..	Yes
MS Robotic Studio[24]	Graphical	comm..	Yes
DialogOS[25]	Graphical	comm..	No
NXT Python[26]	Python	OS	Yes
Mathlab[27]	m Code	comm.	Yes
Simulink[27]	Graphical	comm.	Yes

Licence: comm.: commercial; OS: Open Source

There are three different products that can be used to run NXT robots in virtual worlds without a physical robotics kit. This enables the development of online robotics courses for kids with computer access but without robotics hardware:

- Microsoft Robotics Developer Studio [24]
- SimLejos with leJOS programmes
- Virtual Worlds with RobotC [28]

If the NXT is used to teach embedded developing, even JTAG debugging can be performed. The brick has to be opened and the JTAG connectors wired. Figures 1 and 2 show the author's NXT brick: JTAG-enabled (ARM and AVR processors) with additional reset button on front. A custom adaptor cable connects the frontside female connector and the standard debug header of the JTAG probe.

Figure 1: NXT with JTAG modification

Figure 2 NXT frontside with 1.27" female connector and reset button

The LEGO NXT kit is widely used for educational purposes (see Table 6). Its projects range from workshops for kids [29], demonstrating complex science projects like the Rosetta landing on a comet [30], to high-tech hardware extensions [31].

TABLE 6
LEGO NXT WORKSHOPS AND EDUCATIONAL PROJECTS

Project and Institution
Cooperation between LEGO and Tuffs University [32]
Carnegie Mellon Robotics Academy [29]
Roberta, Fraunhofer IAIS [33]
Rosetta, ESA [30]
H.A.L.E, University of Nevada Reno [34]
TUMlab, TU Munich, Deutsches Museum Munich [35]
LEGO Beyond Toys, TU Eindhoven [31]
LEGO Engineering from Kindergarten to College, Tuffs University [36]

There are exciting national and international LEGO NXT robotics competitions (see Table 7) that serve to promote technology in the public eye and motivate the competition participants.

TABLE 7
ROBOTICS COMPETITIONS WITH LEGO NXT

Competition
FLL, First LEGO League [37]
RoboCup [38]
World Robot Olympiad [39]
RobotChallenge [40]

III. APPROACH

We use robotics kits as a tool to generate enthusiasm for technology for the following reasons:

Robot kits are a fast and easy way to enter mechanics, electronics and programming. A beginner's achievements become visible very quickly.

Robots are real hands-on items that make technology understandable and let formulas come to life so they are no longer experienced as abstract entities.

The complete development of a robot trains complex system development skills. It begins with a precise identification of the requirements, draft planning, construction, programming, testing, optimisation and finally documentation of the finished robot. This process calls for interdisciplinary thinking and coordination so that hardware, software, electronics and mechanics all work together.

Greater efforts are needed to promote sustainability. This is why the TiRoLab concept starts in early kindergarten using curiosity and the natural fascination of exploring new things to give small children a positive attitude toward technical concepts like robots. As the kids grow up we accompany them through school, deepen their knowledge and demonstrate capabilities for the next step into a technical profession or a university degree in engineering.

Robotics Workshops

A team consists of two kids with two computers and one LEGO NXT robotics kit. Kids should have the opportunity to incorporate their own ideas as well as to act as a team when constructing the robot.

Modern teaching means incorporating gender aspects. A nice side-effect is that girl-sensitive concepts work just fine with boys, but not vice versa [41].

To present the workshop's goal we use storytelling techniques instead of technical problem definitions. For example:

"A frog is hiding in the pond, waiting for a dragonfly. When the frog sees the dragonfly, it jumps out to catch it."

instead of:

"The robot is on standby at position 1 and waits for sensor input 3. If the value exceeds the threshold, motors A and B turn on forward for 2 seconds."

Decorative material like neon bricks, colored paper, feathers and adhesive tape help kids turn technical components into fantasy creations.

Instead of holding competitions, we encourage the kids to run presentations with their robots in order to enhance communication between teams.

Workshop duration depends on the complexity of the challenge and on the age group, e.g. in kindergarten it is limited to 50 minutes. Older kids can attend half- or full-day workshops. Summer camps can even hold big projects that run up to a week.

The result is that kids learn to use a computer as a tool to implement their creative ideas, and not merely as a passive game station.

Diploma and License

When a child finishes a workshop she or he will receive a diploma with a picture of her- or himself, the built robot and details on the workshop. The diploma includes a separate licence (like a driving license) that documents the training level achieved. This is represented by a letter for the particular age group (A-H, see Table 8) and a number that increases with the level of complexity of the challenge mastered (1-17, see Table 9).

With the diploma and license the child can demonstrate its success to family and friends and is thus motivated to proceed to the next level.

TABLE 8
AGE-BASED GROUPS

Group	Designation	Age / years
A	Kindergarten	5 – 6
B	Primary School	7 – 10
C	Junior High School	11- 14
D	Senior High School	15 – 18
E	Apprentices	15 +
F	University Students	17 +
G	Adult hobbyists	18 +
H	Adult professionals	18 +
I	Experts	25 +

TABLE 9
COMPLEXITY-BASED LEVELS

Level	Content
1	Teach-In Programming
2	Simple Sequences
3	Using Display and Sound
4	Loops
5	Read and Display Sensor Values
6	Wait for Sensor
7	Conditional Branch
8	Variables and Calculations
9	Data Logging
10	Data Visualising
11	Robot-to-Robot Communication
12	Robot-to-PC Communication
13	Feedback Loops
14	Build Sensors and Actors
15	Embedded Programming
16	Build Complex Hardware
17	Build Realtime Applications

The maximum attainable level depends on the age group: for kindergarten youngsters it is A3, primary school kids B7, junior high school kids C12, senior high school kids E14, apprentices and above (F to I) up to level 17.

Teachers and Schools

Teachers are important disseminators for the project. The best practice example would be for a robotics lesson to be prepared in school, embedded in the curriculum, and for the kids to build their robot projects at TiRoLab with post-processing of the learning units back at school.

We will run teacher training sessions to enable the teachers to hold their own robotics lesson at school.

We will offer schools a limited number of robotics kits and notebooks that can be rented for one to eight weeks for a nominal fee. In this way they can test robotics lessons in their own classrooms without having to make a big investment.

Parents

We will also take family and friends on board, because social environment is a major factor in kids' development. Parents are encouraged to drop by before the end of the workshop and attend their child's robot presentation.

Information

On the TiRoLab website we will provide information about schools with a technical focus, training opportunities, resources for independent learning, apprenticeships, the high-tech industry in Tyrol, continuing education and technical university degree programs in Tyrol and job prospects.

IV. CONCLUSIONS

While some of the ideas behind the TiRoLab Concept are new, many of them were adopted from and inspired by outstanding projects like those of Tuffs University or the MIT Media Lab with Lifelong Kindergarten and trials like MoMoTech, among many others.

We acquired early experience by running robotics workshops for medical students in the elective course “Theoretical Surgery” and at the Kids and Youth Academy at Innsbruck Medical University running workshops for junior high school and kindergarten kids.

We are working hard to move TiRoLab from the drawing board to a true robotics lab by the start of 2012.

References

[1] „Technikermangel: Risikofaktor für Österreichs Industrie - WKO.at“. [Online]. Available: http://portal.wko.at/wk/format_detail.wk?AngID=1&StID=523391&DstID=0&titel=Technikermangel:,Risikofaktor,f%C3%BCr,%C3%96sterreichs,Industrie. [Accessed: 04-Feb-2011].

[2] K. Hofer, *Motive für die Berufswahl - Eine qualitative Studie*. Arbeiterkammer Wien, Abt. Bildungspolitik, 2009.

[3] G. Rötzer-Pawlik, *Jobs mit Zukunft - Technik*. Arbeitsmarktservice Österreich, 2010.

[4] W. Schildorfer, R. Hasenauer, A. Kurz, und D. Bankosegger, *Technikermangel und Studienwahl - eine explorative Studie*. Vereinigung High Tech Marketing, 2008.

[5] A. A.-D. A. der Technikwissenschaften, *Evaluationsstudie: Monitoring von Motivationskonzepten für den Techniknachwuchs (MoMoTech): Best-Practice Empfehlungen*, 1. Aufl. Springer, Berlin, 2011.

[6] F. Rhodes, *The LEGO MINDSTORMS NXT Zoo!: A Kid-Friendly Guide to Building Animals with the NXT Robotics System*, 1. Aufl. No Starch Press, 2008.

[7] „TETRIX® Robotics: Training Resources“. [Online]. Available: http://www.tetrixrobotics.com. [Accessed: 13-Juni-2011].

[8] „Vernier Software & Technology - Probeware for Science, Technology, Engineering, and Math Education“. [Online]. Available: http://www.vernier.com/. [Accessed: 13-Juni-2011].

[9] „www.mindsensors.com“. [Online]. Available: http://www.mindsensors.com/. [Accessed: 13-Juni-2011].

[10] „HiTechnic Products“. [Online]. Available: http://www.hitechnic.com/. [Accessed: 13-Juni-2011].

[11] „Sensors for Lego Mindstorm NXT Robots | Dexter Industries“. [Online]. Available: http://www.dexterindustries.com/index.html. [Accessed: 13-Juni-2011].

[12] „CODATEX Hainzlmaier GmbH & CO.KG - RFID for LEGO® MINDSTORMS® NXT - Lego Mindstorms nxt“. [Online]. Available: http://www.codatex.com/index.php?en_LEGO_MINDSTORMS_NXT. [Accessed: 13-Juni-2011].

[13] „LEGO.com MINDSTORMS : Support : Files - Advanced - HDK, SDK, BKD“. [Online]. Available: http://mindstorms.lego.com/en-us/support/files/default.aspx. [Accessed: 15-Juni-2011].

[14] M. Gasperi und P. E. Hurbain, *Extreme NXT: Extending the LEGO MINDSTORMS NXT to the Next Level*, New. Apress, 2009.

[15] „NXT Programming Software Overview“. [Online]. Available: http://www.teamhassenplug.org/NXT/NXTSoftware.html. [Accessed: 13-Juni-2011].

[16] „NXT-G“. [Online]. Available: http://www.ni.com/academic/mindstorms/works.htm. [Accessed: 13-Juni-2011].

[17] „LEGO® MINDSTORMS® NXT - Powered by NI LabVIEW - National Instruments“. [Online]. Available: http://www.ni.com/academic/mindstorms/. [Accessed: 13-Juni-2011].

[18] „NBC/NXC - NeXT Byte Codes and Not eXactly C“. [Online]. Available: http://bricxcc.sourceforge.net/nbc/. [Accessed: 13-Juni-2011].

[19] „ROBOTC“. [Online]. Available: http://www.robotc.net/. [Accessed: 13-Juni-2011].

[20] „LeJOS, Java for Lego Mindstorms“. [Online]. Available: http://lejos.sourceforge.net/. [Accessed: 13-Juni-2011].

[21] „nxtOSEK“. [Online]. Available: http://lejos-osek.sourceforge.net/. [Accessed: 13-Juni-2011].

[22] „IAR Embedded Workbench for LEGO MINDSTORMS NXT“. [Online]. Available: http://www.iar.com/website1/1.0.1.0/1483/1/. [Accessed: 13-Juni-2011].

[23] „NXTGCC“. [Online]. Available: http://nxtgcc.sourceforge.net/. [Accessed: 13-Juni-2011].

[24] „Microsoft Robotics Developer Studio“. [Online]. Available: http://msdn.microsoft.com/en-us/robotics/aa731517.aspx. [Accessed: 13-Juni-2011].

[25] „CLT Sprachtechnologie » DialogOS“. [Online]. Available: http://www.clt-st.de/en/produkte-losungen/dialogos/. [Accessed: 13-Juni-2011].

[26] „NXT Python“. [Online]. Available: http://home.comcast.net/~dplau/nxt_python/index.html. [Accessed: 13-Juni-2011].

[27] „LEGO MINDSTORMS NXT software for MATLAB & Simulink“. [Online]. Available: http://www.mathworks.com/academia/lego-mindstorms-nxt-software/. [Accessed: 13-Juni-2011].

[28] „FIRE Website :: Virtual Worlds“. [Online]. Available: http://www.education.rec.ri.cmu.edu/fire/virtual-worlds/. [Accessed: 15-Juni-2011].

[29] „Robotics Academy“. [Online]. Available: http://www.education.rec.ri.cmu.edu/. [Accessed: 15-Juni-2011].

[30] „ESA - ESA Spacecraft Operations - ESA’s Rosetta comet-chaser goes LEGO® - printer version“. [Online]. Available: http://www.esa.int/SPECIALS/Operations/SEMN40NO7EG_2.html. [Accessed: 08-Juni-2011].

[31] „Lego Beyond Toys | Christoph Bartneck, Ph.D.“ [Online]. Available: http://www.bartneck.de/2008/03/09/lego-beyond-toys/. [Accessed: 15-Juni-2011].

[32] „Welcome to LEGO Engineering“. [Online]. Available: http://www.legoengineering.com/. [Accessed: 15-Juni-2011].

[33] „Roberta, Fraunhofer IAIS“. [Online]. Available: http://www.roberta-home.de/en. [Accessed: 15-Juni-2011].

[34] „High Altitude LEGO Extravaganza“. [Online]. Available: http://www.unr.edu/nevadasat/HALE/index.html. [Accessed: 08-Juni-2011].

[35] „TUMlab“. [Online]. Available: http://www.tumlab.edu.tum.de/en/startseite/. [Accessed: 15-Juni-2011].

[36] „LEGO Engineering: From Kindergarten to College“, *youtube.com*, 30-Dez-2008. [Online]. Available: http://www.youtube.com/watch?v=EG-izyXfFHI&feature=youtube_gdata_player. [Accessed: 15-Juli-2011].

[37] „FIRST LEGO League | FLL is the result of an exciting alliance between FIRST® and the LEGO group“. [Online]. Available: http://www.firstlegoleague.org/. [Accessed: 15-Juni-2011].

[38] „RoboCup“. [Online]. Available: http://www.robocup.org/. [Accessed: 15-Juni-2011].

[39] „World Robot Olympiad. United Arab Emirates“. [Online]. Available: http://www.wrouae.ae/intl/en/index.php. [Accessed: 15-Juni-2011].

[40] „RobotChallenge: Home“. [Online]. Available: http://www.robotchallenge.org/. [Accessed: 15-Juni-2011].

[41] M. Wagenschein, *Ursprüngliches Verstehen und exaktes Denken*. Stuttgart: Klett, 1970.

Robotics course with the Acrob robot

Richard Balogh
Slovak University of Technology in Bratislava
Ilkovičova 3, 812 19 Bratislava, Slovakia
Email: richard.balogh@stuba.sk

***Abstract*—Robotics course at the Faculty of Electrical Engineering and Information Technology of Slovak University of Technology in Bratislava is intended for the 3rd year students of bachelor studies. In this paper we describe the course organization and some of laboratory exercises with the Acrob mobile robot. We describe its hardware used for the course and tasks solved. We would like to share our experience with work of students in laboratory and main problems we encountered.**

***Index Terms*—mobile robot, project-based learning, education, ultrasonic distance sensors**

I. Introduction

Slovak University of Technology (STU) in Bratislava is a modern educational and scientific institution. Since its foundation in 1937 more than 105,000 students graduated here. On average around 16,000 students study at the STU every year. At present, the University has seven faculties. All STU faculties offer accredited programmes within a complex bachelor, master and PhD study system.

Course *Robotics* is intended for students of the 3rd year of bachelor studies *Industrial informatics* at the Faculty of Electrical Engineering and Information Technology (FEI STU). Its value is 6 ECTS credits, it consists of 3 hours of theoretical lectures and 2 hours of lab sessions each week during the 12-week semester. The same course is offered also in a distance form of study. Assessment of students is based on their written laboratory reports, brief tests and one written report (40% together) and written final exam (60%) – see Fig.1 for details. Last year, 51 students visited the course.

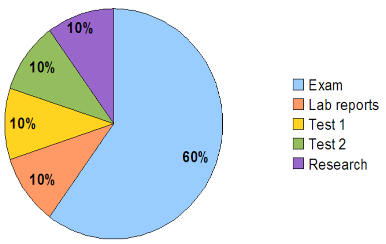


Fig. 1. Evaluation of students during semester.

During the first half of the semester, students are focused on industrial robotics – work with manipulators and compute various transformations and kinematic equations. In the second half of semester, basic concepts of mobile robotics is introduced. As we mentioned before, lectures are intended for students of a broader study branch *Industrial informatics*, so probably most of them will never be involved in the robotics area. Only few of them will develop their knowledge within master studies in *Robotics*.

At present, similar courses are offered at universities all over the world. Comparing our *Robotics* course with e.g. *Mobile Robot Programming Laboratory* at CMU [1], our focus is not only on robot programming – we can also modify robots hardware (e.g. sensors). We also use much simplier robots and work more with hardware interfacing. Another similar course is offered at the CTU in Prague [2]. They use commercial LEGO Mindstorms kits so the hardware and its interfacing is almost hidden to students. Other courses (e.g. [3], [4]) use simulated robots. We have found this concept unsatisfying for our students since they work with simulations in majority of courses. Similarly to [5], our course is oriented to robot programming with an opportunity to interact with hardware directly, however without incorporating the vision system for navigation. Our course is supposed for beginners and less advanced students – we deal with basics of the robotics and instead of image processing we stay on the low level of robot control and interfacing.

For purposes of laboratory exercises we developed the robot Acrob [6] – see Fig. 2.

II. Robot Acrob

This robot is based on a commercial robot Boe-Bot (Parallax, Inc.) [7] with completely new electronics controller board. We had many good experiences with the original Boe-Bot robot, but its programming in PBASIC language was a

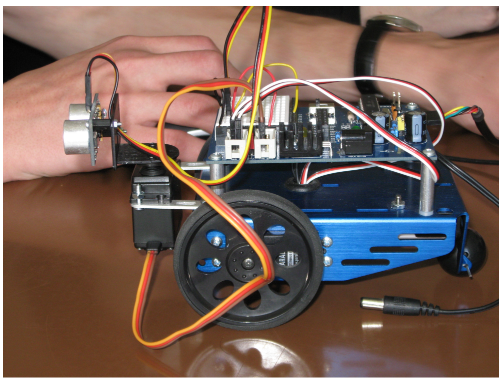


Fig. 2. Robot Acrob with added ultrasonic scanner.

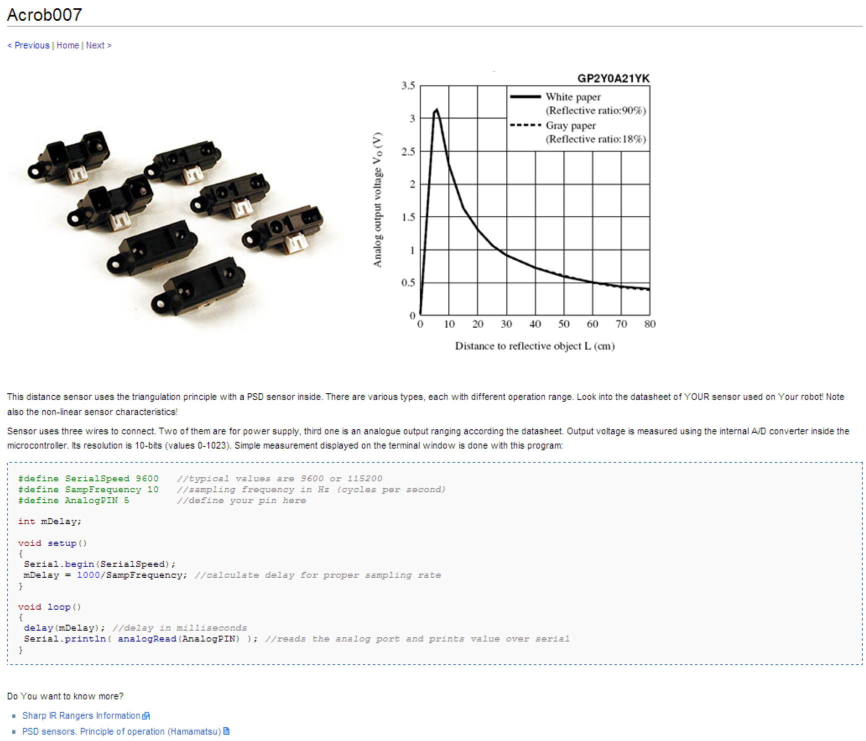


Fig. 3. An example of the course material available at [9].

pain for our students. They strived with the pitfalls of Basic – new language for them, instead of dealing with robotics.

New board is based on the Atmel Atmega328P RISC processor and its design was inspired by original Boe-Bot robot and Arduino [8]. The new board is as much as possible compatible with the original board – dimensions and connectors fits, so it can be replaced without problems. Moreover, we can still use a lot of original extension boards and peripherals. The board is compatible also with the Arduino Diecimila board (electrical and logical connections), so we can use Arduino libraries. Programming in C++ is very straightforward and using libraries effectively hides implementation details of the micro-controller.

On-board voltage stabilizator provides 5V for the micro-controller and its peripherals. Main processor is Atmel Atmega328P with a pre-burned bootloader. It provides 32 kB of program memory, 2 kB of data RAM space and 1 kB of EEPROM. The main area of the board is occupied with a solderless experimental breadboard where various additional components can be connected. On its left side most of 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.

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 standard FTDI Chips USB cable or SparkFun's FTDI Basic module for programming using the internal bootloader. We also developed a RS-232 level converter module to enable operation also with a standard serial interface.

After the program is loaded, the interface is free for any user serial communication operations. This allows to connect e.g. SparkFun's BlueMate communication module to communicate with a computer or between robots using the Bluetooth interface. On the board there is also a connector for an ISP programmer, so one can use any standard Atmel ISP programmer to burn the program into the processor. Together with AVRStudio one can even debug, step and watch programs written in assembler or avr-gcc languages. For programming during students laboratory exercises we used entirely Arduino environment.

The new robot is called Acrob (Arduino Controlled ROBot) and its detailed description is available in [6]. In the next section we will describe the usage of this platform for education in the *Robotics* course.

III. Exercises with Acrob

After a brief introduction of robotic platform and its development tools, we did basic experiments with a differential driven platform, its basic movements and sensory inputs. We performed analysis of robot properties measuring the

Fig. 4. Students of the secondary school testing the linefollowing algorithm during an international lecture in 2010 (project CENTROBOT).

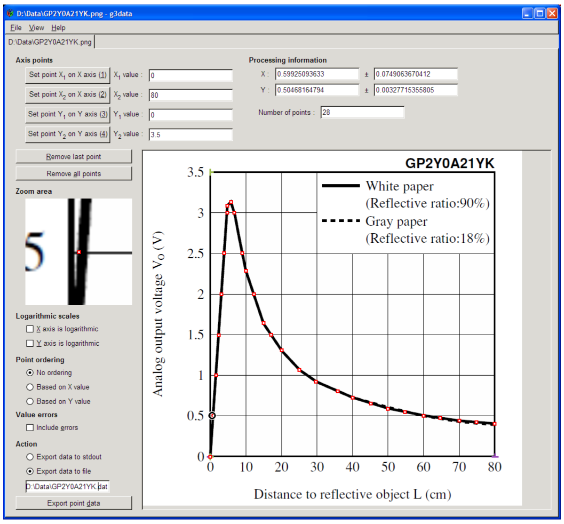


Fig. 5. g3data screenshot during manipulating sensor characteristics.

servomotor characteristics. Results were used for basic manoeuvring (forward, backward, rotations). Students were asked to calculate required speed and time that will drive the robot exactly 10 cm forward or turned it 90 degrees. This part of the course was self-paced with help of the on-line tutorial and example programs (see Fig. 3).

We also emphasised importance of understanding the principle of the sensor operation, knowledge of its properties to use the measured data properly. Also sensor modelling was discussed regarding to the possibility to obtain reliable results from it. Sharp analogue distance sensors (GP2Y family) are very appropriate for this, as their characteristics are non-linear and non-injective (e.g. output signal 2.5 V is measured for both 3 and 8 cm distances – see Fig. 5). Conversion of ADC values to distance was a difficult problem and valuable experience for students.

As the sensor producer does not provide the exact equation for conversion of the measured values, it is necessary to perform calibration measurements or to use the characteristics of the sensor from the datasheet [10]. For the latter we found useful the free program `g3data` [11] allowing to scan points from the image and recalculate its values based on the initial axes calibration (see Fig. 5).

Next step was to add infrared light detectors and to measure their properties. Then we used them as line sensors and created simple line-following robots. We tried to challenge students to create their own line-following algorithms in the form of the state diagram, but their programming skills are weak and require much more practising. Finally, they were lucky when their project simply worked, and they did not deal with a program structure.

An example excerpt from the student's four sensors line-following code follows:

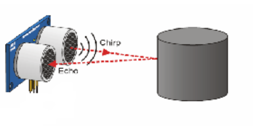
```
char GetState(void)
{
    char sensors[] = {1,3,0,2,-1};
    char i=0,state=0;

    while( sensors[i] >= 0 )
    {
        long val = analogRead( sensors[i] );
        if( val > THRESHOLD )
            state |= (1<<i);
        i++;  }
    return state;
}

void MakeMove( char state )
{
 switch( state )
  {
   /* 0001 */ case  1: Right(FAST); break;
   /* 0010 */ case  2: Right(SLOW); break;
   /* 0011 */ case  3: Right(MEDI); break;
   /* 0100 */ case  4: Left (SLOW); break;
   /* 0101 */ case  5: Right(SLOW); break;
   /* 0110 */ case  6: Fwd  (FAST); break;
   /* 0111 */ case  7: Right(SLOW); break;
   /* 1000 */ case  8: Left (FAST); break;
   /* 1010 */ case 10: Left (SLOW); break;
   /* 1011 */ case 11: Right(SLOW); break;
   /* 1100 */ case 12: Left (MEDI); break;
   /* 1101 */ case 13: Left (SLOW); break;
   /* 1110 */ case 14: Left (SLOW); break;
  /* fault */ default: Stop();      break;
  }
}          /* Author: Juraj Koys */
```

Listing 1: An example Linefollowing code.

A. Ultrasonic Distance Measurements

In the second part of labs we investigated deeply properties of ultrasonic sensors. The PING)))™sensors [12] by the Parallax, Inc., were used. An interesting idea of the sensor connection is used – measurements are controlled using a single I/O pin. Its direction is changed dynamically during the measurement.

Real metrological properties of the sensor are far beyond students' image of ideal sensors. The main misunderstanding which we noticed during classes was that ultrasonic acoustic wave is often improperly appearing in many diagrams as a narrow beam. In reality it is a diffusive acoustic cone (see Fig. 6).



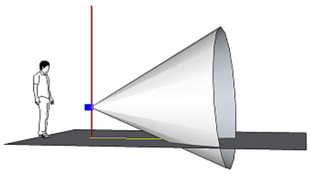


Fig. 6. Different views on the ultrasonic distance sensor principle (a) from a producer's datasheet, (b) in reality.

The students' tasks included

1) measuring a response time of the reflected ultrasonic wave
2) conversion of the measured response time into the distance in cm
3) exploring an effect of temperature changes
4) measuring and comparing sensor characteristics with and without the compensation
5) measuring the critical angle (where the wave is not reflected back to a transmitter)
6) measuring a minimal size of detected obstacles
7) measuring minimal and maximal detected distances
8) measuring sensor angular characteristics.

We found that even the 3rd-year students still had problems with proper data processing – they used unnecessary (and unreal) precision (6 and more decimal places) in tables, and their charts were non-descriptive (often totally useless) – see Fig. 7. This can be improved by quick and detailed feedback. Corrected reports should be returned as soon as possible to enable students to learn on their own mistakes.

distance (cm)	time (µs)	uncompensated distance (cm)	compensated distance (cm)
1	224	3,7128	3,872064
2	167	2,768025	2,886762
5	305	5,055375	5,27223
8	481	7,972575	8,314566
10	620	10,2765	10,71732
15	921	15,265575	15,920406
20	1159	19,210425	20,034474
25	1470	24,36525	25,41042

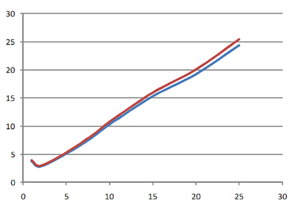


Fig. 7. Students still have problems with data processing.

An interesting question arose during lessons – whether there exist objects invisible to the sensor. One, classical, answer was: of course, there are objects too small or too furry (hairy) to be detected. But some groups implemented their findings about a critical angle and proposed an object similar to stealth technology (well-known in aerospace industry) – objects with many broken surfaces which reflect almost all waves away from the sensor. It was inspiring to see here the implementation of knowledge and experiences from many different areas.

When students were able to obtain a correct, callibrated and compensated value of the distance, we added a simple rotating servomechanism (see Fig. 2). Students' task was to measure distances in a range of angles in the front of the robot. To make this task more attractive, we slightly adapted the Peter Dainty's project Radar Screen [13]. It is a simple application receiving data in the form `XaaVbbb<CR><LF>`, where `aa` is actual angle and `bbb` is measured distance in cm. Data are sent over a serial line (using USB or even better, Bluetooth converter). Then application displays measured and received data in a form of radar screen (see Fig. 8). As there were source files available, we added a serial port opening drop-box. Compiled Processing [14] application was then available for students to see the measured data in a pleasant way.

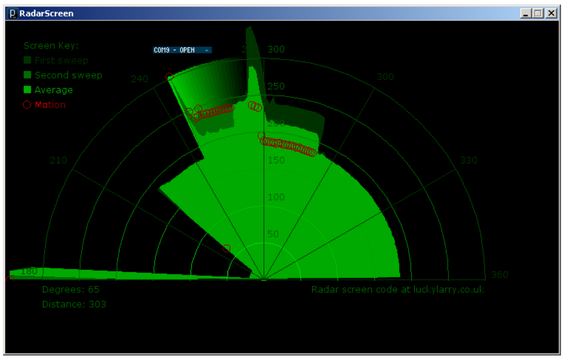


Fig. 8. Radar Screen window.

During measuring angular values from sensors, there was also a good opportunity to explain how to deal with circular data. We asked students to calculate mean value of following four angles measured in degrees (e.g. robot found an obstacle in following directions in four consecutive measurements):

1) 85, 95, 110, 90 (circular mean = 95, mean = 95)
2) 350, 360, 0, 10 (circular mean = 0, mean = 180)
3) 350, 360, 10, 360 (circular mean = 0, mean = 270)

More about *circular statistics* can be found in [15], [16].

The most difficult part of the work was an attempt to analyse measured data and to create a map of the environment. First, we created a probabilistic model of the sensor according [17], [18] and displayed a 3D graph of a single measurement.

Then, again according [17], we combined more measurements using a recursive Bayesian theorem to create a map of environment. This was done in Matlab or GNU/Octave using

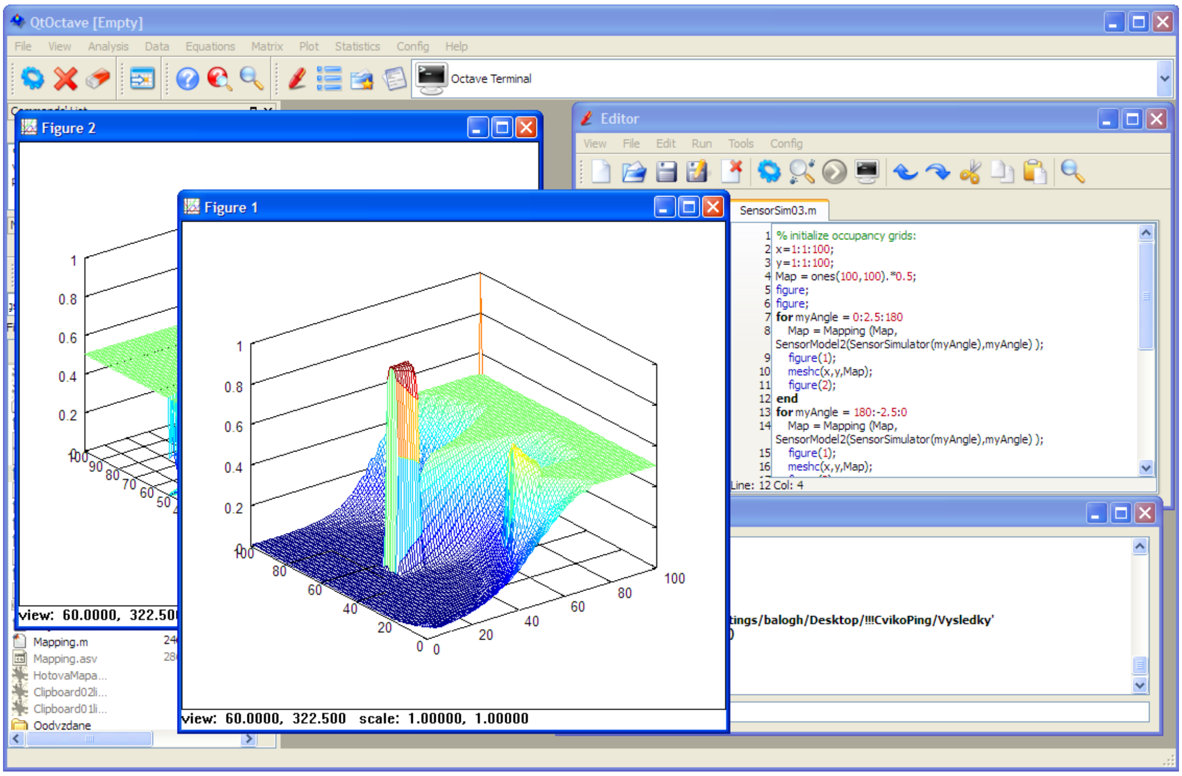


Fig. 9. QtOctave Screenshot.

a set of prepared scripts which were available for students. Some of them just changed simulated example data with their own off-line measurements, but some of them attached the robot on-line and collected data in real time. Similar approach was implemented in the Neptune Mobile Robot constructed at CMU by the team of Hans Moravec in the 80's [19].

```
function result = Mapping(Map,Sensor)
for Row = 1:100
 for Col = 1:100
  Map(Row,Col)=Sensor(Row,Col)*Map(Row,Col)/
  (    Sensor(Row,Col) *    Map(Row,Col)     +
   (1-Sensor(Row,Col))*(1-Map(Row,Col))   );

 end
end
result = Map;
```

Listing 2: An example of used mapping.

It is important to interpret results propely as can be seen in Fig. 9 – ultrasonic distance sensor is not the 3D scanner. It is just a probabilistic map of an obstacle's presence in the plane of the sensor. It means that pillars in the image are not pillars in reality. They are just places with increased probability of an obstacle presence. The green part of the image is not a wall or something similar; it is completely unknown area (the probability is 0.5 – the same as if one uses a coin to decide whether there is or is not an obstacle).

In fact, the pillar in the middle of the plane can be a real pillar as well as a small box of height few cm in the front of the sensor. Also the "tower" is not a real tower. From the measurement principle we cannot "see" behind the front reflective surface of objects, so we have no idea of their shapes. Thickness of the tower is proportional to the sensor uncertainty. It is clear that we can really "see" only the surface plane, not the thickness of the object. In fact, we can see something also from sides of the object and more or less reconstruct a shape of the object. But in our (Fig. 9) case, its shape is determined by uncertainties more than by real knowledge.

This part of the course was attractive for students as they could immediately see results of measurements and compare them with reality. They were able to watch a process of the map creation, but later they stated that not everything was clear, and asked for more time for this topic. The topic was also considered difficult to understand since it has more complicated mathematical background.

IV. Course Evaluation

At the end of the course, we surveyed students to get their evaluation of our conception. Generally, they had positive experience. Almost all of them considered the course interesting and well taught. A typical response was that nothing should be changed, as this was one of the best courses during their studies:

"*...interesting exercises, after all something practical and useful...*"

It proves that our students welcome everything what moves, can be controlled and is somehow connected to real life.

Unfortunately, some students had problems with study materials in English (datasheets, manuals, and some instructions): "*I did not like exercises as I have problems with education materials in English. Something was also translated into Slovak language, but not everything. Therefore, I did not understand many things. I even missed some of the exercises to develop.*"

Although this view was unique, it may represent a group of students which didn't complete the questionnaire. Unfortunately, knowledge of a foreign language is a must and we cannot make compromises here. It is necessary to improve their language skills during first years at the university.

We were also interested whether students worked independently in their groups. Plagiarism is a big issue in undergraduate classes. We were pleased that most of our students declared that all tasks were solved independently. Many of them highlighted benefits of working in pairs, or appreciated that we offered them partially prepared code, which they just needed to modify to fulfil requirements. Only one respondent wrote that he was completely unable to make assignments.

V. Conclusion

Despite students' optimism, our observations are a bit different: even students in the third year of bachelor studies have problems with basics of the C programming language. During the semester, we help them too often to solve "problems" with missing semicolons or brackets. And when finally programs worked, their programming culture was very low. Variables did not have semantic names, comments were not used, and students were not able to use appropriately even limited amount of structures in the C language .

Although the questionnaire was anonymous, we offered the possibility to sign it: 42% did it. We consider it as a sign that there exists an atmosphere of mutual trust.

Also the robot Acrob was well accepted. From teacher's point of view, we were satisfied with robot's reliability and ease of its operation. All planned experiments were easily implemented.

Positive acceptance of the course is obvious if one knows that first years at university are usually more theoretical (courses like Mathematics, Physics etc.).

Our experiences are overall positive and we would like to encourage others to introduce this type of education into their curricula.

Acknowledgment

Publication of this paper was supported through the Scientific Grant Agency of the Ministry of Education, science, research and sport of Slovak Republic with the grant VEGA 1/0690/09.

References

[1] Lalonde, Jean-franois and Bartley, Christopher P. and Nourbakhsh, Illah: *Mobile robot programming in education*. In IEEE International Conference on Robotics and Automation, 2006.

[2] Hlinovsky, Martin and Polcar, Tom: *Subject Robots at the CTU FEE in Prague – using LEGO robots to teach the fundamentals of feedback control*. In Proceedings of the 1st international conference on Robotics in Education, RiE2010. Bratislava : September 2010., pp. 119-122 FEI STU, Slovakia. Available on-line: http://rie2010.stuba.sk/proceedings.html

[3] Raz, Tzvi: *Graphics robot simulator for teaching introductory robotics*, IEEE Transactions on Education, vol.32, no.2, pp.153-159, May 1989 doi: 10.1109/13.28048

[4] Fernndez, J. and Casals, A.: *Open Laboratory for Robotics Education.* In Proceedings of ICRA. 2004, 1837-1842.

[5] Soto, Álvaro and Espinace, Pablo and Mitnik, Rubén: *A Mobile Robotics Course for Undergraduate Students in Computer Science*. In IEEE Latin American Robotics Symposium, Chile 2006.

[6] Balogh, Richard: *Educational Robotic Platform based on Arduino*, In Proceedings of the 1st international conference on Robotics in Education, RiE2010. Bratislava : September 2010., pp. 119-122. FEI STU, Slovakia. Available on-line: http://rie2010.stuba.sk/proceedings.html

[7] The Boe-Bot Robot education kit. Parallax Inc., Rocklin. Available on-line: http://www.parallax.com/go/BoeBot

[8] Arduino. Available on-line: http://www.arduino.cc/

[9] Acrob – an educational robotic platform compatible with the Arduino. Available on-line: http://www.robotics.sk/acrob

[10] GP2Y0A21YK – General Purpose Type Distance Measuring Sensors. Datasheet, Sharp Corporation. Available on-line: http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a_d_e.pdf

[11] Jonas Frantz: g3data.exe ver. 1.5.1. Available on-line: http://www.frantz.fi/software/g3data.php

[12] PING)))TM Ultrasonic Distance Sensor (#28015). Parallax, Inc., Rocklin, 2006. Available on-line: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf

[13] Dainty, Peter: *Arduino + Processing: Make a Radar Screen's Visualise Sensor Data from SRF-05.* Available online: http://luckylarry.co.uk/arduino-projects/

[14] Processing - an open source programming language. Available online: http://www.processing.org/

[15] Berens, Philipp: *CircStat: A MATLAB Toolbox for Circular Statistics.* Journal of Statistical Software, September 2009, Volume 31, Issue 10. Available on-line: http://www.jstatsoft.org/v31/i10/paper

[16] Kulčár, Ladislav: *Problematika určenia aritmetického priemeru pri kruhovom rozdelení.* [in Slovak] In Proceedings of the IMEM International congres. September 2009, pp. 104-112. PF KU Ružomberok, Slovakia.

[17] Murphy, Robin R.: *Introduction to AI Robotics*, MIT Press, 2000.

[18] Elfes, Alberto: *Using Occupancy Grids for Mobile Robot Perception and Navigation.* IEEE Computer, **1989**, 22, 46-57.

[19] Moravec, Hans: Neptune Mobile Robot Project Archive. http://www.ri.cmu.edu/research_project_detail.html?project_id=48&menu_id=261

Robot competitions trick students into learning

Francis Wyffels, Karel Bruneel, Pieter-Jan Kindermans, Michiel D'Haene,
Pierre Woestyn, Peter Bertels and Benjamin Schrauwen

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

***Abstract*—It has been shown in the past that robots help to bring theoretical concepts into practice, while at the same time increasing the motivation of the students. Despite these benefits, robots are hardly ever integrated in education programs and at the same time students feel that they have the competences nor the infrastructure to build a robot on their own. Therefore the workgroup electronics (WELEK) of Ghent University gives students the opportunity to build a robot by organizing workshops and competitions. Up until now, four competitions were organized in which over 200 students voluntarily participated. This paper describes our approach in the hope that it will inspire other educators to do the same thing. We also measured the effectiveness of our competitions by sending each of the participants a questionnaire. The results confirm that students acquire relevant technical competences by building a robot, learn to work as a team and are challenged to use their creativity.**

I. Introduction

There is no doubt that we live in the dawn of the robot era. Just like the introduction of personal computers changed our lives, robots have started to revolutionize our way of living. Since long, people have been fascinated by robots, but only now we have the technology and the knowledge to build cheap robots that can take over some of our daily chores. This fascination for robots is especially prominent in engineering students, which feel the urge of being part of the robot revolution. However, many of them are discouraged from building their own robot because they believe that building even a simple robot requires lots of skills and infrastructure. This is of course not entirely untrue, but in this paper we show that by opening up some of your universities infrastructures and with the right goal in mind, a screeching robot battle, they will gladly learn all the skills they need and have fun doing so.

It has been shown before that using the students fascination for robotics early in the curriculum can be very beneficial: students from varying disciplines learn to value and utilize each others' knowledge, by means of a basic robotics course [1]; Integrating a robot project in a undergraduate education program increases students interest in research [2]; Robots motivate students to solve problems which they otherwise find tedious [3]; The opportunity to participate in a robot competition boosts the interest of both high school students and undergraduates in robotics and engineering in general [4]. Robot competitions encouraged students to apply their knowledge to a real-world problem and motivates them to learn new concepts on their own [5]. This paper confirms these finding and further strengthens the motivational claims.

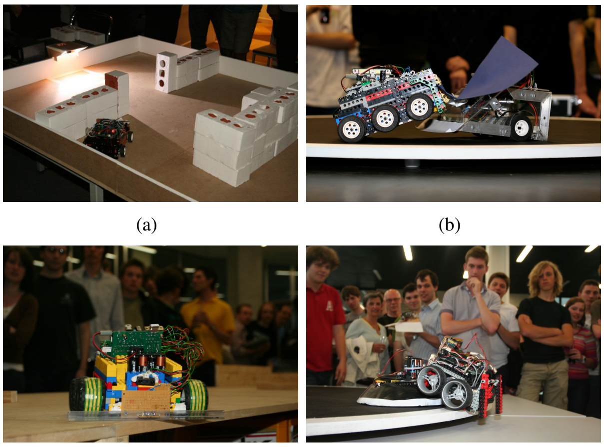
It all started at the engineering faculty of Ghent University, where PhD students founded the workgroup electronics (WELEK) with the aim to organize practical workshops on electronics for students. These workshops give students the opportunity to gain more hands-on experience with electronics. Students can build one of many electronic devices such as an FM-transmitter, a VU-meter or an electronic bat detector, or they can use the infrastructure to work on their own electronics project. Besides the standard workshops, WELEK yearly organizes a series of workshops on robotics which concludes with a robot competition for autonomous robots, since 2008. The goal of WELEK is actually three-fold: (1) lower the threshold for students to get involved into electronics and robotics by providing guidance and infrastructure, (2) teach several concepts in electronics and robotics in a more practical way and as such, (3) motivate students towards electronics and robotics.

More than 200 students have participated in one of the four robot competitions that WELEK has organized so far. This paper describes the results of a questionnaire that was send to all of these students. The questionnaire was answered by more than one third of them (76 students). In the next two sections we describe the details of the four robot competitions we organized and the hardware that was used by the students. Later, in Section IV we describe how our workshops are organized and relate this to the outcome of our questionnaire. We also questioned the students about the knowledge they have gained and whether they enjoyed these robot competitions or not; these results are discussed in Sections V and VI. Finally, some conclusions are drawn.

II. Robot competitions

The electronics workgroup[1] (WELEK) is a student organization that was established at the faculty of engineering at Ghent University in the early nineties. WELEK organizes hands-on workshops which give the students the opportunity to use the universities infrastructure to build one of the available projects or to work on their own application. The universities infrastructure was kindly opened by the head of the ELIS department at Ghent University. During the sessions the students can get assistance from more experienced students and PhD students if needed. The sessions are held every two

[1] See http://www.ieeesb.ugent.be/welek for more information



(a) (b) (c) (d)

Fig. 1. Overview of four year robot competitions: (a) a light eating robot competition in 2008, (b) a sumo robot competition in 2009, (c) an autonomous robot navigation contest in 2010, and (c) a sumo robot competition in 2011.

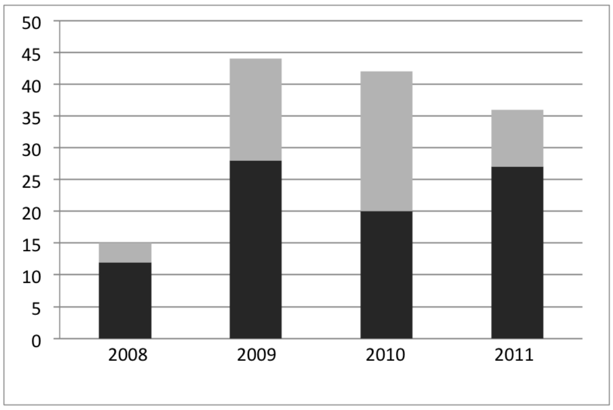


Fig. 2. In light gray, the number of teams that subscribed at the beginning of the semester to our competition. In black, the number of teams that actually participated in the competition (eg had a working robot at the end of the semester).

weeks of the first semester. Students attend the workshops on a voluntary basis. They cannot earn any credits for participating.

In 2008, the idea rose to organize a robot competition for "intelligent" robots, accompanied with some introductory workshops on robotics. Since then, four competitions have been organized: (1) a light eating contest, (2) a sumo robot competition, (3) an autonomous navigation contest and (4) again a sumo robot competition. A visual overview of these competitions is shown in Figure 1. In what follows we give a short description of the setup and guidelines of the four competitions:

- In 2008, *light eating contest*, the goal was to build a robot that can drive autonomously towards a source of light, while avoiding obstacles on their way. The obstacle course gradually became more difficult towards the end of the competition with obstacles that for example tried to trap the robot or block the light. The robots were limited in size and weight: the maximum width, length and height of the robot was 20 cm and the total weight was limited to 1500 grammes. In order to track the light, and to detect obstacles, light sensitive sensors and short range reflective distance sensors could be used. For the transmission, only electric (DC) motors were allowed. The "brain" of the robots in all competitions is a microcontroller board, which will be explained further.
- In the *sumo robot competition* of 2009, the goal was to build a robot that could push other robots out of a circular shaped arena. Robots were limited to a width and length of 25 cm and could be infinitely high. The weight had to be lower than 1250 grammes. Robots could use electrical (DC) motors for driving the wheels, and optionally additional motors or RC-servos for driving levers or expanding pieces to distract other robots. For the sensors, long range distance sensors could be used to find other robots, and reflective sensors to detect the edge of the arena.
- In 2010, an *autonomous robot navigation* contest was organized. The goal was to autonomously navigate through an obstacle course and reach the other side of the course. In contrast to the competition of 2008, there was no navigation light, instead robots could use a compass sensor to keep track of their orientation. Of course, also long and short range distance sensors could be used to detect obstacles. Robot were limited to a width and length of 25 cm and a height of 50 cm. The weight was limited to 2000 grammes. Only electrical (DC) motors and RC-servo motors were allowed.
- In 2011 we organized a second edition of the popular *sumo robot competition* with the only difference being an increase of the weight limit to 1500 grammes.

At the beginning of the second semester the students register to the competition individually, or in teams with two or tree members. During the semester they can attend evening sessions to work on their robots. More than ten sessions are organized during the course of the semester. Two of the sessions are organized as lectures where the basics of robot building are explained, but in most of the sessions the students just use the lab infrastructure of the university and our guidance to build their robots. Apart from the sessions, students often work at home on their robot during their free-time. About two months after the first session, the competition itself is organized. The students bring their friends and family to support them in the battle for some very nice prices.

Over the years, more than 200 students participated in one or more of the robot competitions. The level of graduation of the participants is distributed uniformly, ranging from freshman to senior students. Most students are studying or are intending to study computer science or electronics, with a participation of 32% and 23%, respectively. Others have various backgrounds ranging from mechanical engineering to bio-engineering and

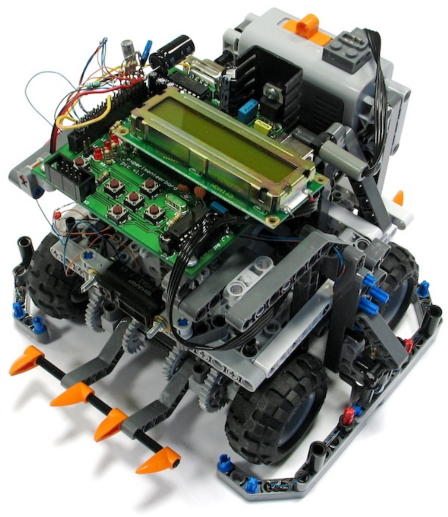


Fig. 3. A typical example of a robot build by the students. One can identify the microcontroller board, a set of motors, battery back and sensors.

even geography. The grey bars in Figure 2 we show the number of teams[2] that subscribed to the competitions. In 2008, the number of participants was rather limited, but the competition quickly grew to about forty teams. The black bars in Figure 2 show the number of teams that actually competed on the final day. While most of the teams manage to build a working robot by the day of the competition, some teams quit because of limited time due to other (mandatory) projects in their education program.

In Section IV we describe our approach to (1) make our workshops and robot competition doable for everybody, and (2) make sure that as many teams as possible finish their robot.

III. A CHEAP ROBOT PLATFORM

A part of getting students motivated to participate in a robot competition is providing them with cheap and easy-to-understand components. A robot has three major parts: (1) the mechanical part, consisting of a robot chassis and motors, (2) the brain of the robot, i.e. a microcontroller board and (3) the sensors. From our experience, we know that students have no difficulty in finding a robot chassis and motors. Most of them recycle Lego™parts from their childhood, others are more creative and use metal plates and fetch some motors elsewhere. The electronics, both the processing unit and the sensors, are regarded more difficult since most of the students have little or no experience with microcontrollers and sensors. Therefore, WELEK provides a microcontroller board and proposes a number of sensors that can be used, depending on the type of competition. Typical sensors that have been proposed include phototransistors, short and long range distance sensors and in case of the autonomous robot navigation contest a compass sensor.

The microcontroller board WELEK proposes is the Dwengo board[3] [6], a good priced platform with a PIC18F4550 and a wide range of onboard peripheral which can be used to easily build a robot without the need of additional electronics. The board comes with a display, some generally applicable buttons and LEDs, a quad-bridge motor driver, a USB and serial port and an expansion connector that enables easy integration with different sensors. In order to make the programming of the robot's intelligence easier, we provide a framework in C which makes all the needed functionality easily accessible so that the participants only have to focus on how they implement the robot's behavior. A battery pack of six or eight AA batteries can be attached easily to power the robot.

For students it is very important that building a robot does not take up too much of their limited budget. In our experience, students can build a robot from scratch for less than EUR 100. This includes the microcontroller platform, two motors and a set of sensors. Thanks to sponsoring, we can even significantly lower the actual price the students have to pay for the robot. Additionally, they have the opportunity to spread the cost by working in teams and by recycling components used in previous competitions. Making sure participating is cheap motivates students because they still have money left to buy some beers.

IV. APPROACH

Everyone who has organized a competition, especially a competition with a technical aspect such as a robot competition, has experienced the phenomenon that people subscribe to the competition but don't participate in the final event. This effect can be observed in Figure 2 which presents an overview of the number of teams that had the intention to participate (light gray) and the actual number of teams that had a working robot on the day of the competition (black). The main reason why students drop out is lack of time, they have other projects which are mandatory in their curriculum. But apart from that, two other aspects are important: (1) the difficulty of the competition, and (2) the availability of guidance.

Figure 2 shows that in the 2010 competition less than usual teams finished their robot. While the exact reason can not be derived from our questionnaire (students claim to drop out because of lack of time), we believe that the complexity of this competition is also a significant reason. In contrast to the other competitions, three types of sensors were needed to build a robot that could successfully complete the obstacle course. Additionally, students needed to come up with more advanced control strategies in order to deal with all possible obstacle configurations. In the future we plan to keep the concept of the competitions simple so that each team can at least build a working robot and compete with their peers in the final event.

[2]On average, each team contains 2.3 individuals

[3]Dwengo vzw is a non-profit organization that supports people who like to experiment with micro-controllers http://www.dwengo.org

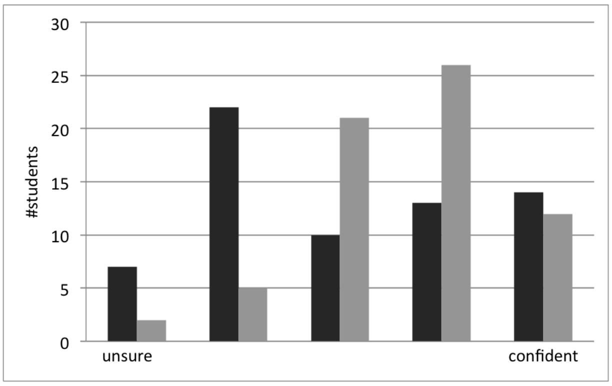


Fig. 4. Illustration of how students estimate their prior knowledge and skills to build their own robot. In black, their believe in prior knowledge before participating the robot workshops and contest. In light gray, the students' estimation of the necessity of having prior knowledge and skills after they participated in the robot competition.

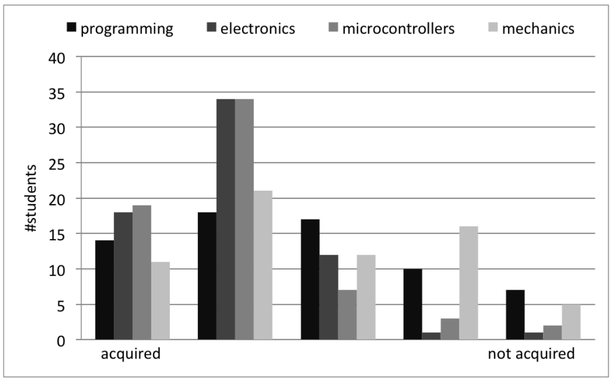


Fig. 5. Competences acquired by the students: programming skills (black), electronics (dark gray), microcontrollers (light gray) and mechanics (lightest gray).

At the same time it should be the case that creativity and hard work leads to better working robots. Sumo competitions are a good example of a simple concept with lots of room for creativity.

Despite our efforts, still a lot of students believe they don't have the knowledge to build a robot and to participate in a robot competition. The black bars in Figure 4 show how students estimate their prior knowledge and skills *before* participation. In the same Figure the grey bars show how students estimate the importance of prior knowledge and skills *after* participating. We observe that students are unsure about whether they are able to build a robot or not, but once they have done it, they see that it is not that difficult at all. We believe that an important reasons of this shift towards more confidence is the intensive guidance we provide during the robot building.

Typically, we start with a kick-off session at the beginning of the second semester. In this session we give an overview of the goal including the rules and a basic explanation of what a robot is and how students can start building one. After this overview, students get the opportunity to subscribe, and to buy and solder the necessary components. Next, two times two hands-on soldering sessions are organized during which the students can solder the microcontroller platform. After that, a theoretical session about sensors and how to program robots is giving. We explain which sensor types can be used, how the output of these sensors can be interpreted and what functionality is available in the programming framework. We observed that even freshman students with no electric experience and almost no programming experience are able to understand the concepts in this theoretical session and apply them to their robot. This theoretical knowledge can then be applied to the their robots in the following four (again two times two) guided sessions. Typically, students build their robot, solder and connect the sensors, and perform some tests. Right before the competition itself there is an extra session which can be used to test their robot and to fix the last few problems. During all sessions, the WELEK team is there to help the students and answer their questions. By attending the sessions, students see the work of the other teams which gives them the opportunity to compare robots and learn from the other teams. In our questionnaire, a huge part of the participants noted that the sessions motivated them in building their own robot.

V. Learn by building robots

That robots can motivate students to learn, even in their free time has been shown before [7]. From our questionnaire we wanted to learn how students estimate the knowledge they have acquired by participating in our robot workshops and robot competition. The results are presented in Figure 5. We explicitly asked how much they feel their knowledge about programming (black bars), electronics (dark gray bars), microcontrollers (light gray bars) and mechanics (lighter gray bars) has improved. Not surprisingly, students feel that they have acquired a lot of knowledge about electronics and microcontrollers. Their programming skills and their knowledge about mechanics have also increased, but to a lesser extent.

But not only knowledge is important. We also asked the participants how much their creativity was stimulated during this competition and wether they learned to work in team or not. In Figure 6 we present the results: the black bars show wether students learned to work in teams, while the gray bars illustrate to which extend the students think their creativity was stimulated. It is clear that the students creativity was highly stimulated, but learned to work in teams to a lesser extend. The latter could be due to the fact that most teams consist of friends which knew each other beforehand and in this way already knew how to work together.

Overall, we can conclude that by participating in the robot workshops and the competition, students feel they really learned useful competences. These competences are in many cases complementary to those acquired in the standard curricu-

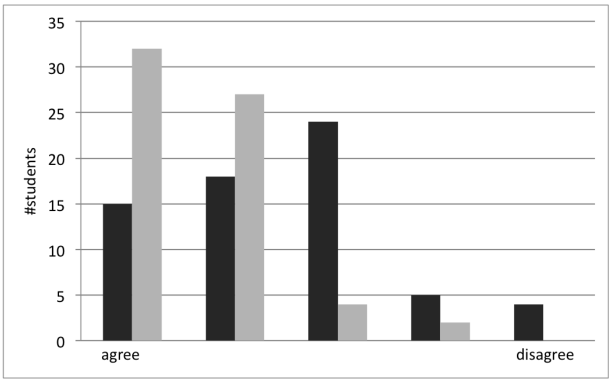


Fig. 6. Acquired soft skills: in black, how much the students feel that they have learned to work in teams, in light gray, how much students feel that their creativity was stimulated.

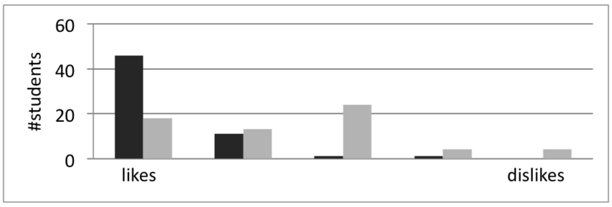


Fig. 7. Estimation of the amount of fun they had: in black, students response on the question how much they liked building a robot, in gray their intention to participate in the edition next year.

lum where the emphasis is put on theory while we emphasize the practical side of electronics.

VI. Robot competitions are fun

We also wanted to learn how much the students liked building robots and participating in robot competitions. We get a first indication from their answer to the question how they learned about our robot workshops and competitions. One third of the students claimed to be notified by other students or friends. Apparently students like the competition so much that they pass it on to their friends. This can also be derived from Figure 2 in which it can be observed that we started small, with a few students that already participated in our standard electronic workshops, while the next year the word was spread and the robot competition became a big event with a lot of interested students.

Additionally we explicitly asked them wether they enjoyed building robots or not. The outcome of this question is presented by the black bars in Figure 7 and is very positive. This is also confirmed by the fact that the majority want to participate in next year's competition (gray bars), however, some still have doubts. A possible explanation for these doubts can be that they don't yet know the work load next years educational program.

VII. Conclusions

In this work we presented our approach for organizing robot competitions for students. Even though these competitions are not part of the curriculum, and thus no credits can be earned for it, every year a huge number of enthusiastic students build a robot and participate in the competition. We believe that there are two reasons for this success: (1) students are fascinated by robots and they feel the urge to build one, and (2) the robot competition has been made as accessible as possible by organizing guided hands-on workshops and opening the universities infrastructure.

With this paper we want to stimulate other educational institutions to give their students the opportunity to participate in an easily accessible robot competition. This is useful because, as we learned from our questionnaire, students acquire relevant technical competences by building a robot. Moreover they learn to work as a team and are challenged to use their creativity. So, by giving students the opportunity to participate in robot competitions, they can acquire skills that are useful for their future career.

VIII. Acknowledgments

The authors acknowledge the contributions of the other members of WELEK, specially: Tim Waegeman, Bart Coppens, Sean Rul, Guy Torfs, Tom Davidson, Dries Van Puymbroeck, Gertjan Van Droogenbroeck and Sander Dieleman. The authors would also like to thank the huge number of students that enthusiastically build their robot and participate in our competitions.

References

[1] S. Shamlian, K. Killfoile, R. Kellogg, and F. Duvallet, "Fun with robots: A student-taught undergraduate robotics course," in *Proceedings of the International Conference on Robotics and Automation*, 2006.
[2] B. Maxwell and L. Meeden, "Integrating robotics research with undergraduate education," *IEEE Intelligent Systems Magazine*, pp. 22–27, 2000.
[3] F. Wyffels, M. Hermans, and B. Schrauwen, "Building robots as a tool to motivate students into an engineering education," *AT&P JOURNAL PLUS*, vol. 2, pp. 113–116, 2010.
[4] N. Sunderhauf, T. Krause, and P. Protzel, "Bringing robotics closer to students - a threefold approach," in *Proceedings of the IEEE Conference on Robotics and Automation*, 2006.
[5] D. Pack, R. Avanzato, D. Ahlgren, and I. Verner, "Fire-fighting mobile robotics and interdisciplinary design-comparative perspectives," *IEEE Transactions on education*, vol. 47, pp. 369–376, 2004.
[6] P. Bertels, F. Wyffels, and K. Bruneel, "Building an intelligent robot in the class room with dwengo," in *Proceedings of the National Informatics Education Conference*, 2011.
[7] J. S. Kay, "Robots in the classroom ... and the dorm room," *Journal of Computing Sciences in Colleges*, vol. 25, pp. 128–133, 2010.

Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions

Luc Guyot
PSE C - EPFL
Cyberbotics Ltd.
Lausanne 1015, Switzerland
Email: Luc.Guyot@Cyberbotics.com

Nicolas Heiniger
Cyberbotics Ltd.
Lausanne 1015, Switzerland
Email: nicolas@photochrome.ch

Olivier Michel
PSE C - EPFL
Cyberbotics Ltd.
Lausanne, Switzerland
Email: Olivier.Michel@Cyberbotics.com

Fabien Rohrer
PSE C - EPFL
Cyberbotics Ltd.
Lausanne, Switzerland
Email: Fabien.Rohrer@Cyberbotics.com

***Abstract*—We introduce a robotics curriculum intended for all levels of learning and discuss results of related in-class experiments and competitions. The curriculum is an open document with a collaborative format, hence freely accessible and extendable. Based on the e-puck mobile robot and the Webots simulator, it addresses a dozen of topics ranging from finite state automata to particle swarm optimization. While beginners familiarize with a user-friendly graphical programming interface, most advanced readers benefit of apt exercises to tackle robotics contests.**

***Index Terms*—education, robotics, e-puck, competitions**

I. Introduction

Robotics is widely considered as an excellent tool for teaching science and engineering [14], [17] and such a belief is reinforced by the ever growing number of successful educational experiments, up-to-date curricula and new didactic approaches [10], [25], [28]. Nevertheless Mataric bemoans the lack of age-appropriate teaching materials in her workshop report [17] and urges the robotics community to broaden both scope and audience of educational supports. Our curriculum is an attempt to fulfill this twofold expectation.

Our curriculum aims at teaching hands-on robotics through the cost-friendly and widely used e-puck robot, benefiting further of its rich interplays with the Webots simulation software. Grounded on a former document involving the Hemisson robot [15], the curriculum presented here was originally written by the last three authors [16], [24]. Distributed for the first time in 2008, it has been used ever since as a support for master courses at the École Polytechnique Fédérale de Lausanne (Switzerland). The document stems from privileged interactions between two educational and research supports, namely e-puck and Webots. Aimed at the broadest possible audience, it defines five levels of learning where beginners can acquire the basics of robotics without any prior programming knowledge (Section II-A) while skilled users are prepared to compete in Rat's life [22] and RobotStadium [21] (Section II-E).

The curriculum is released as a wikibook [4] under the terms of the GNU Free Documentation License and the Creative Commons Attribution-ShareAlike 3.0 Unported License, so it benefits from robotics community contributions. (A PDF version can also be downloaded from Cyberbotics website [3].)

The curriculum begins with a general introduction on robotics and then describes the e-puck robot and Webots. The remaining part divides into five sections, each dedicated to a specific level of learning. In keeping with this framework, our article gives a description of every individual section (except the general introduction) so that interested teachers can quickly grasp the bulk of it. The last section is devoted to the analysis of in-class experiments.

A. The e-puck robot and Webots simulation software

a) e-puck: The e-puck mini mobile robot was originally developed at the EPFL for teaching purposes by the designers of the Khepera robot. The e-puck hardware and software is fully open source, providing low level access to every electronic device and offering unlimited extension possibilities. The robot is already equipped with a large number of sensors and actuators (Figure 1) and possesses a Microchip dsPIC 30F6014A with a frequency of 60MHz. It is well supported by the Webots simulation software which provides simulation models, remote control and cross-compilation facilities. The Webots-oriented programming toolchain comprises a multi-platform gcc cross-compiler and a firmware dedicated to the software-hardware interplay. When cross-compiling, the robot is to be programmed in C while remote control session allows the use of all languages supported by Webots.

The official e-puck web site [9] gathers a large quantity of information about the robot, extension modules, software libraries, users mailing lists, etc.

b) Webots: The Webots robot simulation software is a commercial software for fast prototyping and simulation of mobile robots which has developed tight links with the e-puck robot. It was originally developed at the Swiss Federal

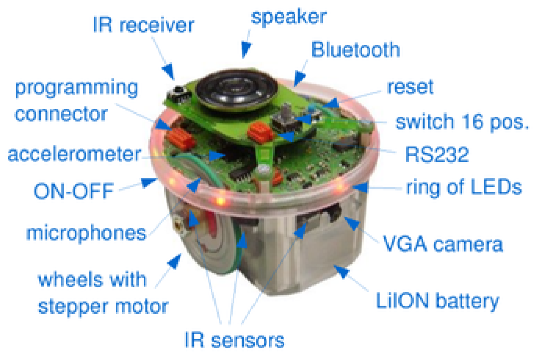


Fig. 1. e-puck devices

Institute of Technology in Lausanne (EPFL) since 1996 and has been continuously developed, documented and supported since 1998 by Cyberbotics Ltd. Over 850 universities and industrial research centers worldwide are using this software for research and educational purposes. Although Webots is a commercial software, a demo version is freely available from Cyberbotics web site [5]. This demo version includes the complete Rat's Life simulation, a cognitive benchmark described at the end of the curriculum (see Section II-E)

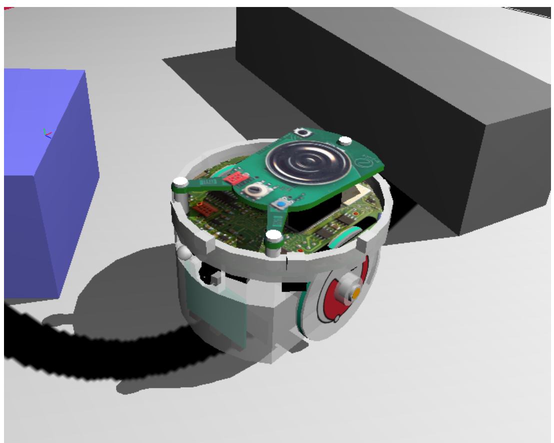


Fig. 2. Simulated e-puck in Webots

c) e-puck interface and simulation model: On hardware side, Webots allows remote control sessions of the real robot through Bluetooth connection and supports cross-compilation. On simulation side, it provides an accurate e-puck model including the differential wheel motors and their encoders, the infra-red sensors for proximity and light measurements, the accelerometer, the camera and the 8 surrounding LEDs (see Figure 2). Webots has moreover a graphical dedicated interface which displays sensor and actuator values (Figure 3) and enables beginners to step into robotics without a single line of code (see BotStudio in Section II-A).

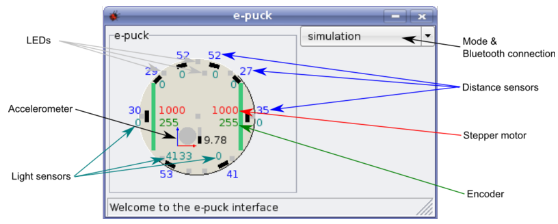


Fig. 3. e-puck window in Webots

B. *Five levels of learning*

The curriculum is designed to be used by pupils, under- and postgraduate students, autodidact hobbyists and researchers as well. The target audiences read as follows.

1. Beginner: The user is totally new to robotics and computer science. A typical beginner would be a high school pupil using a programming environment and a robot simulator for the very first time.

2. Novice: The user has a basic knowledge of programming and can decipher a simple C code but does not necessarily have a background in robotics. A typical novice would be an undergraduate student attending an introduction to robotics.

3. Intermediate: The user has an experience in both programming and robotics and can write his own C code. A typical intermediate user would be a student in science.

4. Advanced: The user has an important experience in both programming and robotics and can program complex controllers. A typical advanced user would be a post-graduate student or a researcher looking for working examples and references.

5. Expert: The user is a professional in robotics. A typical expert would be a participant of a robotics contest.

II. Curriculum's content

Curriculum's exercises are ordered with an increasing level of difficulty; sections break down accordingly.

A. *Beginner*

The beginner section consists of eight exercises culminating with a line following algorithm and a challenge named *Rally*: create a finite state automaton that makes the e-puck robot follow a winding road without leaving it. The user manipulates either a simulated e-puck robot on Webots interface, or the real robot by uploading controllers via a Bluetooth connection. S/he will discover most of the e-puck devices along her/his progression.

BotStudio: BotStudio is a graphical programming interface specially designed for educational purposes. It makes it possible to create a finite state automaton by simple drags and drops (see Figure 4). The automaton is then turned into a controller file that can be run on both virtual and real e-pucks. In the latter case, BotStudio uploads controllers, for the ease of the user.

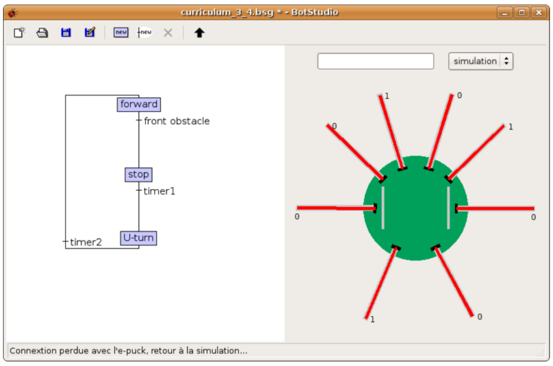


Fig. 4. An automaton in BotStudio to make a U-turn with e-puck

Key features:

- Graphical programming
- Sensors and actuators: infra red sensors, stepper motor, linear camera, LEDs
- Finite state automata
- Collision avoidance algorithm
- Line following algorithm

B. Novice

The novice section consists of eight exercises exploring BotStudio and the e-puck devices in greater depth. C programming is progressively introduced to code old and new behaviours of the e-puck robot. The set up of a remote control session and the cross-compilation process are thorougly described.

Key features:

- Basic C programming
- Elaborated finite state automata
- Simulation of robots interacting with each others
- Wall following algorithm
- Calibration of infra red sensors, e-puck's camera and accelerometer

C. Intermediate

The intermediate section consists of an introduction to behavior-based robotics made of five examples of behavioral modules. It culminates with a program making the e-puck robot avoid obstacles while following a line.

Key features:

- Behavior-based artificial intelligence
- C programming of behavioral modules: Line following, wall following, obstacle avoidance and scanning

D. Advanced

The advanced section consists of seven exercises culminating with a probabilistic approach to the simultaneous localization and mapping problem (SLAM). These exercises aim at giving an insight of today's robotics while laying the foundations for expert robotics competitions. Indeed, exercises on odometry, pattern recognition and SLAM teach how to use the e-puck visual system efficiently and how to navigate in an unknown environment. These are two of the main challenges of curriculum's expert benchmarks.

Key features:

- Odometry and calibration
- Path planning: potential field and NF1 algorithm [27]
- Supervised learning: artificial neural network for pattern recognition [11]
- Unsupervised learning: particle swarm optimization [12], [20], [26]
- Genetic algorithm [13], [18], [19] using Braitenberg vehicles [2]
- SLAM [6]–[8]

E. Expert

At this stage, the user is fit for robotics contests. This last section reads as an invitation for challenges and offers several clues as how to tackle the two following benchmarks.

1) Rat's life:

a) The competition: Rat's Life is a cognitive robotics benchmark particularly suited for research in SLAM, autonomy and vision. Easily reproducible in a lab with limited resources, it relies on two e-puck robots, some LEGO bricks and the free version of the Webots software. It is a survival game where two robots compete against each other for resources in an unknown maze (see Figure 5). Like the rats in cognitive animal experimentation, the e-puck robots look for feeders which allow them to live longer than their opponent. Once a feeder is reached by a robot, the robot draws energy from it and the feeder becomes unavailable for a while. Hence, the robot has to further explore the maze, searching for other feeders while remembering the way back to the first ones. Rat's life online simulation contest runs its third edition in 2011.

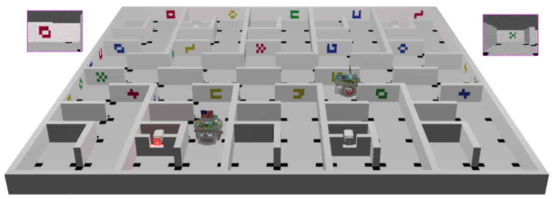


Fig. 5. Two e-puck in Rat's life maze and their camera displays

b) Evolution:

Year	Number of competitors
2008	41
2010	16
2011	15

Simulation movies of the competitions are stored in a data base in order to analyze the evolution of the competition over the time. The movie database contains more than 2500 movies (totaling more than 50 GB of data) and is freely available online at http://www.cyberbotics.com/ratslife/movies/.

Intensive observations revealed a co-evolution dynamic similar to the ones encountered in genetic algorithms. Once a performance breakthrough is made by one competitor, it is immediately analyzed by the other competitors seeing the simulation movies. They take inspiration from it to improve their own robot controller and submit their new improved version to the contest. Rapidly, a large number of competitors replicate this winning behavior on their robot and most of the robots in the contest adopt this new efficient behavior. We list the most significant breakthroughs made by challengers in chronological order.

- *Random Walkers:* The random walkers came actually from the very first version of the sample source code included with the contest software development kit, made available to all competitors. This simple control algorithm similar to Braitenberg vehicles [2] lets the robots move randomly while avoiding the obstacles.
- *Vision-Enabled Random Walkers:* They are an improved version of the original random walker making an extensive use of vision to recognize the feeders and adjust the trajectory of the robot to reach the feeder instead of simply moving randomly. This results in slightly more efficiency as robots will not pass in front of a feeder without getting energy from it.
- *Right Hand Explorers:* A Braitenberg vehicle behavior is not very efficient at exploring extensively a maze and hence at finding the feeders. Maze exploration algorithms exist and are much more efficient. The right hand algorithm is one of the simplest and best known maze exploration algorithms. It consists in simply following the first wall found on the right hand side of the robot. Using this algorithm combined with some vision to reach efficiently the feeders, a significant performance breakthrough was reached.
- *Energy-aware robots:* Getting the energy from the feeder as soon as you find the feeder is nice, but there is an even better strategy: Once a robot finds a feeder, it can simply stop and sit in front of the feeder, thus preventing the other robot from reaching this feeder. In the meanwhile the robot sitting in front of the feeder should watch its energy level and decide to move to the feeder once its energy level reached a very low value, just enough to make that move to the feeder and refuel. During this waiting time, the other robot may be struggling to find a feeder and possibly loose the game if it runs out of energy.
- *SLAMers:* Compared to other techniques mentioned above, it involves a much more complicated algorithm and requires an efficient image processing. SLAMer robots actually seems to use the right hand algorithm on a first stage to explore extensively the maze, but they dynamically build a map of this maze while exploring it and eventually don't use the right hand algorithm at all. Their internal representation of the environment contains the walls, the feeders and likely the landmarks. This map is then used by the robot to get back to previously found feeders. It turned out to be very efficient and clearly outperformed the simpler reactive controllers.
- *Super-SLAMers:* A major improvement of SLAM-based robot controller was probably the estimation of the status of the feeders, combined with an estimation of the time needed to travel through the maze to reach the feeder. A Super-SLAMer seems to be able to anticipate that a mapped feeder will become available again: when the feeder is still red, it starts to navigate towards this feeder and about one second before it reaches the feeder, the feeder becomes green again.

2) *RobotStadium:*

a) The competition: Robotstadium is an online simulation contest based on the new RoboCup Nao Standard Platform League (SPL) [23] and relying on a free version of Webots. Running every year since 2008, this simulation features two teams with four Nao robots each team, a ball and a soccer field corresponding the specifications of the real setup used for the new RoboCup Nao SPL (see Figures 6 and 7). Competitors simply register on the web site and download a free software package to start programming their team of soccer-playing Nao robots. Once they have programmed their team of robots, competitors can upload their program on the web site and see how their team behaves in the competition. Matches are run every day and the ranking is updated accordingly in the "hall of fame". New simulation movies are made available on a daily basis so that anyone can watch them and enjoy the competition on the web. The contest is running online for a given period of time after which the best ranked competitors will be selected for an on-site final during the next RoboCup event.

Fig. 6. A match in RobotStadium

b) Evolution:

Year	Number of participating teams
2008	15
2009	14
2010	7
2011	14

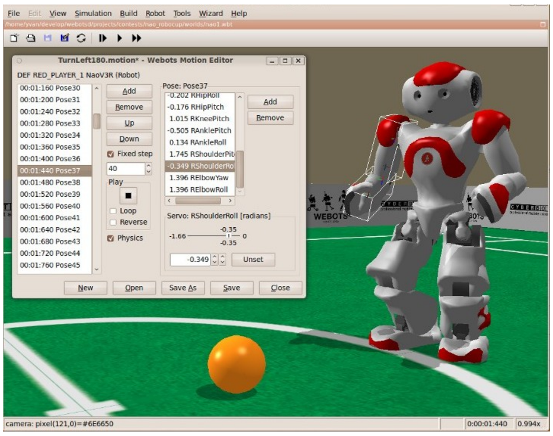


Fig. 7. Simulated Nao in Webots

Every year, participants to RobotStadium also take part into RoboCup Nao SPL. In 2008, CMRobokids (Carnegie Mellon University, USA) won RobotStadium and performed as finalist of the RoboCup Nao SPL. They notably used the same team strategy and representation algorithms in both competitions and had only to adapt movements and image processing for the real robots. The same year, Kouretes (Technical University of Greece) performed at the fifth and the third place in RobotStadium and the RoboCup Nao SPL respectively.

III. Analysis of in-class experiments

The curriculum has been tested in high-school and university. Teachers provided us with assessment results and comments.

A. High school

For his master degree at EPFL, Nicolas Heiniger led in-class experiments from November 2008 to February 2009 in seven groups coming from different Swiss high schools. Two sets of exercises were designed to assess the part of the curriculum that does not need C programming (eight in the beginner section, one in the novice section). A group consisted on average of ten pupils and each group was given one of the two sets. All in all 64 pupils participated in the experiment and answered Heiniger's survey. Statistics and a detailed analysis are available on Cyberbotics website [16]. The results of the survey led to improvements of the shape, the content and the usability of the curriculum, especially of the beginner section. This educational project led to the first publication of the curriculum on wikibooks. We only give here a short account:

- The exercises were considered as easy by the pupils, the average mark being 3.32 in a range from 0 (too easy) to 6 (too hard).
- More than 80 % agreed that they learned something about robotics through the exercises.
- More than 90 % agreed that using the real robot was more pleasurable than simulation alone.
- More than 70 % agreed that Webots was easy to use.

Nicolas Heiniger observed that the curriculum cannot substitute the guidance of a teacher and that using the text alone requires a reasonable level of autonomy. Indeed, the pupils largely preferred asking him directly every time they faced problems although the provided text contained answers to most of their questions. The feedback sessions also revealed that they expected a longer oral introductory presentation.

B. University

The intermediate section of the curriculum is used at the EPFL as a support for master courses on robots navigation every year since 2009. Here we collect observations of assistants having taught with it:

- The exercises are easily completed by a large majority of students.
- The simulation part takes most of the time and is crucial to experiment intensively with algorithms.
- The set of exercises could be extended with an example of navigation in a maze needing more intricate behavioral modules.

IV. Conclusion

In accordance with our main goal, the curriculum is being used both in high school and university and evolves on the impulse of teachers and wikibook contributors. Regarding the expert level, we observed that worldwide competitors renew their participation every year to Rat's life and RobotStadium. By continuously rising the level of these competitions, they keep them attractive to robotics experts and researchers looking for benchmarks. However intermediate levels of learning should be assessed to validate the entire document. We expect that its open format and its frequent improvements will help to reach every target audience in a near future.

References

[1] Bailey, T.; Durrant-Whyte, H., "Simultaneous localization and mapping (slam): part II". Robotics & Automation Magazine, IEEE, 13(3):108-117, Sept. 2006.

[2] Braitenberg, V., "Vehicles: Experiments in Synthetic Psychology", Cambridge, MA: MIT Press, 1984.

[3] Curriculum PDF. http://www.cyberbotics.com/curriculum, 2011.

[4] Curriculum wikibook. http://en.wikibooks.org/wiki/Cyberbotics\%27_Robot_Curriculum

[5] Cyberbotics Ltd. http://www.cyberbotics.com, 2011.

[6] Dissanayake, M.; Newman, P.; Clark, S.; Durrant-Whyte, H; Csorba, M., "A solution to the si- multaneous localization and map building (SLAM) problem. Robotics and Automation, IEEE Transactions on, 17(3):229-241, June 2001.

[7] Dodds, Z.; Greenwald, L.; Howard A., Tejada, S.; Weinberg, J., "Components, Curriculum, and Community: Robots and Robotics in Undergraduate AI Education". In AI Magazine 27(1):Spring 2006.

[8] Durrant-Whyte, H,; Bailey, T., "Simultaneous localization and mapping: part I". Robotics & Automation Magazine, IEEE, 13(2):99-110, June 2006.

[9] e-puck web site. http://www.e-puck.org, 2011.

[10] Gaudiello, I.; Zibetti, E.; Carrignon, S., "Representations to go: learning robotics, learning by robotics". Proceedings of SIMPAR 2010 Workshops Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, Darmstadt (Germany) November 15-16, p. 484-493, 2010.

[11] Gerstner, W., "Supervised Learning for Neural Networks: A Tutorial with JAVA exercises". To appear in: Intelligent Systems, An EPFL graduate course. D. Mlynek and H.-N. Teodorescu (Eds.).

[12] Eberhart, R.; Kennedy J., "A new optimizer using particle swarm theory" in Proceedings of IEEE International Conference on Neural Networks (ICNN '95), vol. 4, p. 19421948, 1995.
[13] Harvey, I. ; Husbands, P. ; Cliff, D. ; Thompson, A.; Jakobi, N., "Evolutionary Robotics: the Sussex Approach. In Robotics and Autonomous Systems", v. 20, p. 205224, 1997.
[14] Johnson, J., "Children, robotics, and education, Artificial Life and Robotics, 7 (1-2), p. 16-21, 2003.
[15] K-Team, Hemisson. http://www.k-team.com/kteam/index.php?site=1&rub=22&page=16&version=EN, 2011.
[16] Heiniger, N., "Design and Assessment of a Robot Curriculum based on the E-puck Robot and Webots", master project EPFL-Cyberbotics, 2009. Available at http://www.cyberbotics.com/files/papers/, 2011.
[17] Mataric, M. J., "Robotics Education for All Ages", Proceedings, AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education, Palo Alto, CA Spring 2004.
[18] Mondada, F.; Floreano, D, "Evolution of neural control structures: Some experiments on mobile robots". Robotics and Autonomous Systems, 16, p. 183-195, 1995.
[19] Nolfi, S.; Floreano, D., "Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines", MIT Press, 2000, ISBN 0-262-14070-5.
[20] Poli, R.; Kennedy, J.; Blackwell, T., "Particle swarm optimization". Swarm Intelligence, 1(1):33-57, August 2007.
[21] Michel, O.; Bourquin, Y.; Baillie, J.-C.,"RobotStadium: Online Humanoid Robot Soccer Simulation Competition". In Robocup 2008: Robot Soccer World Cup Xii, vol. 5399, 2009, p. 580-590 Springer-Verlag New York, 2009.
[22] Michel, O.; Rohrer, F.; Bourquin, Y.," Rat's life: A cognitive robotics benchmark". In European Robotics Symposium 2008, vol. 44, p. 223-232, 2008. Springer-Verlag New York, 2008.
[23] RoboCup website. http://www.tzi.de/spl/bin/view/Website/WebHome, 2011.
[24] Rohrer, F.,"Curriculum of exercises for e-puck and Webots", master project EPFL-Cyberbotics, 2008. Available at http://www.cyberbotics.com/files/papers/
[25] Sergeyev, A. and Alaraje., N., "Promoting Robotics Education: Curriculum and State-of-the-Art Robotics Laboratory Development", The Technology Interface Journal, Volume 10 Number 3, 2010.
[26] Shi, Y.; Eberhart, R., "A modified particle swarm optimizer". Proceedings of the IEEE International Conference on Evolutionary Computation, p. 69-73, 1998.
[27] Siegwart, R. ; Nourbakhsh; I. R. , "Introduction to Autonomous Mobile Robots". MIT Press, 2004.
[28] Tejada, S.,"AI, AIBO and ART: Inspiring Interaction with Computer Science", AAAI Spring Symposium on Using AI to motivate greater participation in Computer Science, Stanford, CA 2008.

Promoting scientific thinking with robots

Juan Pablo Carbajal
Artificial Intelligence Laboratory
University of Zrich
Email: carbajal@ifi.uzh.ch

Dorit Assaf
Artificial Intelligence Laboratory
University of Zrich
Email: assaf@ifi.uzh.ch

Emanuel Benker
Synthes GmbH
Switzerland
E-mail: benker.emanuel@synthes.com

***Abstract*—This article describes an exemplary robot exercise which was conducted in a class for mechatronics students. The goal of this exercise was to engage students in scientific thinking and reasoning, activities which do not always play an important role in their curriculum. The robotic platform presented here is simple in its construction and is customizable to the needs of the teacher. Therefore, it can be used for exercises in many different fields of science, not necessarily related to robotics. Here we present a situation where the robot is used like an alien creature from which we want to understand its behavior, resembling an ethological research activity. This robot exercise is suited for a wide range of courses, from general introduction to science, to hardware oriented lectures.**

I. The Braitenberg vehicle exercise

A simple self-made robotic platform built by the authors was used for the activity. The robot had two wheels, each one actuated by a DC motor. Two light sensors [1] were attached to the robot. The robot was controlled by a simple on-board program that defined a relation between inputs coming from the sensors and output signals sent to each motor.

We provided the robot with the behavior of Valentino Braitenberg's vehicle number 3 [2]. The light sensors of the robot commanded the rotational speed of the two motors. The connection was inhibitory, meaning that when the sensor measured light, the speed of the motor connected to it was reduced proportionally to the sensor's output. This sensor-motor configuration generates a light following behavior (Figure 1). More details about the robot, the control program and how to reproduce this exercise are explained in later sections.

Next we describe how we used the robot to engage students in scientific thinking. This exercise was part of a class on modeling mechatronics systems that took place at the Baden-Wuerttemberg Cooperative State University Loerrach, Germany. The students were mainly 3rd year bachelor students. The objective of the activity was to let students find out the sensor-motor relationship by means of hands-on experimentation and free exploration. The students had to create a hypothesis about the controller implemented in the robot and later verify it through experiments.

a) Introducing the robot.: The activity started with the presentation of the robot and a demonstration of its behavior when a light was placed in front of it. The robot moved by default in a straight line, and when it passed close to the light it turned towards it. The robot was even able to track the light (this depends on the sensor gain and motor speed, therefore it requires calibration prior to demonstration). This light loving behavior, though simple, always captivates the audience as well as the teachers.

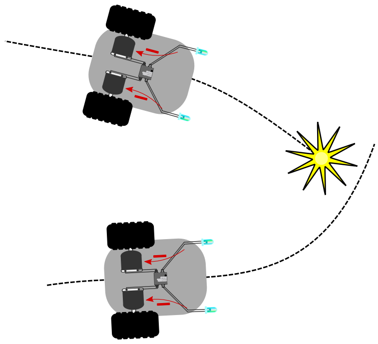


Fig. 1. Braitenberg vehicle 3, the light lover. Each sensor reduces the speed of the motor on its side proportionally to the measured light intensity. The figure shows the qualitative behavior of the robot: it moves towards the light and tends to stop close to it.

b) The assignment.: After several playful tests with the light, the students were asked to give explanations, in the simplest possible way, about the controller implemented in the robot such that it shows this behavior. Additionally, they were asked to propose an experiment that tests their explanation. In other words, they were asked to develop a model of the internal works of the robot and to produce a hypothesis verifiable through experimentation. The robot allowed us to create a complete and interesting research situation. At this point, to avoid diverging explanations, we suggested to the students to focus on the role of the sensors.

c) Hands on.: The production of models and tests was done in small groups (3-4 people) and we let the students form the groups by themselves. During this phase, we visited each group and discussed their ideas to assure the experiments will help deciding whether a given model should be discarded or not. It is important to remark that we did not correct the models, since any model is just an approximation. Thus, we just suggested changes in the model to simplify the verification process. After several minutes of group discussion, the groups presented their models, the experiment to be conducted on the robot and what they expected to observe. Since the number of available robots was enough, the students were able to perform their experiments. Otherwise the teacher could select a few experiments and try them in the robot.

d) The closure.: The conclusion of the activity is left to the criteria of the teacher. In our case, due to the lack of time, we explained the controller and introduced Braitenberg's ideas. In other circumstances, we would have requested the students to produce a short report of the experience and postpone the explanation to the next class.

II. Robot Hardware

The custom robotic platform is shown in Figure 2. Next, we describe the hardware that is needed to reproduce the robot exercise just described.

As mentioned above, the robot has two motors that can rotate individually at different speeds. Light sensors are placed at the right and left front side of the robot. These sensors can detect a light source within a range of about 10 cm and were previously calibrated by the students by measuring the output voltage as a function of the distance to a light source. The robot control program was implemented such that each light sensor is *connected* (via the controller) to one motor and influences its speed directly. Whenever a light sensor measures light the speed of the motor is reduced proportionally to the sensor's measurement. The less light a sensor detects, the faster the motor rotates and vice versa.

A commercial Arduino control board (http://www.arduino.cc) was used to control the robot. Figure 2 shows the components of the robot. Six rechargeable batteries are used for power supply. An USB communication unit is used for programming and monitoring the control board. Two light sensors provide sensory input to the control board which controls the two motors and wheels through the motor driver component. Since the robot was designed to be used in different experiments [5], it can actually be equipped with many more sensors and therefore the controller board is more powerful than what would be required for the exercise presented here.

Nowadays materials to build these robots are abundant. For example, ready-to-use chassis can be acquired from online retailers such as Maker SHED (http://www.makershed.com) or Dwengo (http://www.dwengo.com). Tutorials on how to build robots are easily accessible as in Make magazine (http://makezine.com) or any of the many blogs on robotics. The approximate material cost for the robot presented here is EUR 140. Information about how to rebuild the robot and the required software libraries is available on Dorit Assaf's website (http://www.embed-it.ch).

III. Robot Software

The Arduino project provides open source programming libraries and software development kits. Alternatively, the MATLAB language offers the ArduinoIO[1], an easy to use programming interface. Below we show a snippet of the C code used for a controller that produces Braitenberg's vehicle 3 behavior. Lest the unexperienced user find the source code daunting, the Arduino project offers very easy tutorials to get started.

The digital output that controlled the wheels had an 8-bit resolution (it can produce 256 different values), therefore the speed of the motor is given by a number between 0 and 255, being 127 the middle value or half-speed. The preamble of the code includes our custom libraries needed and initializes sensors and motors. Next a function to set up the robot is defined, it initializes the default robot speed (127 = half-speed) and forward direction. After this function is executed, the continuous loop() routine starts. There, the sensor values of light sensor 1 and light sensor 2 are read and saved in the variables `sensorValue1` and `sensorValue2`. The sensor values range from 0 (dark) to 1023 (bright). The map() function, as its name indicates, maps the first two arguments (the sensor range [0,1023]), to the range [255,0]. This value will replace the default speed of the robot via the setSpeed() function, therefore, bright light will slow down the robot.

```
// Include libraries with functions
// for the specific sensors and motors
#include <LightSensor.h>
#include <DCMotor.h>
// Define sensors and motors
// Two sensors connected to pins 1 and 2.
LightSensor lightSensor(1, 2);
// Connect pins to motor driver component
DCMotor motor1(12, 8, 10);
DCMotor motor2(18, 19, 11);
void setup()
{
// This function is loaded
// at startup and after each reset
  // Set default speed of the motors
  motor1.setSpeed(127);
  motor2.setSpeed(127);
  // Set default direction of rotation
  motor1.setDirection(FORWARD);
  motor2.setDirection(FORWARD);
}
void loop()
{
// This function runs while the robot is alive
  // Read sensor values
  int sensorValue1 =
  lightSensor.readSensorValue1();
  int sensorValue2 =
  lightSensor.readSensorValue2();
  // Convert sensor values to motor speed
  int newSpeed1 =
  map(sensorValue1, 0, 1023, 255, 0);
  int newSpeed2 =
  map(sensorValue2, 0, 1023, 255, 0);
  // Apply new speed values to motors
  motor1.setSpeed(newSpeed1);
  motor2.setSpeed(newSpeed2);
}
```

Litle more is needed to get the robot running. The source code is available on the website http://www.embed-it.ch together with some programming guidelines.

IV. Discussion and Conclusion

During the class we observed that students were fully engaged and were having fun. Based on their feedback, we attribute this to the presence of the robot, a non-standard tool for teaching.

The students produced creative models (with a tendency to complicated schemes), hypotheses, and interesting experiments. No group actually found the correct solution (Braitenberg's vehicle 3) or anything equivalent. However, one group proposed a feedback controller that, despite its complexity, seemed aligned with Braitenberg's ideas. Nevertheless, our

[1] MATLAB is a widespread scientific computing language, almost a standard in the scientific research community nowadays, http://www.mathworks.com.

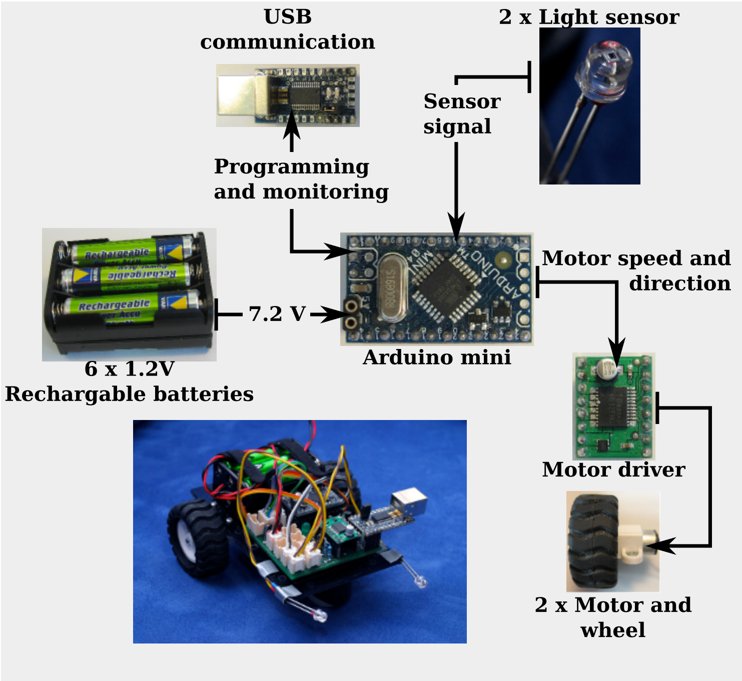


Fig. 2. The robot and its components. Six rechargeable batteries are used for power supply. An USB communication unit is used for programming and monitoring the control board. Two light sensors provide sensory input to the control board which controls the two DC motors and wheels through the motor driver component.

goal was to challenge the students and allow them to build their hypothesis based on hands-on evidence, therefore this goal was met. Using robots as a learning tool allows to prepare fully customizable class activities with different levels of difficulty and which can emulate real research situations.

The validation process, given that expectations were clearly stated, resulted to be fairly simple: either the model predicted the behavior or not. Several students showed determination to find a working model and automatically reworked theirs without being told to do so. We were surprised to note that the cycle: build a model, test it, rework the model; emerged naturally after the little push given when we described the activity to the students.

From the teacher's perspective this activity requires some extra work especially in the preparation phase. However, the effort was worth it and we encourage other teachers to try. A caveat of this kind of exercise is the difficulty to define criteria to grade a student's performance, due to the unstructured nature of the activity and the variety of possible solutions. This could be avoided by complementing the activity with a written report or a presentation. In the case where parallel activities are also performed such as calibration of sensors or construction of a speedometer[2], grading could be simplified.

A more physics based experience could be to set up *a robotic car crash* and engage students in a forensic physics experience, where they could determine initial speeds and directions or maneuvers made by the *artificial car drivers* [3].

We invite other teachers to try similar activities. We offer our support for the programming and assembly of the robot and invite the reader send us feedback.

[2]Students could build wheel speed sensors (speedometers) and verify the discussion in [4]

REFERENCES

[1] Mickey Kutzner, Richard Wright and Emily Kutzner, *An inexpensive LED light sensor*, Phys. Teach. 48, 341 (May 2010)

[2] Valentino Braitenberg, *Vehicles, experiments in synthetic psychology*,(MIT Press, Cambridge Massachusetts, 1986), p. 10.

[3] Arthur C. Damask, *Forensic physics of vehicle accidents*, Physics Today 3, 40 (March 1987).

[4] Clifton Murray, *Wheel Diameter and Speedometer Reading*, Phys. Teach. 48, 416 (September 2010).

[5] Assaf, D. and Pfeifer, R. (2011). *Robotics as Part of an Informatics Degree Program for Teachers* In Proceedings of Society for Information Technology & Teacher Education International Conference 2011 (pp. 3128-3133). Chesapeake, VA: AACE.

System of Indicators and Methodology of Evaluation for the Robotics in Classroom

Jonathan Ortiz[1,2], Raime Bustos[1,2], Antonio Ríos[1]

[1]*Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Chihuahua*
Chihuahua, México
[1]jortiz@chi.itesm.mx
[3]antonio.rios@itesm.mx
[2]*Centro de Investigación y Tecnología Aplicada*
Chihuahua, México
[2]raime.bustos@itesm.mx

Abstract— Robotics in the classroom is a multi discipline that has taken a global momentum because teaching science and technology creates a very large range of benefits. However, the benefit has not been clearly measured and defined because there is not a system of indicators and a standardized evaluation methodology for them. The present study defines a measurement Model of Robotics in the Classroom, validated by a methodology of experts, and a system of indicators.

***Keywords*— Measurement of robotics in the classroom, benefits of robotics, system of indicators of robotics.**

I. Introduction

In the early 80's, robotics started to be used for educational purposes in different ways, especially because it promotes engineering and science as an entertainment subject. [1]
In order to understand why there is an emphasis on robotics in the classroom and to be able to define the concept itself, it is necessary to delve in some aspects of education. Countries, institutions and individuals are facing significant challenges with regard to educational systems [2]. There is an inherent need for a flexible learning system, before this need of searching for new learning techniques, there is nowadays an emphasis on improving educational models based on practical models, which, promote the acquisition of knowledge in a tangible way for the student. In these new systems, knowledge should be conducted based on theoretical – practical models; in other words, it should exist equilibrium between the given theory and the practice, in order to strength the knowledge of the student.

Coupled with this, a factor that new generations are influenced by technology and digital age is presented. Technology native generation is defined as the generation born after the 1980 and 1990 digital boom; this feature makes an extraordinary fit for future technologies like no previous generation. It is said that this generation is the 18% of the world population, so its influence on the social and the educational context is very important to consider [3].

A theoretical - practical model, which meets the needs of integrating knowing and doing [4] and covers the technological requirements of new generations, is the incorporation of educational robotics, which is an interdisciplinary and inherent field of engineering that is formed by electrical, mechanical engineering and computer science, as well as, mathematics, physics, engineering systems and in some instances psychology, cognitive neuroscience and philosophy [5]. The extent of the problems presented by robotics motivates the development of the integration of knowledge and problem-solving methods from different ranges of approximation. The study of the discipline of robotics can give students a valuable perspective on systems integration and field experience in a real-world problem-solving.

It has been discussed that educational robotics offers great advantages and great benefits to improve the performance of the student without having a system of indicators to assess the impact of educational robotics applied to Mexican and international contexts. Companies and institutions, that handle this type of supplementary material for education, have not defined a methodological framework for assessment of their products or courses. Also, some authors managed only the before and after in students' grades or apply a test of knowledge, focusing on the obvious and direct benefits without assessing the full gamma advantages of this educational technique.

II. Model of Robotics in the Classroom

The absence of formal models of how to measure the benefits of robotics and the complexity of the modeling, led to investigate the most relevant variables to achieve a concise and synthetic model of robotics in the classroom, which incorporates a group of variables that demonstrate the benefits of it. Several qualitative and quantitative variables were identified, given the complexity of the design of the tool for measurement; a process of reducing the list of considered variables based on the experiences of various authors on robotics in the classroom was implemented. The identified variables are: *Creativity* [6-8, 19, 21, *Teamwork* [6-11,13,19,21-22], *Motivation* [6-7,9,13-14,21-22], *Problem Solving* [5-6,8-12,14-15,19-22], *Auto-Identification with*

Science and Technology [6-7,11-14,19], *Applied Basic Sciences* [6,11-15,19-20,22] and *Applied Engineering* [5-6,8-9,11-15,19-22].

The ordered variables based on their number of hits are: tied in first place, *Problem Solving* and *Applied Engineering*. *Team Work* and *Applied Basic Sciences* are tied in second place; *Motivation* and *Auto-Identification with Science and Technology* are tied in third place and *Creativity* in fourth place.

The Model of Benefits (variables) of Robotics in the Classroom is a construct resulting from the literature and it is shown in Figure 1, outlining how the variables interact with each other variables, variables that are considered property of robotics in the classroom. In a logical way, we can observe a direct relationship between *Applied Basic Sciences* and *Applied Engineering*. Its direct influence on the process makes them input variables in the model. Note that *Problem Solving* is closely related to *Motivation*, *Creativity* and *Teamwork*.

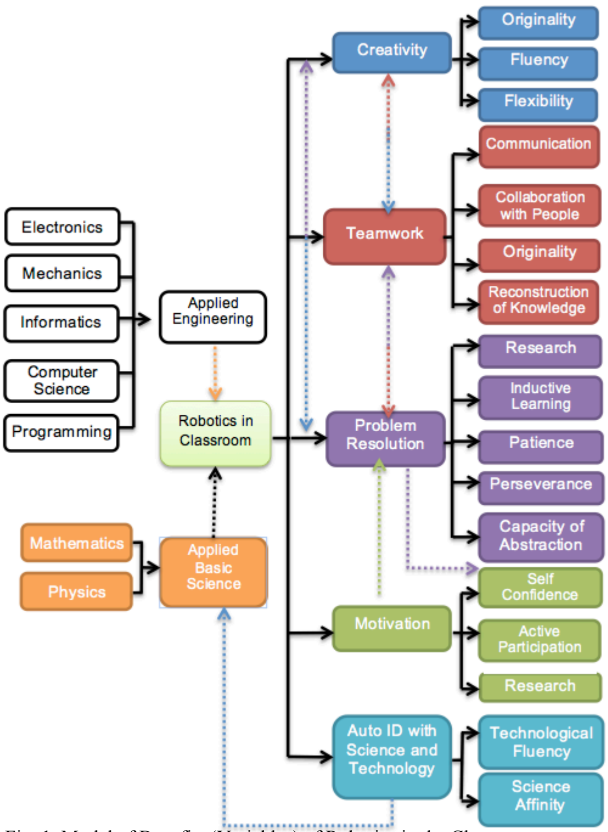


Fig. 1 Model of Benefits (Variables) of Robotics in the Classroom.

The Model of Robotics in the Classroom contains the set of variables and sub significant variables identified in the literature on educational robotics and pedagogical robotics. The ANOCHI method is used [16], to evaluate an idea, its validity and reliability through an expert consultation in a particular discipline, in this case, the model of robotics in the classroom. When the ANOCHI index exceeds 0.80, it is considered that there is a high or a very good match. After applying this methodology to the opinion of experts in robotics in the classroom, an ANOCHI index of 0.83 was obtained. This good value of reliability and agreement allowed to validate the model and to develop the instruments.

The conceptual definition and criteria for measuring variables and sub variables of the Model of Benefits of Robotics in the Classroom are explained below:

1. *Creativity*. Creativity is not only a set of techniques or inspiration of individuals, but also contains postulated observations, systemic and systematic observations, is not merely a brainstorming session, but is a group of innovative ideas that add value.
   1.1. *Originality*. Is the ability that valid proposals are novel [16].
   1.2. *Fluency*. Is the ability of the emergence of valid proposals aimed to overcome partial or totally the challenges in the classroom [16].
   1.3. *Flexibility*. Is the ability of the proposals to present different points of view leading to possible adaptations [16].

2. *Teamwork*. Learning how to work effectively in teams is an ingredient for success in many endeavors. Specific skills in teamwork include: generating and sharing new ideas, assigning roles and responsibilities, reconstructing knowledge through observation, imitation, conversation and other social cognitive processes [11].
   2.1. *Communication*. Freedom to express ideas freely and these are heard and valued.
   2.2. *Collaboration with Other People*. The interaction and exchange of ideas to arrive at a proposed solution to a problem or activity that is being treated.
   2.3. *Originality*. Described above in 1.1.
   2.4. *Reconstruction of Knowledge*. To obtain knowledge through observation, imitation and conversation with other individuals.

3. *Motivation*. Is to ensure that students full fill the requirements of their classes, projects and subjects without pressure and with responsibility, generating a learning environment in which the student can identify himself with the class.
   3.1. *Active Participation*. Student's interest in seeking knowledge or prove it either through technological means or speech. This implies, also, the interest of coming to class and participate in activities outside the classroom.
   3.2. *Research*. Students are interested in their work or project and spend extra time getting to know why or how things work.
   3.3. *Confidence*. The student's ability to feel safe when expressing their ideas or their achievement at the implementation of an activity.

4. *Problem Solving.* The process of understanding and proposing solutions to a given problem. This ability is worth the patience, perseverance, and learning from others through teamwork, creative and innovative ideas, individual motivation or group work when generating the solution. New forms of learning through research, inductive learning and guided discovery are created.
   - 4.1. *Research*. Described in 3.1
   - 4.2. *Inductive Learning*. Is the ability to solve problems through experimentation and to generate conclusions based on observations
   - 4.3. *Patience*. It is the ability of people to act calmly in given situations and continue to seek an optimal solution.
   - 4.4. *Perseverance*. It is the ability of people to continue to seek a solution without losing patience in time.
   - 4.5. *Abstraction capacity*. Is the ability to solve a given problem based on previous experiences or cognitive knowledge of the student.

5. *Auto Identification (ID) with Science and Technology*. Is the ability of empowerment with respect to science and technology. This includes the development and interest in technology, the confidence to work with technology and interest in further study in the future in some area of science and technology. That is, if the student considers himself in the future and if he feels himself capable of conducting technological explorations. [11].
   - 5.1. *Technology Fluency*. It is the student's interest in its development in technology and the confidence to work with.
   - 5.2. *Sciences Affinity*. Is the interest of the student to the study of basic sciences and thereby identify a prospective engineer.

6. *Applied Basic Sciences.* It is the knowledge the child generates from mathematics and physics to solve practical problems.
   - 6.1. *Math*. It is the knowledge generated from the properties and quantitative relationships between abstract entities (numbers, shapes, symbols).
   - 6.2. *Physics*. Knowledge that seeks that its findings can be verified by experimentation and the theory can make predictions for future experiments in areas such as mechanics, electromagnetism, among others.

7. *Applied Engineering*. It is the knowledge the student generates from electronics, mechanics, computer science, computer systems and programming to solve practical problems. These practical areas of knowledge are vital for the development of technology.
   - 7.1. *Electronics*. Ability to understand principles and foundations from the electronic area
   - 7.2. *Mechanics*. Ability to understand principles and foundations from the mechanic area.
   - 7.3. *Computer Systems*. Ability to understand the use and interaction with the computer.
   - 7.4. *Programming*. Ability to develop code, pseudo code or algorithms that generate a standalone solution of a given problem.

With the construction of the Model of Robotics in the Classroom, theoretical and conceptual bases are set in order to define robotics in relation with the specific benefits it sustains and is presented as an essential precondition for the development of a system of indicators and the methodology of evaluation presented below.

III. Methodology of Robotics in the Classroom

The following methodology is based on the premise that there is not a current model that includes indicators and measures of the benefits of robotics. For this reason, it is necessary to specify the importance of the experts' evaluation method to validate what the theory and practice mention. The methodology is based on the understanding that robotics in the classroom is not a direct form of education, but it is a supplement to an established educational system.

Figure 2 shows a graphic of the assessment methodology of robotics in the classroom. The column on the right shows the stages on which the steps of the graphic focuses and the stages are: the stage of distribution and application of knowledge, instrument development stage, followed by the stage of system indicators and finally, the stage of analysis of results.

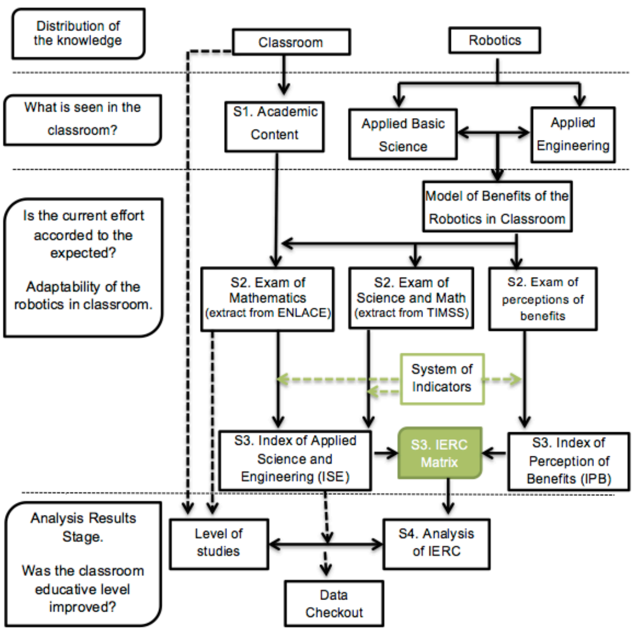


Fig. 2. Evaluation Methodology of Robotics in Classroom.

There are a huge variety of robotic programs and models of commercial robots. The intent of the proposed evaluation methodology is to present a generic model for any program or scheme used in a robotics course. It is important to note that the instruments are designed for the educational content of fourth grade with the aim to define the under study and allow a deeply evaluation of the model's results and its methodology.

Each stage has specific content and implies the use of their own tools but those tools do not exist, they have to be designed and validated according to the assumptions made in the previous chapter. It is important to highlight the difficulty of considering the intangibles elements, such as the Benefit Perceptions Index which requires the combined use of a set of qualitative and statistics techniques for its measurement. Moreover, the Applied Science and Engineering Index is easier to develop due to the abstraction of content of the ENLACE test (Spanish acronym of Evaluación Nacional del Logro Académico en Centros Escolares, National Assessment of Academic Achievement in Schools) and TIMSS exam (English acronym of Trends in International Mathematics and Science Study). Below, the content, the scope and the tools used in each one of the stages are considered.

A. Stage One (E1). Distribution and Application of Knowledge.

At this stage, the context on which the methodology is being applied is defined. At first instance it is imprecise to apply a methodology that is equally effective for an elementary school student and a junior high school student. Although, there are many common elements, the methodology is partially applicable in different educative levels because the information contents and cognitive process vary depending on the individual's age. This is why it is important to place and contextualize in a framework the object of study with whom; the robot in the classroom will be evaluated.

B. Stage Two (E2). Development of Instruments to Measure the Benefits of Robotics in the Classroom.

This phase begins with the input and output variables described in the Model of Variables of Robotics in the Classroom (Figure 1). The input variables are those inherently applied by robotics, which, in this case are: Engineering and Applied Sciences. The output variables are the benefits generated by the input variables. A series of instruments are designed to be applied to children and youth, based on the interrelationship of these variables and sub variables.

The designed instruments enable to obtain a quantitative and a qualitative scope, each one of these outcomes strengthens and supports an index that is used into the array of the Index of Effectiveness of Robotics in the Classroom (IERC). The aim of the qualitative method is to measure the intangible elements (*Creativity*, *Teamwork*, *Problem Solving*, *Auto ID Science and Technology* and finally *Motivation*) which generate the Benefits Perceptions Index (IPB), using statistical techniques to process the instrument.

Likewise, in the quantitative approach, math and science tests are developed from an extraction of the ENLACE Exam and the TIMMS test, to generate the Index of Applied Science and Engineering (ISE). These two tests were used because of their importance and institutional support. ENLACE is used for the Mexican government to standardize a measurement system comparable to the national level in study subjects especially math and Spanish. This test has been applied since 2006 in all of Mexico, with the advantage that it has a large amount of historical data for statistical validation and statistical analysis. Furthermore, the TIMSS test is an international exam that evaluates the math and science levels of performance of students in a given country. This test is implemented by the International Association for the Education of Educational Achievement (IEA) every four years (1995, 1999, 2003, 2007 and 2011) in countries that are willing to collaborate. The Latin American countries involved are Chile, Colombia, El Salvador and Honduras. In 2007, 425.000 students were assessed in 59 countries, obtaining the gradual progress of the groups where the test was applied. [17]. Likewise, it provides statistical data as well as ENLACE.

Three instruments are generated in order to measure the perception progress of the student and also, measure the effect of the implementation of the complementary robotic courses. This implies an initial, intermediate and final evaluation. Each instrument has 25 questions that directly involves the variables and sub variables in the Model of Robotics in the Classroom. Each instrument is introduced in a Liker scale form, from 1 to 5, where one represents a high perception and five indicates a low perception. The purpose of having three instruments is to determine whether or not there is an evolution, which can be plotted within the IERC Matrix (see Figure 3). The matrix and its representation and interpretation are developed in stage three.

C. Stage Three (E3). Indicators System and Computation of the results.

The scientific basis of this step lies in the importance of generating a set of indicators and the inter relationship generated by the indices to present in a graphical manner the location of the robotics course into the IERC matrix, in other words, the necessary correlation that must exist between the student's perception and knowledge of engineering and science which are being taught. In this way both ISE and the IPB, as measuring factors, should manifest simultaneously and be commensurate in a matrix where we can easily observe the efforts of robotics in the classroom.

The IERC matrix shown in Figure 3, it is a graphical way to express the results of the evaluation on a given robotics course and it is designed not only to show the quality of the course based on the sciences level but also the level of perceptions of the benefits of the students involved in the course, the IERC matrix also helps to improve the course trough a series of recommendations related on which quadrant the course is. The IERC matrix has on the horizontal (x) axis the IPB and on the vertical (y) axis the ISE, which scale of values correspond to the maximum marks of their respective tools (ISE = 500, IPB = 500). As can be observed, the matrix is divided in four quadrants, the first quadrant corresponds to a high ISE and a low IPB; the second quadrant is the ideal quadrant, as it represents a high ISE and high IPB; the third quadrant is the opposite of the second quadrant, it represents a low IPB and a low ISE; the fourth quadrant indicates a low ISE and a high IPB, this quadrant actually indicates that the robotics course is taught merely for fun and serves a purpose merely of entertainment.

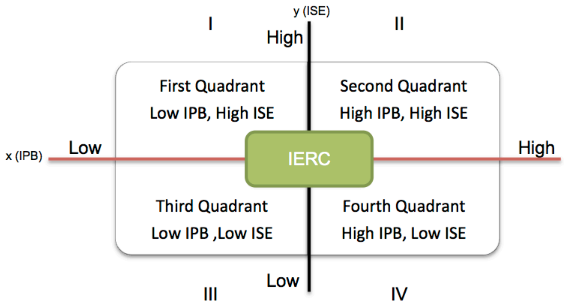


Fig. 3 Matrix of the Effectiveness Index of Robotics in the Classroom (IERC).

The impact of any course of robotics depends on its position within the IERC matrix; recommendations depending on the course location are as follow:

Quadrant I (low-high). This quadrant represents the situation in which the student is not interested in robotics courses even when their use and learning in science and engineering are good. The most likely reason is that the form of teaching is saturated with content, but in a monotone, boring way or that the teacher does not know how to guide the group. Another factor is that the robots used are not fun, either for being too simple or too complex, but it reflects that the student is not interested in using them. It is likely that the student is unwilling to return to the courses and thereby threatening the reputation of the company affecting the recruitment of more customers. It is recommended the review of the course structure and to find teaching techniques that are fun to put in practice the science and engineering content to amuse the child, but without neglecting learning. Also, we must see the instructor's pedagogical ability to transmit knowledge.

Quadrant II (high-high). This is the quadrant called ideal, since the efforts of engineering and applied sciences, their way of teaching, the theoretical content, and the use of the robots, are highly appreciated for the students and thus ensure the success of not only the learning of the student but also the course offerings from an organization and its reproducibility in the future.

Quadrant III (low-low). Is the quadrant in which non school of robotics would like to be. It indicates that the students are not interest in the course and also the course is not practical because is not generating any important knowledge. It is recommendable to restructure the course and to incorporate pedagogical techniques, motivational, of high content of science and engineering and incorporate better educative robots.

Quadrant IV (high-low). This is the quadrant that presents an inherent danger to the philosophy of robotics in the classroom, since it is used only in the form of entertainment. Probably the course is based in an autocratic philosophy and with guided instructions, not allowing the students developed their own knowledge through experimentation. As a result, the child is being entertained with novel and playful activities without fomenting any applied science and engineering knowledge. It is broadly recommended to integrate applied science and engineering knowledge to the course in a fun and emotive manner, and also train teachers.

In order to validate the application of the IERC Matrix, an investigation is carried out in public, federal and state schools in the city of Chihuahua, to determine the proper functionality of the tools and the generation of the indices (IPB and ISE). A total of 8 federal schools and 8 state schools in the fourth year participated.

D. Stage Four (E4). Analysis of Results.

The central idea of this stage is to analyze the results provided by the instruments through statistical methods. The computer program SPSS 17 from IBM is committed to the task. Among the tests being developed are: factor analysis, correlation tests and calculation of basic statistics such as Mean, Standard Deviation and Mode.

For the analysis of results, the premise departed from that there is no model to compare it with, for that reason, three questionnaires were designed for the IPB, to measure a gradual increase over a curve of knowledge or to position the group before the course in one IERC quadrant and a after the course in the same IERC matrix. However, the decision to establish a control group was taken; this means that in each one of the participating schools in the robotics program has at least one group of the same grade level outside the educational robotics program. That is, a group "B" who was not involved in robotics in a given school, the same initial and final tools were applied, similar to the ones applied to a group “A”, to evaluate the difference between taken or not taken a course in robotics and in this way, be able to validate the model and the methodology proposed by the author.

Besides the comparison with the control group, a comparative analysis with the historical results of the ENLACE exam and the overall hit rates of TIMSS was made, because it is not possible to determine the IPB for this comparative due to is not possible to apply the instruments to the students as long as they were evaluated for the IEA in an international level. The information is presented as a relationship between bar charts for explanatory and comparative reasons.

IV. Indicators System to Measure Robotic in the Classroom

The measurement of robotics in the classroom is conceive of two indices in order to generate one final index (IERC): The first index (ISE) is relative to the input variables that the theory supports what is robotics and the second index (IPB) is about the benefits or advantages that the same theory says that it generates in its application in the classroom. Both comprise a set of magnitudes where each one represents its utility, interpretation and mathematical proposal for its calculation. To determine the coefficient of each of the variables, we proceeded, first of all, to validate the model with experts (ANOCHI method) applying a questionnaire with a Liker scale from 1 to 7, where 7 indicates that the experts are in complete agreement and 1, indicates that the experts disagree completely. This scale is used to provide a wider range of opinion to the respondents. Then, each one of the variables was divided into equal percentages with respect to each one of the sub variables, later; this is multiplied by the average interview of the Liker scale, obtaining a weighted average that later was normalized to 100, in order to obtain a common scale in each variable.

As an example of the explained above, we are going to take the variable Creativity, it has as sub variables: Originality, Fluency and Flexibility (fig. 1). Once we applied the questionnaire based on Liker scale to the experts, we proceed to calculate the average of each sub variable trough all the five evaluated experts (i.e. Originality for expert one: 7; expert two: 7; expert three: 7; expert four: 7; expert five: 5, mean: 6.6). Then, having as consideration that the variable Creativity has three sub variables, we proceed to give a ponderation of 33.3% for each one in order to represent the variable as 100%. Later to obtain the weighted average of the sub variable it is calculated trough the multiplication of the mean of the experts average by the corresponding ponderated sub variable (i.e. Originality: 6.6 x 33.33% = 2.2). Now this last value gets normalized performing the sum of all the weighted average of each sub variable and considering the result as 100%, finally a cross- multiplication is used to calculate the factor for each sub variable. (i.e. Creativity = 6.33 , Originality = 2.2, Factor: 34.7%)

Following is the explanation of the equations used with their respective coefficients.

A. *Index of the Effectiveness of Robotics in the Classroom.*

$$\text{IERC} = \text{IPB} \times \text{ISE} \quad (0 \leq \text{IERC} \leq 250{,}000)$$

Where:
IPB: Index of Perception of Benefits.
ISE: Index of Applied Science and Engineering.

B. *Index of Perception of Benefits.*

$$IPB = \sum_{k=1}^{n}(CREA_k + TW_k + MOT_k + PS_k + AutoID_k) \quad (0 \leq IPB \leq 500)$$

Where:
k: students tested (k=1,…n).
n: number of participating students in the classroom.
CREA: *Creativity* variable.
TW: *Teamwork* variable.
MOT: *Motivation* variable.
PS: *Problem Solving* variable.
AutoID: *Auto Identification with Science and Technology* variable.

1) *Creativity.*

$$\text{CREA} = \sum_{k=1}^{n}\frac{(34.7\ x\ O_k + 34.7\ x\ F_k + 31.6\ x\ Fl_k)}{500n} \quad (0 \leq \text{CREA} \leq 100)$$

Where:
O: *Originality* sub variable.
F: *Fluency* sub variable.
Fl: *Flexibility* sub variable.
O, F and Fl are calculated trough mean formula: $\sum_{k=1}^{n}\frac{(O,F,Fl)_k}{n}$

2) *Teamwork.*

$$\text{TW} = \sum_{k=1}^{n}\frac{(24.6\ x\ Co_k + 24.6\ x\ Cl_k + 25.4\ x\ RK_k + 25.4\ x\ O_k)}{500n} \quad (0 \leq \text{TW} \leq 100)$$

Where:
Co: *Communication* sub variable.
Cl: *Collaboration* sub variable.
RK: *Reconstruction of Knowledge* sub variable.
O: *Originality* sub variable.
Co, Cl, RK and O are calculated trough mean formula: $\sum_{k=1}^{n}\frac{(Co,Cl,RK,O)_k}{n}$

3) *Motivation.*

$$\text{MOT} = \sum_{k=1}^{n}\frac{(33\ x\ AP_k + 35\ x\ R_k + 32\ x\ SC_k)}{500n} \quad (0 \leq \text{MOT} \leq 100)$$

Where:
AP: *Active Participation* sub variable.
R: *Research* sub variable.
SC: *Self-Confidence* sub variable.
AP, R and SC are calculated trough mean formula: $\sum_{k=1}^{n}\frac{(AP,R,SC)_k}{n}$

4) *Problem Solving.*

$$PS = \sum_{k=1}^{n}\frac{(19.1\ x\ IL_k + 21.6\ x\ R_k + 19.8\ x\ Pat_k + 19.8\ x\ Pe_k + 19.8\ x\ AC_k)}{500n} \quad (0 \leq \text{RP} \leq 100)$$

Where:
IL: *Inductive Learning* sub variable.
R: *Research* sub variable.
Pat: *Patience* sub variable.
Pe: *Perseverance* sub variable.
AC: *Abstraction Capacity* sub variable.
IL,R, Pat, Pe and AC are calculated trough mean formula: $\sum_{k=1}^{n}\frac{(IL,R,Pat,Pe,AC)_k}{n}$

5) *Auto Identification with Science and Technology.*

$$\text{AutoID} = \sum_{k=1}^{n}\frac{(50\ x\ TF_k + 50\ x\ AS_k)}{500n} \quad (0 \leq \text{AutoID} \leq 100)$$

Where:
TF: *Technological Fluency* sub variable.
AS: *Affinity to the Sciences* sub variable.
TF and AS are calculated trough mean formula: $\sum_{k=1}^{n} \frac{(TF,AS)_k}{n}$

C. Applied Science and Engineering Index (ISE).

$$\text{ISE} = \sum_{k=1}^{n} (ABS_k + AE_k) \quad (0 \le \text{ISE} \le 500)$$

Where:
ABS: *Applied Basic Sciences* variable.
AE: *Applied Engineering* variable.

1) Applied Basic Sciences.

$$ABS = \sum_{k=1}^{n} \frac{(50\ x\ Mat_k + 50\ x\ Phy_k)}{200n} \quad (0 \le ABS \le 250)$$

Mat: *Mathematics* sub variable.
Phy: *Physics* sub variable.
Mat and Phy are calculated trough mean formula: $\sum_{k=1}^{n} \frac{(Mat,Phy)_k}{n}$

2) Applied Engineering.

$$\text{AE} = \sum_{k=1}^{n} \frac{(21\ x\ EE_k + 21\ x\ ME_k + 21\ x\ IE_k + 19\ x\ CS_k + 18\ x\ PL_k)}{200n} \quad (0 \le \text{AE} \le 250)$$

Where:
EE: *Electronics Engineering* sub variable.
ME: *Mechanical Engineering* sub variable.
IE: *Informatics Engineering* sub variable.
CS: *Computer Science* sub variable.
PL: *Programming Languages* sub variable.
EE, ME, IE, CS, and PL are calculated trough mean formula: $\sum_{k=1}^{n} \frac{(EE,ME,IE,CS,PL)_k}{n}$

In this section, the developments of the theoretical and operational fundaments models of the Model of Robotics in the Classroom and each one of its methodological stages, which are considered important contributions for the present investigation as well as the development of instruments that it applies, were demonstrated. Finally, the design and metrics of the model, which are imperatives to document and evaluate with respect to the applicability degree of the present proposal were presented.

V. Implementation and Scope of Study

For implementation reasons of the research, fourth grade of elementary school was defined as the object of study. Being 16 participating schools and a total of 960 students, of whom half are children taking robotics course and half are not taking them. This sample was calculated from a universe of children in the fourth year of 18,900 people in the city of Chihuahua and with a confidence level of 95% and a confidence interval of 5%, resulting in a sample of 377 as minimum. As for the courses, robotic models are used based on the B.E.A.M. philosophy (English acronym of Biology, Electronics, Aesthetics and Mechanics) [18], these robots are chosen for their low cost and because no computer is required for their operation. So, the courses are complemented with science practices. In this object of study is where the evaluation methodology and the developed instruments are implemented for later processing and analyzing applying the IERC matrix.

Two different exams were applied, the first consists in an exam of 44 questions based on TIMMS and ENLACE and trough the formulas explained before it generates the ISE, it worth to mention that the ISE exam has not questions related directly to the robotics course, it has only general questions related to science and mathematics. Later, for IPB it is generated also trough the formulas and is based on a questionnaire of 22 interrogations with a Liker scale from 1 to 5. So far only 6 schools have been evaluated, figure 4 shown the percentage of improvement for the tested school related to the group the has taken robotics against the group on the same school the has not taken (Formulas: IPB Robotics / IPB No Robotics and ISE Robotics / ISE No Robotics).

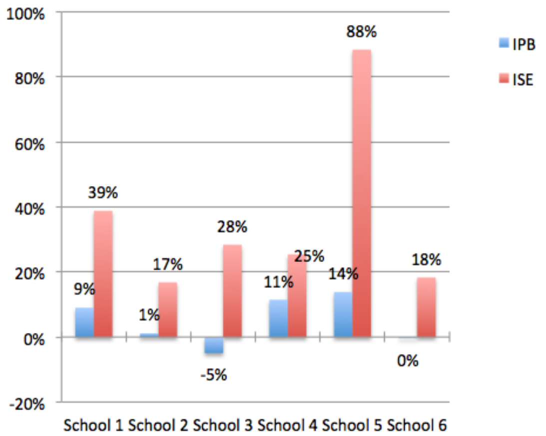


Fig. 4 Percentage of improvement per school for both indices.

As it can be observed, all the evaluated schools shown an improvement on the level of ISE for at least 17%, but the results varies on the IPB, where 5 of the 6 evaluated schools shown an improvement of the perceptions of the benefits. Moreover it is necessary to shown the IERC for the evaluated schools, figure 5 presents the visual interjections for both indices mentioned above.

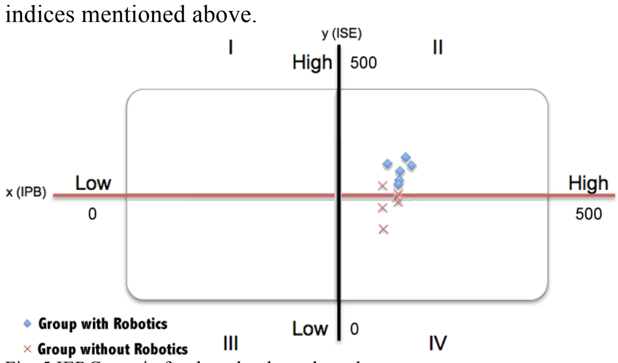


Fig. 5 IERC matrix for the schools evaluated.

Figure 5 shows that all the schools has a good level of IPB but it represents considerable differences related to the ISE level, all the six schools that has taken a robotics course are in the quadrant II (high IPB, high ISE) while five of the six schools that has not taken robotics are in the quadrant IV (high IPB, Low ISE).

VI. Conclusions

So far, a methodology proposal and a system of indicators used to measure robotics in the classroom were presented. Also, one of the most important premises of this research: generate a model of benefits of robotics in the classroom, which is endorsed by experts in this field was shown. In addition, this research supports the central idea that the intangible benefits, such as *Creativity*, *Teamwork*, *Problem Solving*, *Auto ID with Science and Technology* and *Motivation* are measurable from a qualitative point of view and applicable to statistical methods in order to interlink them with the teaching material which is being used based on a robotics course of *Applied Science* and *Applied Engineering*, in order to generate the quantitative part of an oriented metric and presented in a quadrant system. It is important to mention that the level of sciences and engineering was measured in an indirect way, it means, that the applied exam has not questions related to the course. For the IERC matrix (fig. 3), the main reason for its conceptualization is to present in a graphical form where the current efforts of the robotics course of a given institution are placed and with this, determine in a clear and concise manner the direction and magnitude that is having on the group, so the improvement area or content opportunity or improve the quality of the courses on their teaching technique can be predicted. It get concluded that the efforts in robotics can be measured from the quantitative and the qualitative approaches, related to level of science and engineered and the perceptions of benefits respectively and can be used to improve the course level in a graphical way as the IERC matrix proves.

Acknowledgements

This author wishes to thank Raime Bustos, PhD., and Antonio Ríos, PhD., for their confidence and unconditional support in carrying out this project, also, thanks to Nancy Moreno, PhD., for her interest and academic support in the formation of the Masters Degree of the author.

Special thanks to José Luis Bruni, Sergio Tejeda, Mauricio Sánchez, Javier Rubio and Olmo Moreno for acting as educational robotics experts evaluating the Model of Robotics in the Classroom with their invaluable experience and dedication in this area.

Finally, largely thanks to the National Council of Science and Technology (CONACYT), the Government of the State of Chihuahua and the Secretary of Public Education for their support for the implementation of the project through their representatives: Carlos Calleros, Lic. and Hector García, PhD. (CONACYT), from the Federal System of primaries to: Marisa Tirado, Eng., for her invaluable support and assistance, to Prof. Martin Ortiz and Prof. Graciela Rodríguez and from the State System: Prof. Guadalupe Zapata and Prof. Myriam García

References

[1] J. Malec, “Some thoughts on robotics for education,” *2001 AAAI Spring Symposium on Robotics and Education*, 2001.

[2] C. Buiu, “Hybrid Educational Strategy for a Laboratory Course on Cognitive Robotics,” *IEEE Transactions on Education*, vol. 51, 2008, pp. 100-107.

[3] D. Tapscott, *Growing up digital*, New York; Mexico City: McGraw-Hill, 2008.

[4] J. Dewey, *Democracy and Education*, Chapter 25. Theories of Knowledge. The Macmillan Company, 1916.

[5] P. Robinette, R. Meuth, R. Dolan, and D. Wunsch, “LabRat TM : Miniature Robot for Students , Researchers , and Hobbyists,” *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1007-1012.

[6] A.D. Alimisis and C. Kynigos, “Chapter 1 Constructionism and robotics in education,” *Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods*, School of Pedagogical and Technological Education (ASPETE), 2009.

[7] M.U. Bers, I. Ponte, K. Juelich, and J. Schenker, “Teachers as Designers : Integrating Robotics in Early Childhood Education,” *Information Technology in Childhood Education*, 2002, pp. 123-145.

[8] I. Jormanainen, Y. Zhang, K. Kinshuk, and E. Sutinen, “Pedagogical Agents for Teacher Intervention in Educational Robotics Classes: Implementation Issues,” *2007 First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL’07)*, Mar. 2007, pp. 49-56.

[9] L. Xuemei, L. Jin-hu, and D. Lixing, “Constructing Innovative Laboratory through Intelligent Learning Objects,” *2009 International Conference on Industrial Mechatronics and Automation*, 2009.

[10] S. Bostock, “Constructivism in mass higher education: a case study,” *British Journal of Educational Technology*, 1998, pp.225-245.

[11] I.R. Nourbakhsh, K. Crowley, A. Bhave, E. Hamner, T. Hsiu, A. Perez-Bergquist, S. Richards, and K. Wilkinson, “The Robotic Autonomy Mobile Robotics Course: Robot Design, Curriculum Design and Educational Assessment,” *Autonomous Robots*, vol. 18, Jan. 2005, pp. 103-127.

[12] A. Vitko, L. Jurišica, A. Babinec, F. Duchoň, and M. Kľúčik, “Some didactic aspects of teaching robotics,” *1-st Slovak-Austrian International Conference on Robotics in Education*, 2010.

[13] S. Papert, *Mindstorms: Children, Computers, and Powerfull Ideas*. Second edition, New York: Basic Books, 1993.

[14] M. Sánchez, Ambientes de aprendizaje con Robótica Pedagógica (tesis inédita de maestría). Universidad de los Andes. Colombia, Bogotá, 2003.

[15] E. Ruiz-Velasco, *Educatrónica, Innovación en el aprendizaje de las ciencias y la tecnología*, Distrito Federal: Díaz de Santos, 2007.

[16] J.A. Araujo, *Propuesta metodológica para realizar planeación estratégica creativa* (tesis de doctorado), Universidad de la Habana, La Habana: Cuba, 2009.

[17] Eduteka, Prueba TIMSS 2007, preguntas de ciencias y matemáticas Recuperado el 7 de mayo de: http://www.eduteka.org/Timss2007PreguntasGrado4.php , 2011.

[18] M. Tilden & D. Hrynkiw, *Junkbots, Bugbots and bots on wheels, building simple robots with BEAM Technology*, Berkley: Osborne, 2002.

[19] I. Verner and D. Ahgren, “Conceptualising educational approaches in introductory robotics,” *International Journal of Electrical Engineering*, 2004, p. 183.

[20] P. Robinson, “Robotics education and training: a strategy for development,” *The Industrial Robot*, vol. 23, 1996, pp. 4-11.

[21] F. Wyffels, M. Hermans, and B. Schrauwen, “Building robots as a tool to motivate students into an engineering education,” *1-st Slovak-Austrian International Conference on Robotics in Education*, 2010.

[22] D. Ahlgren, “Meeting Educational Objectives and Outcomes through Robotics Education,” *Automation Congress, 2002 Proceedings of the 5th Biannual World*, 2002, pp. 395 - 404.

The Minor Specialization Robotics at FEE CTU in Prague

Miroslav Kulich, Libor Přeučil, Karel Košnar, Tomáš Krajník, and Jan Chudoba
Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics
Technicka 2, Prague 6
Czech Republic
Email: {kulich,preucil,kosnar, tkrajnik, chudoba}@labe.felk.cvut.cz

***Abstract*—The Faculty of Electrical Engineering (FEE), Czech Technical University in Prague offers seven master degree study programs. Although classical setup of study plans does perform well, recently founded study branches of the "Open Informatics" program employ a novel approach, that provides new, individually configurable study plans. These study plans offer education which is not strictly binded to already existing programs but truly diverse across existing study branches. This allows more efficient tailoring of individual needs and interests of students. The programs bring up an opportunity to select and add a minor specialization to an existing major study branch. Flexibility of this setup combines the teaching process itself with access to the state-of-the-art knowledge in the field of latest research results in the domain. The following paper presents core concepts and ideas of the minor entitled Robotics, which has launched in 2010 and guides students from fundamental concepts of information processing in robotics and basic robot control to latest approaches to robot autonomy, cognition, collective robotics and intelligent mobile robotics.**

I. Motivation

Advanced robotics with all its aspects definitely belongs to one of the most multidisciplinary subjects as it crossbreeds knowledge from plenty of very diverse fields. The contributing fields range from classical mechanical engineering, mechatronics, electronics and control engineering which form up the classical solutions in the domain (i.e. industrial robots) and end up in the computer science area. As the latter mainly stand for the latest hi-tech of advanced cognitive systems - intelligent and autonomous robots, these represent the cutting edge of the nowadays robotic technologies.

Considering the robot technology being truly interdisciplinary field, it does exhibit needs for either afore mentioned expertise: (1) Expertise in the classical robotic disciplines (e.g. mechanical and electrical design) and (2) Expertise in advanced robot control, comprising of modern data and knowledge processing (covering sensing, data interpretation, robot planning and scheduling, cooperation and coordination, etc.) which make the robot cognitive, autonomous and therefore capable of fulfilling complex tasks in uncertain conditions.

As modern robotics desires expertise from diverse fields, which can not be efficiently maintained by a single expert, it becomes reasonable to reflect this specific fact also in educational programs in the robotics domain. Although the extreme range of desired expertise can be well treated via proper distribution of specific expertise amongst multiple entities, this is possible only on the condition that these are able to formulate and communicate the problems and their solutions. In other words, such experts besides their core expertise are expected to have also certain amount of (even of an overall level) expertise in potentially neighboring fields. As an illustrative example could be a case of an autonomous robot design. This task will definitely require participation of an expert in AI, whose skills in robot hardware design and sensing constrains may be very shallow. This fact imposes many possible inefficiencies in resolving the given task together with other, highly specialized experts in diverse fields as electronic and mechanical engineering. But providing this expert with, even a basic, knowledge from these tackled and neighboring fields bridges this expertise gap and speeds up efficiency of communication, locks out possible misunderstandings and therefore minimizes the so called over the wall types of problems. The consequences are straightforward: (1) excellent coupling of student/expert needs and future interests through less constrained and flexible range of offered topics to study, what basically leads to educating (2) well focused experts, being capable of (3) the best efficiency at problem solving.

The afore sketched backbone observations and ideas were taken as the core motivation for design of innovative, less constrained and much more open and flexible study plans. The herein elaborated domain of robotics seems to have been the proper choice due to its wide range of the incorporated expertise from many other domains.

Here, in particular the minor specialization Robotics has been designed for students willing to apply their theoretical knowledge from the informatics domain into the field of advanced robotics. The aim is to combine and extend theoretical knowledge and software skills, the both gained in the major study branch, with the ability to use and develop robots operating in real environments. Graduates are expected to understand processing of uncertain information and decision-making processes and will be able i.e. to design, develop and implement these to embody robot autonomy and cognition. Added value of the acquired skills stands in their ability to deal with decision-making processes for robotics and related processing of information collected from real environments, which represents a superstructure of the deterministic data-processing procedures. On the other hand, this minor intentionally avoids addressing issues of mechanical, electrical or electronics design of the robot or its parts. It concentrates on building skills, how to interpret the data obtained, at being only aware of possible constrains originating from real world conditions, hardware properties, etc. The minor targets implementation and management of the planning and decision-making processes necessary to ensure objective-oriented behavior of such robotic system.

II. Overview

As to the afore sketched motivation, and to satisfy the given limit of four courses assigned to the minor-dedicated space, the courses according to the breakdown depicted in Table I are offered. Moreover, not to constrain the students a priori, the given minor specialization (as well as other minors) is being recognized even retroactively - for the cases, the participants have decided for particular minor even after having the corresponding courses already passed.

TABLE I: Recommended study plan of the minor Robotic together with time donations of lectures and labs.

1th semester	**Practical robotics** 6 lectures × 90 min 14 labs × 135 min	
2nd semester	**Automatic Control** 28 lectures × 90 min 14 labs × 90 min	**Intelligent Robotics** 14 lectures × 135 min 14 labs × 90 min
3rd semester	**Mobile and Collaborative Robotics** 14 lectures × 90 min 14 labs × 90 min	

Practical Robotics course is an introduction to the filed of robotics and common robotic problems. This course gives an overview of the algorithms and methods solving the basic problems of path planning, collision avoidance, mapping, localization and exploration. The complex robotic task is presented to motivate students to gain deeper understanding. As the students are solving the given robotic task, they discover problems of real-world applications. As the students try to find a way to overcome these problems, they gain not only the knowledge but also the skill to use the knowledge in practical application.

Automatic Control represents a foundation course of automatic control. It introduces basic concepts and properties of dynamic systems of physical, engineering, biological, economic, robotic and informatic nature and explains principles of feedback and its use as a tool for altering the behavior of systems and managing uncertainty. Classical and modern methods for analysis and design of automatic control systems are introduced as well. Students targeting continuing study of systems and control are expected to build on ideas and knowledge gained herein through the succeeding advanced courses. Students of other branches and programs will find out that automatic control is an inspiring, ubiquitous and entertaining field worth of a future cooperation.

Intelligent Robotics course teaches general principles allowing to build robots perceiving the surrounding world, undertaking self-decisions and planning activities to achieve given goal(s) and even to modify the environment. Various architectures of robots with cognitive abilities and their realizations are introduced. The studied material is applicable in more wide manner for building intelligent machines in general sense. Students have access to and experiment with robots in practical assignments.

Mobile and Collaborative Robotics course integrates and extends the knowledge and skills gained in previous courses. Whereas the *Automatic Control* introduces the control theory of dynamic systems and the Intelligent Robotics deals with the general principles of the robotics and is more focused on the wide area of manipulators, *The Mobile and Collaborative Robotics* focuses mainly on the problematics of the mobile robots. Contrary to stationary robots, the mobile robots operate in the common environment and deal with uncertainty in a larger degree. Therefore this course focuses on the methods and algorithms for processing data affected with noise, representing uncertain information, and planning under uncertainty. As the mobile robotics advances from single robot to cooperating groups of multiple robots, the principles of communication, coordination and cooperation increase their importance and become an important part of the course. The students verify their knowledge gathered in all the courses of the Robotics minor by solving the state-of-the-art problems of the mobile or collaborative robotics.

As majority of the afore listed courses are of standard shaping, being mainly overtaken from other existing study branches, their selection is driven purely by the goal to gather sufficiently wide range of robotic-related expertise, which may future experts need for this domain. A remarkable novelty in this concept is brought in by founding an introductory course entitled *Practical Robotics*. The course has been composed having the mission to be a motivation, a trigger point, for further deeper studies of robotics. Therefore, no specific prerequisite type of knowledge are required (with the exception of basic programming skills and mathematics and physics background, which are common at the branch of electrical engineering and computer science studies anyhow). The course deals with carefully selected topics, avoiding possibly demanding theoretical elaborations. The aim is to introduce in a stepwise manner design and development of an intelligent robot, to simulate its behavior and to port these solutions onto a real experimental robot. On the way to the final solution, the course participants discover and face problems which invoke interest to a deeper study of the robotic domain. The following chapters describe in brief the existing setup and contents of this course and comprise the conditions and early experiences and future developments after its first run.

Administrative constrains requires to build the minor from already available courses, if possible, which is motivated by willingness to not increase number of courses taught at the university. Therefore, we decided to use *Automatic Control* and *Intelligent Robotics* courses that are general courses taught at CTU several years, although they were modified in order to reflect actual needs of new study programs, which they are a part of and two new courses. Even though *Mobile and Collaborative Robotics* course is newly created, it was also added as a part into *Cybernetics and Robotics* study program and therefore it was accepted to be contained in the minor. *Practical Robotics* is a new course designed especially for needs of the robotic minor and a special exception has been afforded from the council of the study program for the course.

This paper focuses on description of mobile robotics courses, which were newly introduced the last year. While section III describes *Practical Robotics*, section IV relates to the *Mobile and Collaborative Robotics* course. Teachers' experience and topics for improvements of courses contents are described in section V.

III. Practical robotics

The course aims to create an interest in the ideas and possibilities of intelligent mobile robotics. It should motivate students to ask questions, think over solutions of nontrivial robotic problems and to look forward to further advanced and specialized courses. Moreover, the course should mediate practical skills in the area of robot navigation in a complex task, from design of robot architecture, sensor data processing and model building to planning and intelligent decision making.

To fulfill the aforementioned aims, the emphasis is given to individual student's work under teacher's supervision in the laboratories while the role of lectures is to provide a theoretical background to the tasks the students solve. The course consists of six theoretical lectures (in the first six semester weeks, one lecture per week) lasting 90 minutes each and fourteen lab sessions (one per each semester week). One lab session lasts 135 minutes. This organization allows the students to acquire necessary knowledge to the problem in the first half of the semester and to focus primarily on solving the problem in the remaining time.

At the beginning of the course, a complex task comprising of several fundamental robotic problems is presented to the students in the form of the game. This means that the students form several teams, each team consisting of two or three students. Each team solves the whole task so that at the end of the semester each team has implemented its own solution. The solutions of particular teams can be thus compared via competition.

The task in the winter semester 2010/2011 - *Mine searching* - was inspired by the exploration problem:

Suppose a robot operating in an unknown environment (i.e. the robot has no a-priory map of the environment). There is an unknown number of mines randomly placed in the environment. Implement a client application for the Player/Stage system[1] that navigates the robot in the environment in order to detect and find all the mines in the shortest time. The robot is equipped with odometer providing robot's position (but can be subject of errors), laser range-finder measuring distance to surrounding obstacles, and mine detector detecting mines close to the robot.

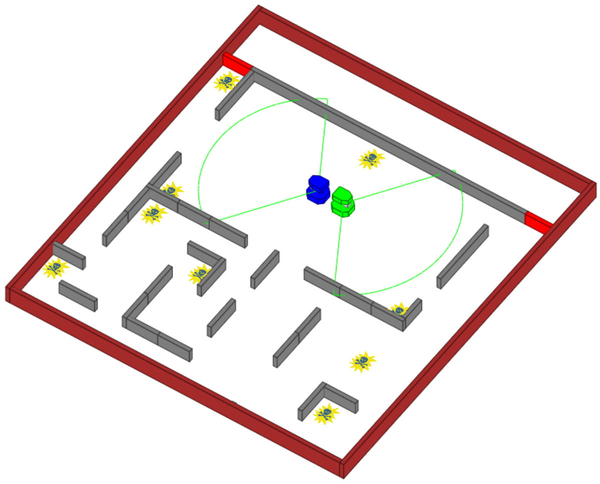


Fig. 1: Configuration of obstacles and mines for the competition.

A final evaluation of the course has a form of a graded assessment. It consist of the following criteria:

- behavior of the robot during final competition,
- presentation of the functioning application to the teacher,
- understanding of the presented code, and
- fulfilled protocols/questionnaires that describe solutions of particular steps of the solved task as well as student's experience with solving these steps.

The task can be divided into several essential subproblems, each of them is discussed at one lecture and then solved by students in the labs in the groups of two or three. Description of the subproblems together with corresponding course schedule is presented in the following subsections.

A. Introduction, task formulation

The first lesson is introductory, where the basic terms are defined (robot, cognitive robot) and essential robot architectures according to information processing are introduced. Moreover, primary sensors used in robotics, their principles and characteristics as well as fundamental terms and manners of wheel kinematics are discussed. Finally, the task to be solved is formulated and analyzed and an application architecture is introduced.

The lab presents basic software environment, tools and libraries that are expected to be used during task implementation. A special attention is paid to the Player/Stage system, its extensions and a basic skeleton of the application prepared by the teacher. In the second part of the lab session the students solve three simple tasks in order to familiarize with the tools and to demonstrate that they understand main principles of implementation of a robotic application. The aim of the first task is to port `laserobstacleavoid` (which is an example code distributed with the Player) into the prepared skeleton application. In the second task, the students familiarize with the `MineDetectorProxy` and modify their application to detect mines and report their positions. Finally, `Graphics2dProxy` is used in the third task for on-line drawing of laser range-finder measurements.

B. Planning on a binary grid

The lecture in the second week is concerned with planning a path of a robot in the two-dimensional world. An important issue of each planning algorithm and the first topic for discussion is the representation of the operating environment. A grid-based representation (i.e. the representation, where the environment is regularly split into cells; each cell representing the corresponding part of the environment) has been chosen from several reasons. First of all, this representation is easy to understand, its implementation is straightforward and also algorithms running on it are not complicated. Moreover, the grid is a general approach and with different meaning of values in the cells it can be used as a supporting structure for many robotic algorithms. The grid is thus main data structure in the *Practical robotics* course and the students will become acquainted with its different forms as they solve the particular subproblems.

The lecture continues with motivation of path planning and its connection to and cooperation with other modules. Dijkstra algorithm is then described in details as well as its features and comparison to A*. Adaptation of Dijkstra's algorithm results to real world conditions is discussed – (a) usage of Minkowski sum of the grid and a robot model that provides collision-free path for a disk robot, (b) smoothing of the generated path with Bresenham's algorithm [2].

Two labs are dedicated for implementation of the presented algorithms. The students get prepared maps of the environment in the text form together with routines for reading a map and building the grid. Dijkstra's algorithm and Minkowski sum are implemented from the scratch (although the code of a binary heap, which is used as a priority queue, is provided). Implementation of Bresenham's algorithm for line drawing is also provided, but the students are requested to modify it for the path planning problem. The students work only with the provided maps and routines and they are not requested to connect their codes into Player/Stage. This should be done in the next labs.

C. Collision avoidance

The next step of building the exploration behavior is traversing the found path by the robot. This incorporates detecting obstacles with sensors and avoiding them. The widely used Vector Field Histogram (VFH) and its derivative VFH+ [3] are presented in the lecture.

Realization of the collision avoidance takes two labs sessions. Besides implementation of the VFH+ algorithm the students are requested to integrate the code for path planning and obstacle avoidance into the skeleton application so that they are able control the robot in the Player/Stage environment. The VFH+ algorithm has several parameters that have to be set carefully, therefore tuning of the parameters is an inseparable part of the work and the parameter values are one of expected results of student's work. As implementation of the VFH+ algorithm is an integral part of the Player/Stage distribution, students can look into this implementation (although plagiarism is not allowed) and compare behavior of the implementation and parameter setting with their own.

D. Mapping and localization

Simultaneous localization and mapping (SLAM) has been one of the extensively studied robotic problems last years. Time allocated for the task in the course is not sufficient to solve the problem in its full complexity and moreover difficulty of the problem exceeds expected limits of an introductory course. On the other hand, determination of a robot position and building of a model of the working environment are necessary components of the exploration tasks. Localization and mapping are therefore not presented as an integrated SLAM approach, but as two independent components that can be solved separately. This simplifies the original SLAM problem, but it is still suitable for the exploration as specified in the first lecture.

The first part of the lecture is dedicated to mapping based on occupancy grids [4]. A probabilistic model of one range measurement and Bayesian approach to integration of a measurement into the occupancy grid are presented. The second part then introduces the continuous localization method based on cross-correlation of histograms built from range data [5]. This method has been chosen for its simplicity and straightforward implementation. Moreover, one-dimensional histograms are used as a key data structure for the localization algorithm, which is also the case of VFH+. The students can therefore reuse their code from the previous work and think about different applications of this data structure in robotics.

The students have three lab sessions to implement learned algorithms. Implementation of occupancy grid mapping is relatively not time consuming, one lab is enough for it. Other two lab sessions are dedicated to the localization problem.

E. Exploration and application integration

In the next lecture, exploration problem is defined and Yamauchi's frontier based exploration [6] is introduced. The approach is based on processing of an occupancy grid so the lecture shows another possible usage of this data structure.

The labs concerning the exploration topic have three sessions. Besides implementation of the particular exploration steps (frontier detection, evaluation, and selection) the students integrate their code created in the previous labs into a client application for the Player/Stage system.

F. Final demonstration, competition

The last lesson is different from the previous ones as it does not describe any theoretical problem. Instead, the lecturers draw from long-term participation at different robotic competitions (FIRA robotic soccer, Eurobot, and Robotour) and present acquired experience and observations. Practical issues of building a mobile robot for this kind of competition are mentioned, including hardware design, proper sensor selection, software architecture, sensor data processing, navigation, and cognition functionalities. Moreover, aspects concerning project management, scheduling and realization for a team of students building a robotic system are presented from a practical point of view. The aim of the lecture is to show that a nice theoretical solution is not enough for "real-world" problems and that a simple and robust approach gives better results in many applications than a sophisticated, generally-usable solution. The lecture should also motivate the students to further study of robotics, to participate in robotic competitions as a member of a department team or to join other robotic activities at the department.

The remaining labs in the semester are dedicated to work completion and testing and presentation of the work to the teacher. The presentation is oral, where the students show their code to the teacher and explain key parts of the implementation. The teacher can ask questions to ensure that each student understands the code.

The final presentation of student's work and competition of the implemented clients are planned for the last lab session. Before that, the students deliver source codes of their application to the teacher (commit their code into subversion repository). The teacher then compiles the application on its computer which guaranties that all applications run in the same environment.

The competition is organized in several rounds, where two teams play against each other in one particular round and every team plays against all other teams during the competition (the map, i.e. placement of obstacles and mines is the same for all rounds). One particular round consist of two games. At the beginning of the game, robots of the competing teams are placed at predefined positions. The aim of the game is to find as much mines as possible during the defined time (3-5 minutes). If the robot enters on a mine it is penalized by stopping its motors for a defined time period. The opponents change their positions in the second game and the number of found mines adds up with the first game. The team with higher number of found mines in both games wins the round.

G. Questionnaires

The teams are requested to fulfill questionnaires prepared by the teacher that overview their work on the particular subproblems.

The first questionnaire concerns path planning as described in section III-B. The students have to run their code on the prepared map with defined start and goal robot positions, generate requested outputs and images, and to insert the following images into the prepared document:

- the structural element for Minkowski sum,
- the map for a disk-like robot (application of Minkowski sum),
- the path found by Dijkstra's algorithm, and

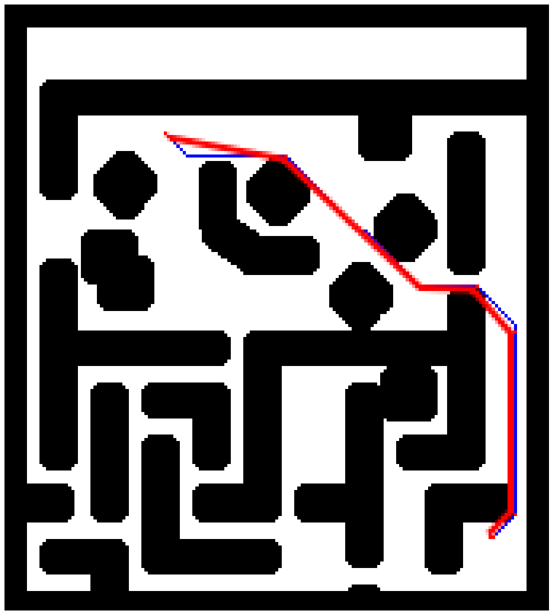


Fig. 2: A part of the Questionnaire 1. Path found by Dijkstra algorithm (blue) and the smoothed trajectory (red)

- the smoothed path (i.e. the path after application of Bresenham's algorithm).

Moreover, the students have to discuss whether the smoothed path is optimal (shortest possible) and why.

The second questionnaire deals with robot control and obstacle avoidance as presented in section III-C. The teams have to describe the key parameters of VFH+ algorithm (according to their opinion) and their setting and meaning. Furthermore, they are requested to discuss features of the algorithm (e.g. behavior of the algorithm in particular situations, etc.) and draw trajectories traversed by a robot when following the path generated in the Questionnaire 1.

IV. Mobile and Collaborative Robotics

Mobile and Collaborative Robotics course is intended to be an advanced course as it concludes the the whole robotic minor. The course is focused on mobile robotics, it introduces mobile robot architectures together with control methods aimed to achieve autonomous and collective behaviors for mobile robots. Methods and tools for data acquisition and processing are presented herein with the overall goal to resolve the task of autonomous navigation for mobile robots comprising the tasks of sensor fusion, environmental modeling including localization and mapping approaches [7]. Besides sensor-processing related tasks, methods for robot trajectory planning are introduced. The central topic of the course stands in specific usage of the afore methods capable of execution with groups of robots and taking the advantage of their cooperation and coordination in groups. Therefore, multi-robot systems are introduced, key aspects and problems of their design as well

Fig. 4: The SyRoTek arena and robots

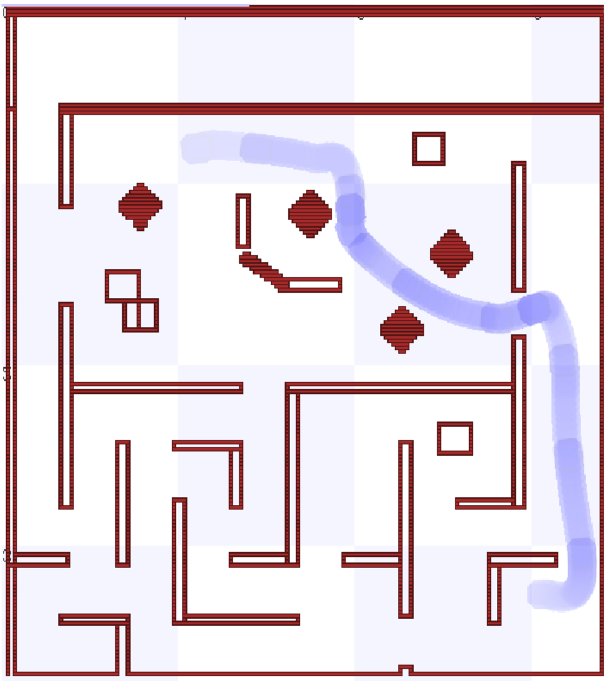


Fig. 3: A part of the Questionnaire 2 - a trajectory traversed by a robot.

as architectures and essential components of these systems are presented. Attention is also paid to planning and task allocation in multi-robot systems. Finally, fundamentals of robotic swarms are introduced.

The above described content is split into 14 lectures (one per each semester week), each lecture lasting 90 minutes. The course also contains 14 laboratory sessions with the same time donation as the lectures. It is assumed that students passed all other courses in the minor and that they have fundamental knowledge of general robotic systems and problems. Moreover, many of them obtain a “hands on” experience with some robotic system or tasks during their study with seminar works, bachelor or diploma thesis. Therefore, setting yet another “introductory” task to be solved within the labs brings students no (or little) new knowledge nor experience. To attract the students, the labs don’t follow the structure of the lessons. Instead, the students solve a task according to their knowledge, capabilities, intentions, and choice. The task is primarily selected from the pool of tasks/problems prepared by the teacher, but the students are not limited to this pool as they can choose their own task after negotiation with the teacher. The task specification is based on recent journal or conference papers that are expected to be implemented by the students and their topics go beyond the topics presented at the lectures. The students form groups by two or three solving one task together, which allows them to deal with more complex problems and to split the work. To the typical task belong for example cooperative environment cleaning, cooperative coverage, indoor pursuit/evasion game, decentralized planning for flocking of swarm robots, coastal navigation, etc.

The labs are organized in a form of Open Laboratory where the students work independently under teacher’s supervision. Teacher’s role is thus to help the students to understand the given papers, and to discuss solutions of problems not clearly explained in the papers. The students are not restricted to use some software and tools, although the Player/Stage is recommended for majority of tasks. For example, students solving the flocking problem used Breve – a 3D simulation environment for multi-agent simulations [8] that is more appropriate to swarm robotics.

Evaluation of the course consists of two parts: an examination and an assessment. The examination has a form of a dialog between the teacher and the student, where theoretical topics presented in the lecture are discussed. To get the assessment the students have to present the working code of their algorithm and to answer teacher’s questions regarding the code and the algorithm. Moreover, the last laboratory is dedicated to the presentation of the solved problems and their solutions to other students. This allows the students to compare their work with others and to gain an idea what problems and how the other students solved. Preparing and performing the presentation of their work helps to improve student's presentations skills, which are generally low in Czech republic.

V. Conclusion

During the first year of the *Practical Robotics* course we recognized that implementation of all the particular steps was too time consuming and although software modules prepared by the teachers were available, the students were not able to finish the particular steps in time. The main difficulty appeared in realization of VFH+ algorithm, which took more than twice more time for some groups than expected. In our opinion, it was caused by not enough programming experience of the students. Because of that, the schedule of the labs was modified so that information about robot's position was provided, and therefore, the students didn't have to implement their own localization algorithm. More crucial was that the majority of teams had no time to run the code on real robots and to perform experiments with them. Only one team was able to experiment with a real robot. Student's applications were thus tested in a simulated environment and also the competition was performed in simulation.

As "playing" with real robots is one of the aims and motivations of the course, the schedule for the next years will be modified to give students more time to experiment with real hardware. Therefore, implementation of VFH+ will not be requested. Instead, the students will have to properly tune parameters of the VFH+ driver distributed within Player/Stage and/or compare its behavior with Smooth Nearness Diagram Navigation (SND) [9], which is another local navigation method.

Based on the previous experience [10] `subversion` version system [11] was used in both courses as a code repository and a tool for task commitment. Majority of the students had experience with this tool from other lectures, so they had no problem to use it actively. On the other hand, the students didn't like to use prepared libraries. For example, graphical visualization of output of the particular algorithms was provided in two ways: (1) by extension to Stage, and (2) C++ API for `gnuplot`. Instead, several groups wasted time with their own implementation of visualization.

In *Practical Robotics* course, hardware parts of the SyRoTek e-learning system [12] (see also Fig.4) were used by the students particularly. As the whole system is ready now, it will be used as a major teaching platform for both *Practical Robotics* and *Mobile and Collaborative Robotics* courses. This should improve productivity of the students, their immersion into solving of the allocated task, motivate them to finish all the task in time and to simplify their communication with each other and with the teacher.

As the minor runs only one year it is too early to talk about its general effects. On the other hand, discussions with the students show that the minor is interresting for them in general. We will see, whether this interest will project into increased participation of the students in robotics projects in the future. A good news is that many students of *Practical Robotics* look for the topic of their thesis in the robotics domain, but general conclusions about positive effects of the minor can be made after several years.

Although a structure of the minor has been dictated by an administrative limitations, mixture of general courses already taught at the university giving theoretical overview of robotics and control with one practical introductory course at the beginning and one special course dealing with mobile robotics at the end of the study is viable and gives students a general overview of the field. On the other hand, time capacity of the minor does not allow to go into deep details and therefore some methods and problems are mentioned briefly. For example, one of the hottest topics today, simultaneous localization and mapping, is only introduced in few minutes without explaining at least one method.

Acknowledgment

This work has been supported by the Ministry of Education of the Czech Republic, under the Project No. 6840770038 - Decision Making and Control for Manufacturing III.

References

[1] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proc. of the 11th Int. Conf. on Advanced Robotics*, 2003, pp. 317–323.

[2] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[3] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1998, pp. 1572–1577.

[4] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," in *Proceedings of the Sixth Conference on Uncertainty in AI*. 2929 Campus Drive, San Mateo, CA 94403: Morgan Kaufmann Publishers, Inc., 1990.

[5] G. Weiß, C. Wetzler, and E. von Puttkamer, "Keeping track of position and orientation of moving indoor systems by correlation of range- finder scans," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 1994, pp. 595–601.

[6] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*. IEEE Comput. Soc. Press, 1997, pp. 146–151.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. The MIT Press, August 2005.

[8] J. Klein and L. Spector, "3D multi-agent simulations in the breve simulation environment," in *Artificial Life Models in Software*, M. Komosinski and A. Adamatzky, Eds. Springer London, 2009, pp. 79–106.

[9] J. W. Durham and F. Bullo, "Smooth nearness-diagram navigation," in *Int. Conf. on Intelligent Robots and Systems - IROS*, 2008, pp. 690–695.

[10] J. Faigl, T. Krajník, K. Košnar, H. Szücsová, J. Chudoba, V. Grimmer, and L. Přeučil, "Mobile robotics at FEE CTU," in *Proc. the 1st Int. Conf. on Robotics in Education, RiE2010*. FEI STU, Slovakia, 2010, pp. 43–48.

[11] "http://subversion.apache.org," (accessed 8 June 2011).

[12] J. Faigl, J. Chudoba, K. Košnar, M. Kulich, M. Saska, and L. Přeučil, "Syrotek - a robotic system for education," in *Proc. of the 1st Int. Conf. on Robotics in Education, RiE2010*. FEI STU, Slovakia, 2010, pp. 37–42.

Robotics Education at Innsbruck Medical University

Michael Sieb

University Hospital for Visceral, Transplantation and Thoracic Surgery,
Innsbruck Medical University, 6020 Innsbruck, Austria
sieb@TiRoLab.at

***Abstract*—Robotics is having an increasing impact on life sciences, but is still not included in the curriculum of medical universities. In surgery, robots are used like complex instruments to enhance a surgeon's technical possibilities. To give our students the opportunity to learn about the abilities and limits of robots we run a "hands-on robotics workshop" at Innsbruck Medical University. Our aim is not to teach surgical procedures, but to construct robots. To our knowledge, this is the first robotics workshop held at a medical university. LEGO Mindstorms NXT was selected as the workshop platform for easy construction and programming. This paper presents the different goals and the replies received to a feedback questionnaire.**

***Keywords*— robotics, construction, workshop, surgery**

I. Background

One of the probably most cited robotics articles in 2007 was the one by Bill Gates that appeared in Scientific American with the title "A robot in every home"[1]. Today, the robotics industry is booming just like the personal computer business was 30 years ago. Consequently, the next question is, when will we have a robot in every operating room, in every lab, at every patient's bedside? Robots are already present in many places in a hospital: lab[2], transportation[3], operating room (general surgery[4], thoracic surgery[5], cardiac surgery[6], gynaecology[7], paediatric surgery[8], urology[9], orthopaedics[10], neurosurgery[11], radiosurgery[12]), telerounding[13] and telementoring[14]). They are also becoming involved in nursing[15] and rehabilitation[16].

From the standpoint of a surgeon, robots are complex instruments that enhance their technical possibilities, but are no substitute for good surgical knowledge and skills. For efficient and safe handling it is important to know their abilities and limits, especially when they are used on patients. At the moment, robotics is not a standard part of a medical school curriculum like e.g. pathology is. However, hospitals continue to introduce more and more applications for robots. Robotic surgical site training[17] includes surgical procedures and system handling, but not basic robotics like kinematics and path planning.

For this reason Innsbruck Medical University launched the elective course "Theoretical Surgery," which includes a hands-on workshop for medical robotics giving the theoretical background for today's robotics applications and future aspects. At this workshop students are given several challenges to solve by planning, building and programming their own robots.

II. Material and Methods

In 1980 MIT (Massachussetts Institute of Technology) Professor Seymour Papert was the first to devise a robotics kit for education[18]. Through a partnership between the LEGO Group and MIT, Mindstorms was born in 1984 and in 2006 the third generation of Mindstorms NXT was put on the market. Primarily intended for school kids aged about 12, it was happily adopted by universities for research and education purposes[19].
Several commercial robotics kits are available. We decided to use the third generation of LEGO Mindstorms, the NXT set, because we found it to be the most flexible for our purposes.

The NXT kit consists of a programmable brick holding a 32-bit ARM processor at 48MHz, an 8-bit Atmel AVR coprocessor at 8MHz, an LCD matrix display with 100*64 pixels, a Bluetooth and a USB2 interface, three motor driver ports with encoders and four sensor inputs: light, ultrasonic, sound and touch. Third-party companies are building enhanced sensors for the NXT system including GPS (Global Positioning System), compass, acceleration, pressure, color, temperature, pH and many more. There is a good open source IDE (integrated development environment) named brixcc[20] (see bricxcc.sourceforge.net) that permits robot programming with several programming languages like LASM (assembly-like), MindScript and NBC (script-like), NQC and NXC[21] (both c-like), c/c++, Pascal, Java[22] and Forth. For medical students without any programming skills the graphical languages NXT-G[23] and LabVIEW are available[24]. Robots are made with the LEGO Technic system, enabling a wide range of possibilities.

For the first robotics workshop (two units of 5 hours each) we invited our medical students and also biomedical informatics students from The Health & Life Sciences University UMIT[25]. Following a short (30 minutes) oral presentation about the NXT kit, the IDE software and organisation issues, we proceeded directly to the challenges.

Challenges:

Subjects from the operating room were selected as standard tasks on day 1 (line following to get into constructing and programming) and special medical issues on day 2. In every challenge students could receive up to four points by solving the challenge, with an additional point going to the fastest team (start to finish).

1.) In the first challenge we used a 10mm black marker to draw a floor plan showing two operating rooms, a corridor and the central sterilisation room (size: two tables). The challenge was to go from OR 13 to the sterilisation room without crossing wall lines and without using a sensor. The aim was to take used instruments to the sterilisation room following an operation.

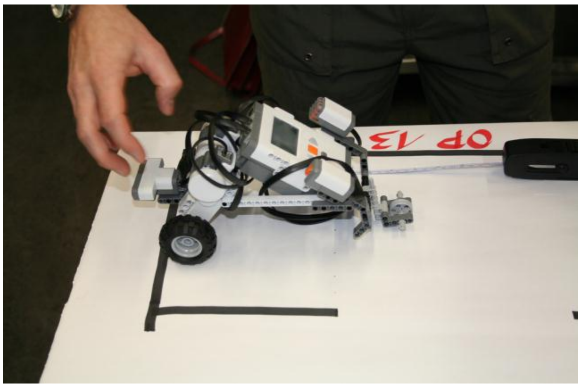


Fig. 1 Floor plan

2.) Same situation as in 1.), but this time a light sensor was used to follow the line drawn down the center of the corridor leading from the operating room to the sterilisation room.

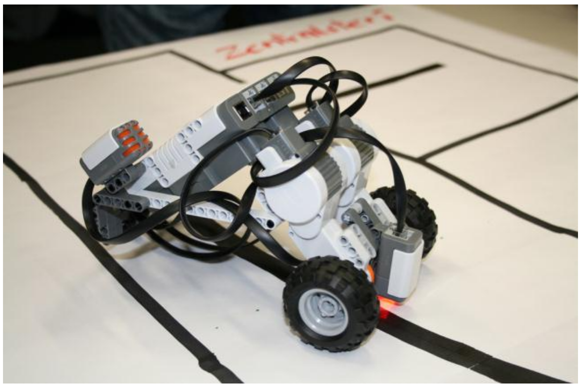


Fig. 2 Follow the line

3.) In this challenge a table represented the operating room: in the middle stood the operating table made of bricks. On the floor were small bricks representing dirty items which had to be cleaned (thrown from the table) without hitting the operating table and, of course, without the robot falling off the table.

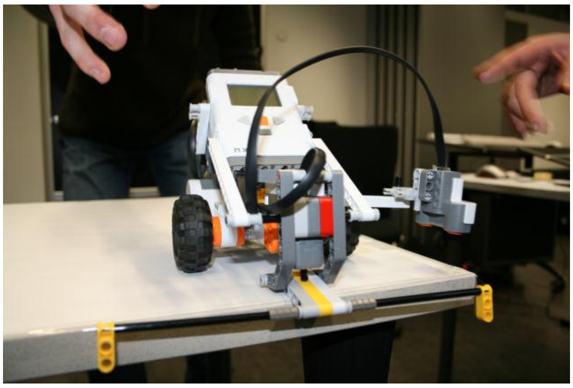


Fig. 3 Cleaning the operating room

4.) Final challenge: Design a robot for intravenous medication administration. Find the artificial forearm (white), artery (red) and the vein (blue). Identify the vein and place the yellow/medication brick on it.

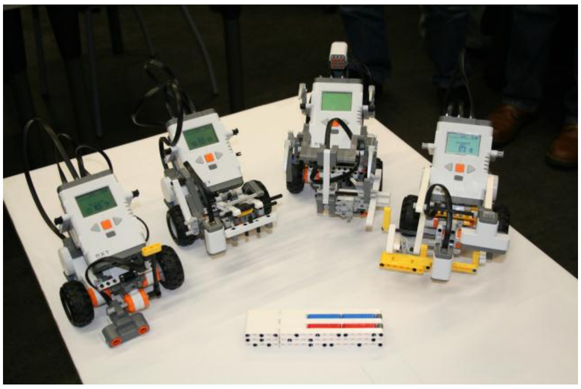


Fig. 4 i.v. Medication robots

The students elaborated on the LEGO Tribot (see mindstorms.lego.com/en-us/support/buildinginstructions/8527-/Tribot.aspx) to construct their robots. More photos and two videos from the workshop can be found at www.TiRoLab.at/imedrws/ .

To evaluate the workshop we distributed a questionnaire asking 27 questions divided into four groups: pre- and post-workshop robotics experience, support during the workshop, preferred programming language, and design of the challenges.

III. Results

All challenges were solved, even when the robots often did something unexpected (see Table 1) in the final rounds (contest) and failed to complete the task.

A maximum of 4 points was awarded for solving the challenge on time, less for a partial solution. The fastest team earned a bonus point, for a grand total of 5 points. Points were deducted for crossing a wall in challenge 1 or 2, touching the operating table or putting the medication in the wrong place. A robot that fell off the table meant immediate disqualification (zero points).

The interdisciplinary team (NOS) including informatics and medical students showed the best results. Another exciting fact was that all robots were made from the same LEGO NXT kit, but no two mechanical solutions to a challenge were even similar.

Team	**Challenge**			
	1	2	3	4
	Training / Contest	Training / Contest	Training / Contest	Training / Contest
Elkdestroyer	2 / 0	4 / 4	3 / 0	3 / 0
Bruteforce	2 / 0	4 / 4	3 / 0	4 / 0
NOS	4 / 5	4 / 5	3 / 0	4 / 0
DontFallFrom TheTable	1 / 0	3 / 1	3 / 0	4 / 0

Table 1 Challenge results

Questionnaire results:

All 11 participating students completed the questionnaire.

Figure 5 compares pre- and post-workshop robotics experience reported by the students themselves on a scale from 1 (excellent) to 5 (poor). Count shows the number of students. It shows that the students gained robotics experience during the workshop.

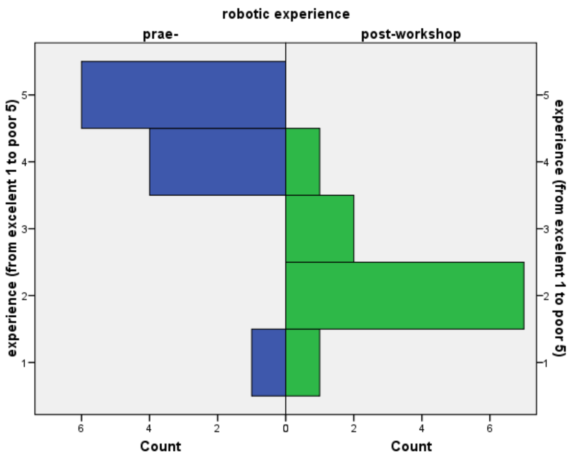


Fig. 5 Robotics experience

Figure 6 shows the replies to the items “Were the challenges didactically useful (learning progress)?” and “Did the workshop meet your expectations?” Grading and count are the same as in Fig. 5. The students rated the workshop as useful, and reported that it highly met their expectations.

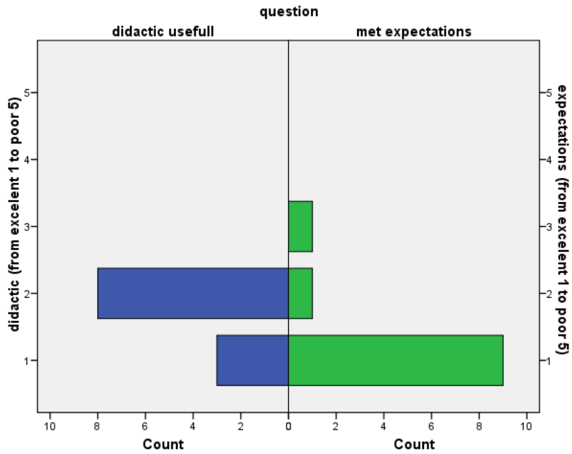


Fig. 6 Didactics and expectations

Further results are that the informatics students were quite happy with the NXC (Not eXactly C), but some would have preferred Java, which has more powerful libraries. The medical students wanted to program graphically, because they had poor know-how of syntax in textural languages.

IV. Conclusions

The LEGO Mindstorms NXT kit is a relatively inexpensive and powerful tool for running hands-on robotics workshops. The LEGO Technic bricks enable very flexible possibilities for constructing robotic models. A wide range of available programming languages ensures quick results and potent applications by running the robots with a graphical approach for novices and the possibility to code programs in C or Java for experts.

We found that interdisciplinary teams have a big advantage due to the transverse knowledge shift on the team and should be given priority. Thus, if you get the chance to cooperate with a technical university - take it! Fascinating team working processes started during the workshop and will hopefully continue. The first challenges showed us that the next to the last program was usually the best solution. The conclusion for the future is that teams need more time to solve individual tasks and team size will be enlarged to four members to ensure enough manpower for constructing and programming in parallel.

The workshop is an opportunity for medical students to make mistakes and learn from them. In hospitals they have to work with the state of the art. Software engineers have a much more intuitive access: they look inside the debugger to see what's wrong and fix it. When errors are allowed, they can help overcome borders to create new things, new approaches and optimize tools for a special task. Despite all the theory, robotics is great fun. If your robot never fell off the table, you weren't trying hard enough.

Today, robotics is widely used in medicine. Even if we don't yet have a robot in every operating room or at every patient's bedside, they are already in every automated lab. Early adopters show us where the way could go, and if we are to believe robotics associations, robots will be standard equipment in the next 20 years, like television and mobile phones are now. Robotics education at medical universities or during hospital residency can help turn users into knowing operators, who understand a little more of the underlying technical stuff. In this way, error handling derives from error understanding and hopefully leads to safer applications for our patients.

Acknowledgment

I am greatly indebted to emeritus Prof. Raimund Margreiter MD, former head of Visceral, Transplantation and Thoracic Surgery at Innsbruck Medical University, for giving me the opportunity to lecture on Theoretical Surgery and robotics workshops at our medical university.

References

[1] Gates B. A robot in every home. Sci Am 2007; 296(1):58-65.
[2] Wheeler MJ. Overview on robotics in the laboratory. Ann Clin Biochem 2007; 44(Pt 3):209-218.
[3] Lob WS. Robotic transportation. Clin Chem 1990; 36(9):1544-1550.
[4] Taylor GW, Jayne DG. Robotic applications in abdominal surgery: their limitations and future developments. Int J Med Robot 2007; 3:3-9.
[5] Augustin F, Schmid T, Bodner J. The robotic approach for mediastinal lesions. Int J Med Robot 2006; 2(3):262-270.
[6] Woo YJ. Robotic cardiac surgery. Int J Med Robot 2006; 2(3):225-232.
[7] Magrina JF. Robotic surgery in gynecology. Eur J Gynaecol Oncol 2007; 28(2):77-82.
[8] Meehan JJ, Sandler A. Pediatric robotic surgery: A single-institutional review of the first 100 consecutive cases. Surg Endosc 2007.
[9] Atug F, Castle EP, Woods M, Davis R, Thomas R. Robotics in urologic surgery: an evolving new technology. Int J Urol 2006; 13(7):857-863.
[10] Adili A. Robot-assisted orthopedic surgery. Semin Laparosc Surg 2004; 11(2):89-98.
[11] Tian Z, Lu W, Wang T, Ma B, Zhao Q, Zhang G. Application of a robotic telemanipulation system in stereotactic surgery. Stereotact Funct Neurosurg 2008; 86(1):54-61.
[12] Coste-Maniere E, Olender D, Kilby W, Schulz RA. Robotic whole body stereotactic radiosurgery: clinical advantages of the Cyberknife integrated system. Int J Med Robot 2005; 1(2):28-39.
[13] Ellison LM, Nguyen M, Fabrizio MD, Soh A, Permpongkosol S, Kavoussi LR. Postoperative robotic telerounding: a multicenter randomized assessment of patient outcomes and satisfaction. Arch Surg 2007; 142(12):1177-1181.
[14] Agarwal R, Levinson AW, Allaf M, Markov D, Nason A, Su LM. The RoboConsultant: telementoring and remote presence in the operating room during minimally invasive urologic surgeries using a novel mobile robotic interface. Urology 2007; 70(5):970-974.
[15] http://www.iward.eu
[16] Billard A, Robins B, Nadel J, Dautenhahn K. Building Robota, a mini-humanoid robot for the rehabilitation of children with autism. Assist Technol 2007; 19(1):37-49.
[17] Chitwood WR, Jr., Nifong LW, Chapman WH, Felger JE, Bailey BM, Ballint T et al. Robotic surgical training in an academic institution. Ann Surg 2001; 234(4):475-484.
[18] Papert S. Mindstorms children, computers, and powerful ideas. New York: Basic Books; 1980.
[19] Martin FG. Robotic explorations : a hands-on introduction to engineering. Prentice Hall; 2001.
[20] http://bricxcc.sourceforge.net.
[21] Hansen JC. Lego Mindstorms NXT Power Programming: Robotics in C. Variant Press; 2007.
[22] Bagnall B. Maximum LEGO NXT - building robots with Java brains. Winnipeg: Variant Press; 2007.
[23] Kelly JF. LEGO Mindstorms NXT-G programming guide. Apress; 2007.
[24] Hirst A, Johnson J, Petre M, Price B, Richards M. What is the best programming environment/language for teaching robotics using Lego Mindstorms? Artificial Life and Robotics 2003; 7(3):124-131.
[25] http://www.umit.at/.

Design-Build-Test: A Project Course for Engineering Students - Implementation of Assistive Functions on a Power Wheelchair

Sven Rönnbäck and Staffan Schedin

Department of Applied Physics and Electronics
Umeå University
SE-90187 Umeå, Sweden
sven.ronnback@tfe.umu.se
staffan.schedin@tfe.umu.se

***Abstract*— We describe a project within the Design-Build-Test course where a student group, based on research, implemented help functions on a power wheelchair.**
The Design-Build-Test course at Umeå University comprises both an industrial relevant student projects and non-technical exercises like project management, teamwork (team dynamics) and communication. The goal is to create a learning environment where students from different study program work together in projects, resembling the conditions for projects in the industry.
We believe that this approach will promote valuable skills in the field of product and system development which are important for the students' future role as engineers.

***Keywords*— Design-Build-Test, multidisciplinary project course, engineering education, power wheelchair, assistive robotics**

I. Introduction

In 2006, a CDIO [12] (Conceive, Design, Implement, and Operate) project course was initiated by the faculty of Technical and Natural Science at Umeå University, Sweden. Six study programs participated in this project course, of which the main goal is to implement the CDIO standards into the education. As a central part we developed a multidisciplinary Design-Build-Test project course (15 ECTS) where students from different study program participated in the same project. The course was introduced in the 3rd year for the Bachelor of Science (BSc) program students, and 4th year for the Master of Science (MSc) program students.

We wanted the students to have gained a considerable technical knowledge within their own discipline, e.g. biotechnology, computer science, engineering physics, applied electronics, or mechanical engineering, to ensure that advanced projects could be offered with an industrial company as customer.

Year 2010 the course ran at 50% study pace over an entire semester, 20 hours/week (15 ECTS), from September 2010 until mid-January 2011. One project team worked on implementing help functions on a power wheelchair. We describe the process, project work, and development of this project team, and give examples of the outcome of the project.

The examination was based on the following components:

- a written individual report analysing the project process in terms of self-evaluation and assessment of their own practical work,
- the project-group's oral evaluation of the project process,
- the final presentation of the project result,
- the level of activity during realization of the project,
- the final project report, and other documentation described in the LIPS[1] model, such as the project plan, time plan, project meeting documents etc.

II. Pre-Course Planning

The students should encounter a design-build experience to learn the process to develop new products and systems in a multidisciplinary environment [2]-[4]. To achieve this goal three key elements were identified:

- involvement of interested and devoted teachers,
- relevant multidisciplinary projects,
- successful outcomes require deep knowledge of different subjects that only can be achieved by creating project groups that involves students from different study programs.

III. Learning Outcome

In addition to train design and build experience, also objectives like formalized project planning/management, administration/documentation, personal communication and oral/written presentation, and to make an overall responsible contribution as a member of a team are all important features to be learned.

The learning outcomes of the course were the following:

- apply engineering skills and knowledge and participate in the entire developing process of a product or a system,
- plan and organize the work in a developing project,
- actively participate in a project group and understand the roles for each of the different group members,
- apply engineering reasoning and creativity,
- practice oral and written communication, both within the project group and externally,

- establish and follow a project plan for a defined project,
- evaluate the product/system with respect to a sustainable- and a commercial life cycle assessment perspective,
- present the results from a large project both in written form and orally.

IV. Contents of the Course

All together 22 students from five different study program participated in the course. The course started, for all students, with introductory lectures in project management (the LIPS project model), communication, group dynamics, and team building. In addition, specific lectures for the wheelchair, the "Wheeli-group", and the biotechnology groups, respectively, were performed to give an adequate background to the specific projects.

A. The LIPS Project Model

The projects were managed using the LIPS project model [1], which has three major phases.

1) Before Phase: During the "Before Phase", four weeks long, the commission was given and the project was planned by the project students. The projects groups received a customer-defined rather unspecific task. After discussion with the customer the task was defined in a requirement specification and a possible realization was outlined in a system drawing. Here each group writes their project plan, time plan, and activity plan, which describe the execution of the project.

2) Under Phase: In this LIPS phase the project group follows the project plan, and the time plan, to meet the requirements in the requirements specifications document.

3) After Phase: Here the project is finalized. Project reports and technical reports are written. The project results, the products are handed over to the customer and the project are closed.

V. Team Work And Team Dynamics

It is very important that a group get formed and that the student feel they belong to the group and that they work on a realistic project. It is also important that the project leader takes the role as the leader seriously, and to make the roles more clear the students sign a contract.

Since the students came from different backgrounds, in this project computer science and mechanical engineering, it was important to find the strength of each individual student within the group.

After a decision by the customer, the execution of the project was allowed to be started. At this point the so called "*During Phase*" began where the practical project work was carried out. This phase lasted for about 10 weeks and was concluded by the *system test* where the projects outcomes were demonstrated for customers and/or the industrial partners. During the "*After Phase*" the project result was transferred to the customer and at the end the project was closed. Finally, at the very end of the course, an evaluation of the project was made including both the project process and the technical outcome.

VI. The Wheeli Project

In one of the DBT projects the students worked on a robotic power wheelchair a Permobil Corpus C350, named *Wheeli*, see Fig. 1. It is differential driven and has two caster wheels in the ground. It can be seen as a vehicle with a circular 2D footprint.

Fig. 1 A Permobil Corpus C350 power wheelchair, from Permobil AB, equipped with a laser range finder (on wheelchair table), a rate gyro, and an USB interface to access the control system.

The power wheelchair is equipped with a computer interface, a laser range finder, and a rate gyro. Through the computer interface it is possible to read the joystick values, set the velocity and turn rate of the wheelchair. Through the interface it is also possible to read the current and the voltage over each motor.

VII. Before Phase – Wheeli Project

A. Project Description

Four students, out of 22, wanted to work in the Wheeli project. The students booked a meeting with the customer who presented the project for the group. The project description, a document that describes the expected outcomes, was handed over to the group. The project description was formulated as follows:

"The project aims to create a wheelchair that can be used by an individual who is paralyzed from the neck down, but also a control unit with multi-touch function, which can be used to steer the wheelchair at a distance such as by an assistant. The interface to the user may be directed by an air hose, tracking of head movements, eye tracking, or a computer mouse.

It must be possible to mount the hardware on the wheelchair, for instance on the backrest or the wheelchair table."

The task given to the project group was to implement assistive functions, and make a simple Graphic User Interface (GUI), where the user could select different supportive navigation functions on the *Wheeli*.

B. Scenarios

The customer also described scenarios to make the implementation of the help functions easier.

1) Scenario 1: A wheelchair user must be able to drive between two buildings at Umeå University, from "Teknikhuset" through a skywalk to the "MIT" building and back within 20 minutes based on GUI control and the implemented help functions. Some difficult areas involve walls of glass and iron fences. A staircase, leading down, can also be seen as a severe obstacle since it can not be detected when the sensor is mounted in horizontal position.

2) Scenario 2: A wheelchair user must, based on the GUI be able to drive the along the pedestrian walk. The pedestrian walk is around 300 meters in distance. The wheelchair must be able to follow a pedestrian walk outside semi-autonomously.

C. Project Perspective

Possible perspectives to the described project were also presented to the group:

1) User Perspective: A user who can use his wheelchair as support so that the risk of collision with objects in environment decreases. It gives a kind of freedom.

2) Manufacturer's Perspective: One can imagine that wheelchair manufacturers are happy to provide a wheelchair with help functions.

3) Assistants Perspective: Severely disabled patients often have one or two assistants. Some auxiliary functions can be of interest to them, such as the "Follow me" function.

4) Economic Perspective: A wheelchair with help functions can reduce the need for assistants.

5) Relatives Perspective: It may be so that relatives of a severe disabled wheelchair user will be pleased if a user can drive the wheelchair by self.

D. Assigned Roles in the Project

After the project description was handed over to the students, the process of forming the group started and they were assigned to set the following roles within the group:

- a project leader,
- a project member responsible for the economy,
- a project member responsible for the handling of documents within the project,
- a project member responsible for the implementation,
- a project member responsible for the hardware,
- a project member with responsibility for tests.

The project specification also stated that no group member was allowed to work more than 300 hours on the project. So it was very important that the time planning works, and continuously updated during the project.

VIII. Analysis of The Requirements Specifications

In a meeting together with the customer and the project group the wheelchair help functions were given different priority levels; high, normal, low, and removed priorities.

A contract was signed, between the group and the customer, which stated that the group must focus on deliver functions in high and normal priorities. If the group has additional time they also work on the delivery of functions with low priorities.

A. Functions with High Priority

1) Collision Avoidance: Based on streamed data from a laser range finder, SICK S300, and algorithms the wheelchair must avoid detected obstacles in the environment [5]. It must also prevent a user from hitting objects.

2) Emergency Stop Functionality: It must be possible to emergency stop the wheelchair through a button that is easy to reach for the wheelchair user. An activated emergency button must directly stop the vehicle. There must also be a way to emergency stop the wheelchair on distance through a radio link, for example WLAN.

3) Design of an User Interface: It must be possible to execute driving commands through a GUI, Graphic User Interface.

4) Graceful Motion: The wheelchair must move gracefully [6].

5) Shared Control: The power wheelchair and the user must be able to control the wheelchair [7], where the user always can override the system.

B. Help Functions with Normal Priority

1) Map Building: Pose the wheelchair on a representative map that represents of the users' environment.

2) Follow Path: Here the wheelchair follows a known path outdoors. A scenario was created for this.

C. Help Functions with Low Priority

1) Detection of Known Objects: Through a web camera the wheelchair system should recognise known objects in its environment. This to feed the navigation software with reference points, to localise the vehicle. This would require a database of objects.

D. Functions Removed from the Priority List

1) Follow Me: The power wheelchair follows a person, also in a dynamic environment where for instance many people are present,

2) Innovative Design: New ideas regarding control and use of the wheelchair,

3) Tactile Display: Information shared to the user by vibrations through small electrical motors placed on the backrest and on the seat.

E. Project Priorities

The project description also informed the students about how they should prioritise delivery time, project budget, and project result.

1) Delivery Time: it is important that the project reach the high priority help functions at the project end.

2) Project Budget: the group may be allowed to exceed the budget, if that is needed.

3) Result: the group does not need to deliver all functions for the power wheelchair.

The hardware/software handed over to the student group were the following:

- Permobil Corpus C350 power wheelchair with USB computer interface,
- a laser range finder, SICK S300 with USB interface,
- a Dell Latitude 2100 netbook computer with Windows XP and MATLAB,
- Tobii C12 Eye tablet computer,
- a rate gyro with USB interface,
- USB Interface to power wheelchair,
- a digital camera for documentation,
- a 20 channel USB GPS receiver,
- a Logitech 9000 Pro USB web camera,
- software to interface the wheelchair and Java for accessing the wheelchair, and a skeleton to the GUI[11].

E. Project Milestones

1) Milestone 0: Sept.6 2011 – The customer presents the background of the project and hands over the project charter to the students.

2) Milestone 1: October 1 2010 – Project plan and system sketch is ready.

3) Milestone 2: November 2 2010 – Project status presentation.

4) Milestone 3: January 11 2010 – Delivery of product and official project presentation with live demonstration.

Fig. 2 A laser range finder, mounted on the wheelchair table, was used for perception and detection of obstacles. A rate gyro, to the right of the hand control unit, was to calculate the heading of the power wheelchair

IX. During Phase – Wheeli Project

The customer provided some code as a starting point, MATLAB interface to the sensors, and the wheelchair control system. The student also got a simple GUI that had some of the basic functions implemented.

A. Project Budget

The students were given a budget of 12.000 SEK, approximately 1200 Euro, for the project. The money they could use to buy material to the project, and the group. They could also consult an expert for 15 hours.

B. Scheduled Meetings

1) Group Meetings: The group had weekly meetings where they discussed what needed to be done, and distributed work between the team members.

2) Meetings with the Supervisor: The meetings were scheduled on Monday afternoons, at 15:00, and approximately one hour long meetings. The project leader made an agenda for the meeting. The meetings were documented and uploaded to the Moodle platform where the group kept all their documents.

C. Halfway Presentation for the Customer

At the presentation, the supervisor acted as a customer and invited the group for an oral presentation about their advancements in the project. The presented information is about what the customer can expect and what to be delivered as well as some preliminary results. The group demonstrated some preliminary results on the wheelchair, see Fig. 3. They also stated what they will be able to deliver, and made a warning about a high priority function that they would not implement; the possibility to emergency break the vehicle remotely over a wireless link.

The group had mounted a safety switch on the right side of the power wheelchair that cut off the power to the wheelchair when it is activated.

Fig. 3 Three project students, the project leader as a wheelchair user, demonstrate some assistive functions on the power wheelchair

D. Test Protocols

The student made a test protocol to evaluate the implemented functions. The students identified things in their testing document. Some of them are listed below, and also referred to the requirement in the requirements specifications:

- The user should feel that he has control of the wheelchair (Requirement 1).
- The user should not be able to drive forward into a wall (Requirement 2).
- The wheelchair must be able to drive autonomously around a stationary object in the middle of a room (Requirement 8).
- The wheelchair should be able to pass an obstacle without collision. The wheelchair will not run into the moving objects in front of it (Requirement 9).
- People should be able to pass the wheelchair in motion at a reasonable distance without risking that the wheelchair runs into them.
- The user can choose a driving direction that the wheelchair will try to keep (Requirements 4).
- The user can control the wheelchair using the existing joystick (Requirement 15). Several test drivers will drive the wheelchair and feel they have are in control.
- The user should be able to tell the wheelchair to keep to the right or left in a corridor (Requirement 6).
- The user can drive the wheelchair via a user interface on the computer (Requirement 17). The wheelchair must be able to follow a wall to the right or left side of a corridor.
- The user should be able to turn 90 or 180 degrees to the right / left (Requirement 7).
- The user can control the wheelchair via an user interface on the computer (Requirement 17). The wheelchair must be turned 90 degrees to the specified direction.
- The wheelchair should be able to detect obstacles from floor level up to the sensor height (Requirement 10). The wheelchair must detect obstacles at a reasonable distance.
- Acceleration of the wheelchair must be graceful, without jerks (Requirement 11).
- The user should be able to steer the wheelchair with the eyes using Tobii's products CEye (Requirement 18). The user can use the graphical user interface with the eyes.
- It must always be possible to manually override the help functions (Requirement 19). The wheelchair will stop and turn off when the user presses the emergency stop button.

X. After Phase – Wheeli Project

In the after phase the students have to deliver the following items:

- final project report.
- a reflection document in which each project member describes his/her role in the project,
- an oral presentation about the outcome,
- return the borrowed and bought equipment.

A. Updated Budget

Updated budget – The student group used 6600 SEK of 12000 SEK for their project. The biggest cost was the purchase of a Acer Aspire 5471G laptop computer with a quad-core processor. The student did not have their own laptop computer, and the netbook and Tobii tablet computer provided to the group was not good for software development. The student s had access to one stationary computer in their project room, and more stationary computers in nearby computer labs.

B. The Final Project Report

The report covered the implemented functions on the vehicle.

C. The Individual Reflection Document

Each student wrote a personal document in which he/she reflected over their own role and work in the project group. It also covers the dynamics in the group and their own place in the group. In the reflection document sometimes the students describe things that are hidden from the other group members, such as workload, skills, and conflicts.

D. Final Presentation and Demonstration of Project Result

All the four members in the project group participated in the presentation.

After the oral presentation the student group demonstrated their final product.

E. Returning Borrowed Equipment

Before the student returned the equipment they checked their inventory and made an inventory list where all items were listed. In the handover they got a signed document that they had returned all the borrowed equipment.

XI. Project result

One of the employees at the university took the chance to test the Wheeli power wheelchair when the project group demonstrated the system, see Fig. 4.

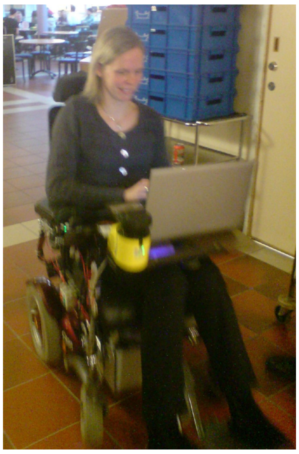


Fig. 4 An employee at Umeå University tests some of the implemented help functions when the students demonstrates the the project result, directly after the final presentation

A. *The Graphical User Interface*

The graphical user interface was designed with big buttons and also shows sensor data in real time, see Fig. 5. The current sensor data and processed sensor data, from the laser range finder, are visible in the middle of the GUI.

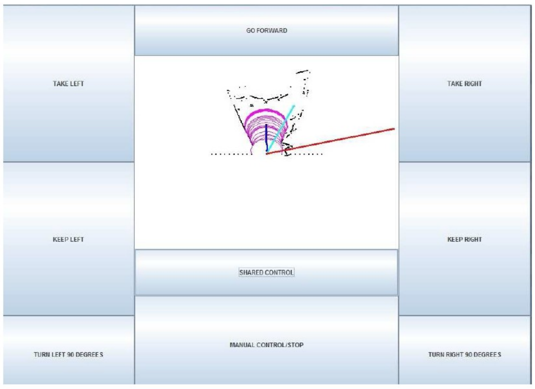


Fig. 5. The GUI with buttons for assistive wheelchair functions. Sensordata are and processed data are visible in the centre.

The GUI buttons, visible in Fig. 5, are:

- *take left / right* – drive the next exit to the left / right automatically,
- *keep left / right* – follow the left / right wall automatically,
- *turn left / right 90 degrees* – changes the wheelchair heading 90 degrees left / right,
- *go forward* – drive forward with automatic control and the systems makes sure the wheelchair drives where there is free space,
- *shared control* – the user place a destination with the wheelchair joystick and the wheelchair system makes sure the there is no collisions,
- *manual control/stop* – take manual control of the wheelchair.

XII. Results and Evaluation

The course was evaluated in several ways:

- analysed feedbacks from project students (individual reports),
- analysed feedbacks from students through the Moodle platform,
- comments from teachers and project supervisors,
- on project results.

A. *Comments and feedback from the students*

Some comments from the students:

- it was too much documents to write in the before phase,
- it worked well to work in pairs (within the group),
- Moodle[13] was used as a communication platform,
- the project meetings on Mondays were good,
- the project room was good,
- it was a special situation when the supervisor was the specialist and at the same time acted as customer,
- the LIPS project model is not suitable for programming project, and they argued that an Agile project model had better since it was a programming projects, and widely used by software companies.

B. *Comments from the Teacher*

The Wheeli group was not that satisfied as new hardware was introduced some weeks after the project started the Tobii Ceye tracker. As a teacher we could have argued that the students should call for a meeting with the customer and discuss things written in the requirements specifications document.

What could be observed by the teacher is that the student study other courses at the same time, and that therefore the time plan needs to be written seriously from the beginning. Often labs and exams in other courses collide with planned time in the project.

C. *Other Comments*

The customer also wanted the group to test out a new 3D time-of-flight camera [14], the Fotonic B70, but the group said that they had enough workload as is.

Based on the results both the customer and the group came to the conclusion that the current sensor cannot be used in a real life setting since it measures ranges in 2D. It is important that a sensor can sense obstacles on the floor and up to around 1.7 meters above the ground in front of the vehicle. In this setting, as shown in Fig. 4, it was a problem to see obstacles on floor level.

D. Course feedback from students in Moodle

At the end of the course, the students were asked to fill out a course evaluation with about 20 questions on the learning platform Moodle. The course received in general good ratings in the evaluation. One of the main questions was: "How would you assess the overall quality of this course?" The response to this question was an average rating of 4 (on a scale where 5 means very good and 1 very bad). Along with each question, the students could give comments. We noticed that the Wheeli group was not that satisfied as new hardware was introduced.

E. Evaluation of the Evaluations

The course evaluation is a very important tool for the teachers and supervisors to improve the course. Our web-based evaluations are optional, which is a problem. Although the students handed in the questionnaires anonymously, not all of them completed the evaluation. Only about 2/3 of the students made it. The results from such an evaluation may be of lower relevance than if all the students taking the course participate in the evaluation.

XIII. Conclusions

It is important that the project group get formed immediately in the beginning of the course, and that the goal is clear. The goal must not look too easy so that the student get motivated to solve the problem. It may be so that the students get more devoted if the customer pays some sort of grant on a successful delivery. Sometimes it seems like the students focus on other courses and do not work more than necessary to pass the course.

It is important that the project leader is motivated, since he is maybe the most important person in the project.

The red line was the project plan that was written at the start of the course. It needs to be stated that it a living document that sometimes needs to be updated during the project.

We also believe it is important that the students must be taught to follow the project plan and not work on things that are not stated in the contract with the customer, the requirements specifications, since the final delivered product is scored against the requirements specifications.

XIV. Discussions

Our CDIO course, Design-Build-Test (DBT), is an interesting course where students, researcher, and industry can work together on prototypes development as well as research problems.

Regarding the technical content, a comment is that the students were not that satisfied with the SICK S300 sensor since it had limited view in 3D. Of course the Kinect sensor by Microsoft is promising in this aspect; it is cheap, has fairly good range and good view angle, and has a rather high frame rate.

Acknowledgment

Thanks to Permobil AB, in Sweden, for the power wheelchair. Thanks to the Swedish company Tobii for leding out the Tobii C12 Ceye eye tracker equipment. Also thanks to the four project students for their work; Tor Sterner, Jonas Holmström, Gustav Brännström, and Tomas Mattson.

References

[1] T. Svensson and C. Krysander, *"Projektmodellen LIPS. Ver 1.3",* Linköping University, Sweden, 2007

[2] J. Malmqvist, P. W. Young, S. Hallström, J. Kuttenkeuler, T. Svensson, *"Lessons Learned From Design-Build-Test-Based Project Courses",* International Design Conference - Design 2004, Dubrovnik, May 18 - 21, 2004

[3] G. Gustafsson, *"Experiences From The Transformation Of An Engineering Education Introductory Project Design Course Into A Project Design-Build-Test Course"*, NordDesign 2004 – Product Design in Changing Environment, Tampere, Finland, 18-20 August 2004

[4] S. B. Andersson, J. Malmqvist, M. Knutson Wedel and D. R. Brodeur, *"A Systematic Approach To The Design And Implementation Of Design-Build-Test Project Courses",* International Conference On Engineering Design, ICED 05, Melbourne, August 15-18, 2005

[5] S. Rönnbäck, T. Berglund, H. Fredriksson, K. Hyyppä*,"Circle Sector Expansions for On-Line Exploration"*, Proceedings of the IEEE International Conference on Robotics and Biomimetics, Kunming, China, Dec., 2006

[6] S. Gulati, B. Kuipers, *"High Performance Control for Graceful Motion of an Intelligent Wheelchair"*, IEEE International Conference on Robotics and Automation, 2008

[7] T. Carlsson, Y. Demiris, *"Increasing Robotic Wheelchair Safety With Collaborative Control: Evidence from Secondary Task Experiments"*, IEEE International Conference on Robotics and Automation, Anchorage, Canada, 2010

[8] U. Uridailes, et. al., *"Efficiency based modulation for wheelchair driving collaborative control"*, IEEE International Conference on Robotics and Automation, Anchorage, Canada, 2010

[9] R. Geraerts, *"Planning Short Paths with Clearance using Explicit Corridors"*, IEEE International Conference on Robotics and Automation, Anchorage, Canada, 2010

[10] M. C. Ramirez Cajiao, J. A. Carvajal Diaz, Jose Tiberio Hernandez Peiialoza, *"Innovation and Teamwork Training in Undergraduate Engineering Education: A Case of Computing Engineering Course"*, The International Journal of Engineering Education, Volume 26:1, 2010, pp. 1536-

[11] S. Rönnbäck, J. Piekkari, K. Hyyppä, T. Berglund, S. Koskinen, *"A Semi-Autonomous Wheelchair Towards User Centered Design"*, Lecture Notes of the 10th International Conference on Computers Helping People with Special Needs (ICCHP'06), Linz, Austria, 2006

[12] The CDIO Initiative, [OnlineMay 2011], Available: http://cdio.org

[13] Moodle (Modular Object-Oriented Dynamic Learning Environment), [Online July 2011], Available: http://en.wikipedia.org/wiki/Moodle

[14] Time-of-flight Camera, Fotonic B70, [Online July 2011], Available: http://en.wikipedia.org/wiki/Time-of-flight_camera

Simultaneous navigation and fault detection of legged robot

Anton Vitko[#1], Ladislav Jurišica[#3], František Duchoň[#3], Andrej Babinec[#4], Martin Dekan[#5], Dušan Kaštan[#6], Marian Kľúčik[#7], Michal Bachratý[#8]

[#]Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava
Ilkovičova 3, Bratislava, Slovakia
[1]anton.vitko@stuba.sk
[2]ladislav.jurisica@stuba.sk
[3]frantisek.duchon@stuba.sk
[4]andrej.babinec@stuba.sk
[5]martin.dekan@stuba.sk
[6]kastan.dusan@gmail.com
[7]marian.klucik@stuba.sk
[8]michal.bachraty@stuba.sk

***Abstract*— The paper describes results obtained in the development of adaptive fuzzy-neural navigation subsystem for mobile legged robot. In order to keep the motion sufficiently smooth, free of sharp turnings and transversal swings when moving between closely located obstacles, the fuzzy rules are updated on line. To this end the fuzzy rules are expressed through a layered feed-forward neural network and parameters and parameters in two steps – rough and fine updating. That is followed by the description of the learning fault diagnosis using binary neural network based on the Carpenter and Grosbergs' adaptive resonance theory.**

***Keywords*— navigation, mobile robots, fuzzy – neural system**

I. Introduction

Defined as a process of reaching a distant goal location, the navigation is a primordial task for any mobile robot. But there are significant differences between the wheeled and legged robots. As usual, task of the legged robot is not to move in office-like environment or smooth roads on which car-like vehicles run, but in irregular and unstructured terrains which are found in the natural environment. This fact implies differences in many aspects including the kind of information to be processed.

Restricted locomotion capabilities of the wheeled robots are sufficient in structured scenarios in where the ground is sufficiently flat. Examples are straight corridors, right angle corners with marks on the floor, standard door appearance etc. That is why the wheeled robot navigation can do with simple contact sensors, sonar and infrared rangers and so on. Contrary to that the legged robots are expected to walk on irregular terrain.

In comparison with wheeled robots, the legged robot moving in a harsh natural terrain calls for flexible locomotion system and intelligent control system. Besides the robotic he system should be able to cope with uncertainties and unforeseen failures, which can occur in mechanical construction of the legs as well as faults or malfunctions of the sensor and communication system. Therefore our attention is focused on intelligent navigation using both soft computing learning strategies and fault identification in order to secure sufficient level of an autonomous operability.

The control community is familiar with the term of "intelligent control", denoting the abilities the conventional control system cannot attain. Leaving alone the general meaning of the concept, it would be useful to single out some basic features that could be used for characterizing an intelligent system. Intelligent control was linked with the features that were traditionally out of the scope of specialists in conventional control systems. These are mainly the abilities of making decisions, adapt to new and uncertain media, self-organization, planning, image recognition, and more. Intelligent systems should not be restricted to those that are based on a particular constituent of soft computing techniques (fuzzy logic, neural networks, genetic algorithms and probabilistic reasoning), as 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 today's systems intelligent is just a synergic use of these techniques, which in time and space invoke, optimize and fuse elementary behaviors into an overall system behaviour. For instance, fuzzy inference is a computing framework based on the fuzzy reasoning. But as to the fuzzy system is not able to learn, a neural network must provide its 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. [1]

Intelligence of neuro-fuzzy systems springs from successive generalization of the information chunks (granules) from singular ones, through crisp granular, to fuzzy granular information. An inferential process then runs over (overlapping) information granules. Due to the information granularization a system becomes robust with respect to imprecision, uncertainties, and partial truth. Thus, the system intelligence comes from the system architecture i.e. from an inner organization of the both system elements and functionalities. To demonstrate this, let us look at the *subsumption architecture*, developed in 1986 by Brooks [2] and used also in the design of navigation algorithm of our mobile robot. The subsumption architecture was inspired by the behavior of living creatures and, it is worth saying that, it heralded a fundamentally new approach to achieving more intelligent robots. In this architecture the robot behaviour is typically broken down into a set of simpler behaviours that are loosely co-ordinated towards a final goal in a sense, that every single behaviour selectively assume the control of all subsumed behaviours. The behaviours with higher priorities are subsumed under those with lower priorities; hence a layered structure is developed. The layer (i.e. a set of behaviours of the same priority) with higher priority can inhibit or even supersede those having assigned lower priorities.

II. The navigation

Within the development of the navigation algorithm it was supposed that the robot is equipped with an ultrasonic ranger which rotates and scans the environment around, providing information about the distance and azimuthal angle of the nearest obstacle. The output signals (angle and size step) control the robot to turn left or right and to modify its speed. The navigation is exclusively of reactive character. It doesn't need any environmental map.

In order to keep the motion sufficiently smooth, free of sharp turnings and transversal swings when moving between obstacles, the parameters of fuzzy rules are updated on-line. It is done periodically in two steps for each period. Within the first step takes place the tuning of rectangular membership functions (MF). To this end the fuzzy rules are updated using algorithms of unsupervised learning within which a cost function is evaluated. The cost function is chosen in such a way that its minimal value should prevent the robot from possible overthrowing due to high speed along a bend path. That conception allows us to flexibly change the radius of the curved path and thus to account for instantaneous dynamic conditions during motion (this aspect has not been included in this paper).

The fine tuning of MF's takes place within the second step. To be more specific, normally straight walls of the trapezoidal MF's are deformed into appropriate irregular shapes. This is done with the aim to reach yet smaller value of the cost function. To avoid the unnecessary reduction of the robot speed the two steps should be repeated with high frequency, which is derived from actual speed of the robot. Simulation results have showed that the described learning philosophy conception is feasible. Besides, it also prevents the robot from intensive transversal swings which are natural in a purely reactive navigation. The fuzzy system can mimic the human reasoning, and possesses human tolerance for incompleteness, uncertainty, imprecision. As a means of modelling the decision a fuzzy model, comprising 24 fuzzy rules was used. Typical structure of the fuzzy rules used is:

IF (obstacle is middle) AND (distance is near) AND (target is right) THEN (turn is right)

The premise parts are connected by AND function of the three input variables, namely:

-"obstacle", means the azimuthal angle of the nearest obstacle

-"distance", means the distance from the robot to the nearest obstacle

-"target", means the azimuthal angle of the target

Outputs of the neuro-fuzzy engine are:

-"turn"- turning angle by which the mobile platform is requested to turn in order to avoid the nearest obstacle

-"step"- size of the step to be done in requested direction. Fig.1

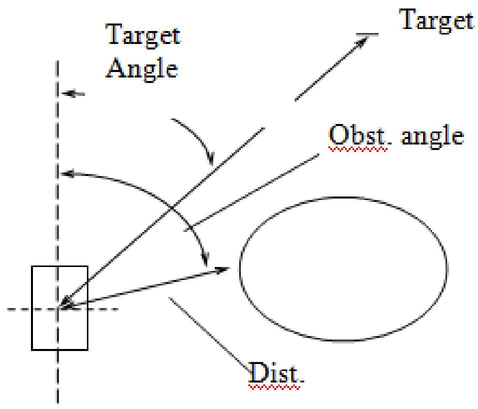


Fig. 1 Definition of input variables

The size of the step is reduced as the robot approaches either the obstacle or the target. The antecedent parts are evaluated through fuzzy reasoning which is based on Min-Max composition rule for fuzzy AND and OR operators respectively. For conversion of the fuzzy set outputs to corresponding crisp was used the bisector method .

As a measure of the closeness of the actual radius to the desired one was used the function

$$E = (r^d - r)^2 = \left(r^d - \frac{\sqrt{a^2 + c^2 + 2ac\cos\alpha}}{2\sin\alpha} \right)^2 \qquad (1)$$

where r is an actual radius of the robot's path curvature a and c are two consecutive step seizes with α being an angle between their directions. Finally r^d denotes a desired (meaning maximum allowable) radius of the robot yawing. Such arrangement allows for optimization of the radius *r* with respect to the step sizes *a* or *c* and turning angle α. The error signal for the NN output node can be computed directly. For the particular angle α^* obtained the adaptation error ε is computed in accord with by (2)

$$\varepsilon = \left(\frac{\partial E}{\partial \alpha} \right)_{\alpha^*} \qquad (2)$$

The developed navigation system was verified by both simulation and real experiment. Results of the real verification are depicted in Fig 2. Crosses represent the borders of the obstacles as identified by sonar sensor. The path of the robot movement is represented by circles.

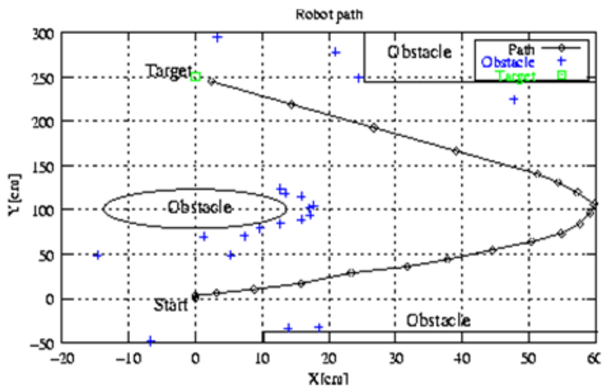


Fig. 2 The map constructed by the mobile robot

III. Neural clustering and classification (a case study)

Due to the extensive use of complex mechanical components like arms, legs, actuators, gears, clutches, grippers etc., the robot's mechanical parts suffer from significantly higher fault rates than pure electric and electronic circuitry. Potential faults should be detected sufficiently soon so as not to avoid a fatal failure. In other words, the system should anticipate possible faults on the basis of their pathologic behaviours. Therefore a novelty detection mechanism is necessary. Imminent failures are often manifested through the declined values of system parameters and variables or their fused complexes. An idea is to identify any deviation from normal behaviour. The component degradation, like wear, increased friction, stiction due to contamination, corrosion etc., is related to an observable effect on the system performance (higher vibrations, increased friction, decreased positioning precision etc). Such relationships may change as the degradation progresses.

Neural based classifiers are today the most powerful means due to their learning ability. They can classify even noisy and sparsely populated sets of measured values. In essence, practically any kind of neural network can be used for fault classification. The NN classifiers make weaker assumptions concerning the shape of statistical distribution of the input patterns in comparison with e.g Kalman filter. Another motivation is the need to detect new and unexpected faults (problem of novelty detection). This can be achieved by unsupervised learning. A serious problem with NN classification is that, in real situations, the problem domain does not always behave well. For instance, if some unexpected and strongly different input patterns appear, in the most NN there is no built-in mechanism for recognizing the novelty. Simply said, the NN should preserve previously learned patterns (stability) while keeping its ability to learn new patterns (plasticity). This phenomenon is known as *stability-plasticity dilemma.*

An elegant solution to the problem of stability-plasticity provides a family of the neural networks based on the "*adaptive resonance theory"* (ART), developed by Grossberg and Carpenter [3]. The ART family of self-organizing networks with competitive learning comprises network architectures, which are able to cluster input patterns based on a given measure of similarity. In particular, the ART1 network which was used in the experiment, allows for incremental learning of prototypes, rather then instantaneous input exemplars. This is because the whole cluster of similar inputs is updated using information from input patterns and therefore preserves main features of already accepted input patterns.

IV. Results of classification

Efficiency of the developed neural classifier was verified by simulation as well as by experimentation with the developed legged robot. The simplest and most evident faults like those related to control sequences controlling the movement of joints and legs or the faults appearing during switching between robot gaits were easily detected and classified by using the deterministic final-state machine, developed for this purpose. It was possible due to the fact that such faults manifest themselves through the total fallouts of particular sensor signals.

Contrary to this, more complex faults may be caused by increased friction in bearings, slipping or dragging clutches, lack of lubrication or a partial loss of energy delivery to a particular joint. Finally, there could be the faults caused by incorrect coordination of legs due to improper timing (fall out of phase or fall out of step and the like). Malfunctions of this kind may remain hidden for longer time and may gradually lead to fatal failures, like the total destruction of bearings or drives, lagging legs movement, which could jeopardize the walking stability or even cause instability of the robot. Such faults are commonly manifested through abnormal trajectories of the joint torques or forces. Therefore, the learning neural classifier was designed just for the task of detection and classification of any abnormal joint torques. In order to teach the neural network to classify abnormal torques, the leg dynamics were simulated in Toolbox SIMMECHANICS (a part of Simulink toolbox in MATLAB, oriented towards simulation of mechanical systems, including actuators and sensors).

The leg can be either in a stance state, when it supports the robot body or in a swing state, when it moves in air to the position where it can begin a new stance. A time-course of the normal (faultless) torque exerted in a femur joint is shown in Fig. 3. One complete step cycle is performed in three phases, each lasting one second. As seen from the figure, these three phases can be easily observed from a torque-time dependence. Particular phases are supplemented with a simple imbedded sub-figure depicting the leg configuration what corresponds to the phase. During the first phase the leg remains in a flexed configuration in the stance. The femur joint exerts the torque value about 30 Nm, which maintains an attitude of the robot body.

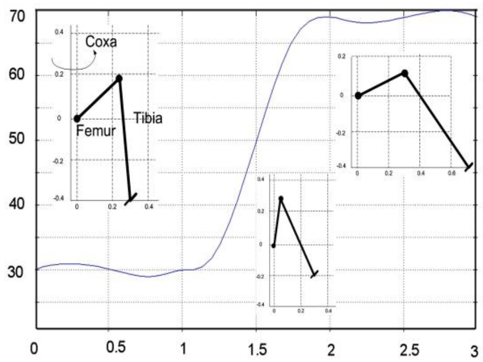


Fig. 3 Normal torque in the femur joint

The second phase starts at one second. The leg is uncoupled from the ground and starts its swing movement in a direction of walking. While the torque exerted in the femur joint causes raising the leg, the coxa joint is rotating the leg about the vertical axis and the tibia joint is extending the leg. When reaching the highest position and maximum extension the leg ends its second phase. At this time instant the femur joint exerts maximum torque of about 70Nm. Just after the third second the femur torque slightly decreases so as to make the foot go down until it reaches the ground. At this moment (at about the fourth second) the leg is entering into its stance state again, and supports the robot body.

During learning, the neural network ART1 is first taught to learn the normal torque. As a result, the neural network appoints the normal torque course as the centre of a receptive field of the cluster of all "approximately normal"torque courses (torque patterns). This is done by adaptation of the bottom-up weights leading to most left neuron in the layer F_2. From this time on the unit value of this neuron will indicated that the current input belongs to the From this time on the unit value of this neuron will indicated that the current input belongs to the cluster of "approximately normal" torque courses and this cluster will represent a class of normal torque courses. Then a training list, i.e. a series of faulty torque patterns, generated by Simmechanics Toolbox, is repeatedly presented. The experimental results have shown that the learning task could be considered accomplished (the weights reach their steady values), after presentation about 5 or 6 epochs. After learning the neural network becomes able to classify abnormal torques. Results of classification shown feasibility of the described design will be presented at the conference.

ACKNOWLEDGMENT

Support of the grant VEGA 1/0177/11 is duly acknowledged.

REFERENCES

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

[2] R. A. Brooks, *Achieving Artificial Intelligence Through Building Robots*, AI Memo 899, MIT, Art. Intelligence Lab., 1996.

[3] G. A. Carpenter and S. Grossberg, *A massively parallel Architecture for a Self-organizing Neural Pattern Recognition Machine*, Comp. Vision, Graphics and Image Proc., vol. 37, pp. 54-115, 1987.

Road Detection Using Similarity Search

Roman Stoklasa
Faculty of Informatics
Masaryk University
Brno, Czech Republic
Email: xstokla2@fi.muni.cz

Petr Matula
Faculty of Informatics
Masaryk University
Brno, Czech Republic
Email: pem@fi.muni.cz

***Abstract*—This paper concerns vision-based navigation of autonomous robots. We propose a new approach for road detection based on similarity database searches. Images from the camera are divided into regular samples and for each sample the most visually similar images are retrieved from the database. The similarity between the samples and the image database is measured in a metric space using three descriptors: edge histogram, color structure and color layout, resulting in a classification of each sample into two classes: road and non-road with a confidence measure. The performance of our approach has been evaluated with respect to a manually defined ground-truth. The approach has been successfully applied to four videos consisting of more than 1180 frames. It turned out that our approach offers very precise classification results.**

***Index Terms*—road detection, similarity search, navigation, image classification, autonomous robot, Robotour**

I. Introduction

Robotour—robotika.cz outdoor delivery challenge[1] is a Czech competition of autonomous robots navigating on park roads, the aim of which is to promote development of robots capable of transporting payloads completely autonomously in a natural environment. Development of the approach presented here have been motivated by this competition.

For a successful navigation some kind of environment perception is necessary. The perception can either be based on *non-visual* techniques, such as odometry, infrared sensors, usage of a compass and GPS signal, or based on *visual* information obtained by a camera (or several different cameras). The non-visual techniques are in general more sensitive to outdoor environment and the information content is not so rich as in the case of visual navigation. Efficient analysis of visual information is very challenging.

Two notable approaches to navigation using visual information have been used by winner teams in the previous years of Robotour competition. The basic principle of the first approach described in [1], [2] is to find a set of interesting points on the camera image [3], which represents some significant points in 3D space. It is essential to have a special "map" that contains a huge number of these points with their position in the environment. This map must be created before the navigation process itself and it is typically built during a series of supervised movements of a robot through all possible roads. All detected points are stored in a database with their estimated position. When the robot navigates autonomously in such mapped environment, interesting points are extracted from the image and compared to the points in the "map". The position and orientation of the robot is determined according to the matching points. The main disadvantage of this approach is the need of creating an ad hoc map of the whole environment where the navigation process would take place. Because building of ad hoc maps is impractical for large environments, this kind of approaches is not allowed from the year 2010 on.

The second navigation approach used by Eduro Team [4]—winner of Robotour 2010—combines a road detection with an OpenStreetMap map. For the road detection they used an algorithm based on the principle described in [5]. The idea is to track similar visual pattern that appears in the bottom of the image. It is assumed that there is a road in the bottom part of the image and everything that looks similar is also the road. This simplification brings a big disadvantage because when a robot gets to an difficult situation (for example when it arrives to an edge of the road) this method can easily be confused and start to follow a non-road visual pattern, or, vice versa, it can cause problems on the boundaries between two different road surfaces.

In this paper we address a subtopic of the whole navigation problem of autonomous robots in the natural environment based on similarity searches (Section II), which does not build any ad hoc map before the navigation. In particular we present a novel approach for road detection from the input images taken by robot's camera (Section III), which can detect roads even with different surfaces. We show (Section IV) that the proposed approach can reliably detect roads under various light and environment conditions and that it can also detect unpredictable situations not present in the training data, which could otherwise negatively influence the navigation process.

II. Similarity Search

Content-based image retrieval is a process of finding images in some image collection or database that are visually similar to the specified query image. We need to represent images using objects in some metric space in order to be able to define some (dis)similarity measure between them [6]. It is very common to use a vector space with an appropriate metric function as a metric space. In such a case, we have to represent images as vectors in this vector space.

[1] http://robotika.cz/competitions/robotour/en

Visual descriptors are used to describe some image characteristics in a form of vectors. There are many different image characteristics which can be described, for example, color properties, textures or shapes. In our case, we are using global descriptors from the MPEG-7 standard [7], namely: edge histogram, color layout and color structure. Edge histogram descriptor (EHD) is a sort of texture descriptor describing the spatial distribution of edges in the image. It produces an 80-dimensional vector and is partially invariant to image resolution. Color layout descriptor (CLD) describes spatial distribution of colors in the image and is resolution-invariant. CLD works in YCbCr color space and produces a 12-dimensional vector. Color structure descriptor (CSD) represents an image by both the color distribution of the image and the local spatial structure of the color. This color descriptor works in HMMD color space. CSD produces 64-dimensional vectors.

In general, every descriptor uses its own vector space with a different metric function due to different dimensionalities. In order to compare images according to multiple criteria, it is possible to combine multiple descriptors together using an aggregation function (e.g., a weighted sum or a product). We used weighted sum as the aggregation function for combining the dissimilarity values for each single descriptor.

There are two basic types of similarity queries: *range query* and *k-nearest neighbor (k-NN) query*. Range query $R(q, r)$ returns all images whose distance from the query image q is smaller than range r. k-nearest neighbor query k-NN(q, k) returns up to k nearest images to the query q. We use k-NN query type in our approach.

In the training phase, we store different samples of categories of interest into a database with a label (attribute) specifying their class. We use two classes: *road* and *non-road*.

Similarity search engine is implemented using MESSIF similarity search engine framework [8].

III. Road Detection

Input of our road detection algorithm are images from a robot's camera. Output of the algorithm is a *classification map*.

Classification map is an image with the same dimension as the input image, which contains for each pixel a likelihood that the pixel belongs to a particular class. In our case, this map contains 2 values for each pixel: (1) the likelihood that the pixel belongs to the *road* class and (2) the likelihood that the pixel belongs to the *non-road* class. In Fig. 1, the classification map is visualized with blue (road) and red (non-road) colors and the likelihood is represented with their brightness. The darker the color the lower the likelihood.

Our road detection algorithm can be divided into the following steps (see Fig. 1).

1) Sampling of the input image—input image is divided into suitable rectangular regions (called samples), which are processed individually
2) For each sample from the input image:
    a) Retrieve the most similar samples of known surfaces from the database using k-NN query

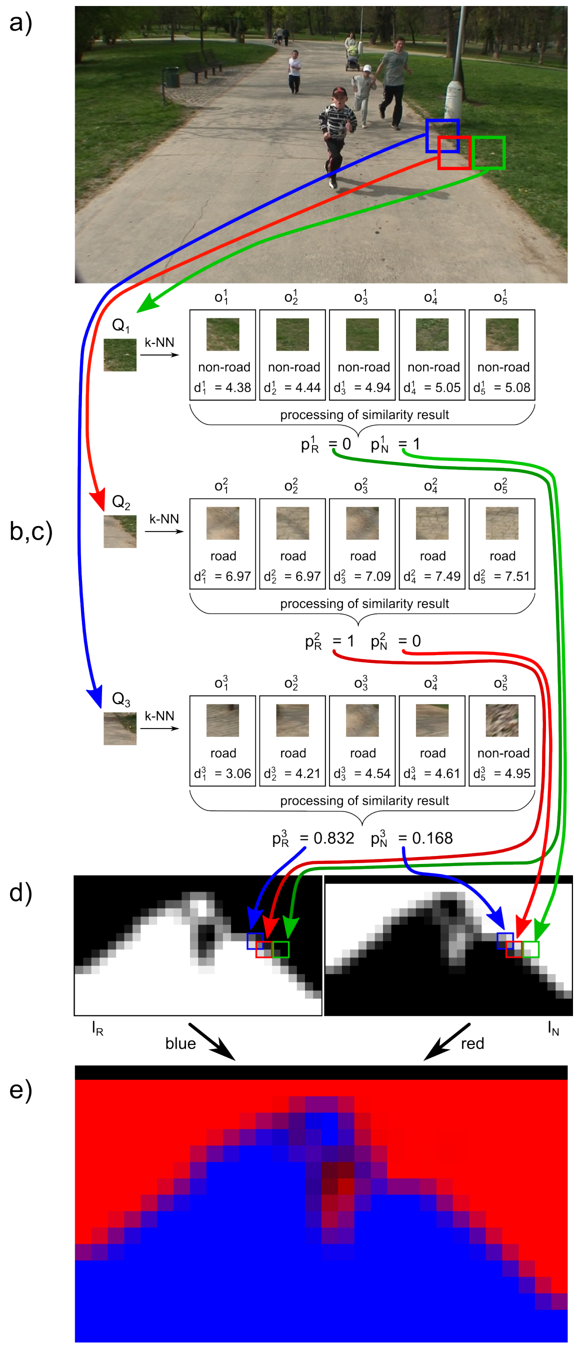


Fig. 1. Illustration of the road detection algorithm. a) Samples Q_1, Q_2 and Q_3 are extracted from the input image. b) Most similar images from database are retrieved for each sample Q_i (for $i \in 1, 2, 3$) using k-NN query ($k = 5$). c) Results from similarity database are combined together and likelihoods p_R^i and p_N^i are computed for each region Q_i. d) Values p_R^i and p_N^i are stored separately in the classification map. e) Classification map, where the likelihood that the pixel belongs to road and non-road classes is visualized with blue and red colors, respectively.

b) Process the retrieved information from the similarity database and estimate the likelihood that the sample from input image contains road or non-road
3) Combine classification result of each sample from input image and create the whole classification map

A. Sampling of input image

We divided the input images into regular rectangular regions with some overlaps. For images of size 720×576 px and 960×540 px, we used samples of size 64×64 px with an overlap of 32 px. The procedure is illustrated in Fig. 2.

With this sampling strategy it can happen that a sample contains both road and non-road areas. However, this is not a problem because the similarity search engine can return the most similar samples from the database and the similarities are combined together. In order to reduce uncertainties in the classification map we use the overlaps.

We use segmentation into regular tiles of same sizes due to strightforward implementation. The size of samples was determined empirically for our testing data set as the compromise between the resolution of classification and the computational complexity. Every single sample should contain enough characteristic visual clues with discrimination power for classification of the particular type of surface. Too small samples would not contain enough visual clues and the total amount of samples would be very high; too large samples would tend to contain more than one type of surface, which would decrease the precision of classification.

B. Similarity query and processing of similarity result

For each sample from an input image we search for k most similar samples in the database using k-NN query. Let Q_i denote i-th sample from the input image. Response of the k-NN(Q_i, k) query contains (up to) k objects $\{o^i_1, o^i_2, \ldots, o^i_k\}$. Each response object o^i_j can be written in a form of triple $o^i_j = (img^i_j, d^i_j, c^i_j)$, where img^i_j denotes the image from the database, d^i_j represents distance from the query image Q_i and c^i_j is the class to which the sample img^i_j belongs. Based on this response we determine the likelihood p^i_R that the sample Q_i contains road and the likelihood p^i_N that it contains non-road.

In order to determine likelihoods p^i_R and p^i_N we combine results of k-NN query based on the information from the search engine. Both probabilities are computed as a weighted combination of $\{c^i_1, \ldots, c^i_k\}$.

1) Weights: Let $\{w^i_1, \ldots, w^i_k\}$ denote weights for classes $\{c^i_1, \ldots, c^i_k\}$ that belongs to objects $\{o^i_1, \ldots, o^i_k\}$. We require that following properties hold:

- If an object o^i_m is λ-times closer to Q_i than an object o^i_n, then classification information c^i_m should have λ-times higher weight than c^i_n:
$$d^i_m = \frac{1}{\lambda} d^i_n \Longrightarrow w^i_m = \lambda w^i_n$$
Note that this rule is consistent also in a situation, when the distance d^i_m is equal to 0 and distance d^i_n is non-zero. In such case c^i_m will be considered as the only one relevant class information, because weight w^i_m will be infinite.
- Sum of all weights should be equal to 1 (except the special case that some of the distances d^i_j would be 0):
$$\sum_{j=1}^{k} w^i_j = 1 \tag{1}$$

Assume that we have a set $\{(d^i_1, c^i_1), \ldots, (d^i_k, c^i_k)\}$ as the input for the aggregation function. Assume that this set is ordered ascending according to the distance so that d^i_1 is the lowest distance and d^i_k is the biggest one. We define a normalizing term for the weights as:

$$N^i_w = \sum_{j=1}^{k} \frac{d^i_k}{\max(d^i_j, \epsilon)} \tag{2}$$

Because the distance d^i_j can be in general equal to 0, we need the term $\max(d^i_j, \epsilon)$ in the denominator to avoid division by zero. ϵ is some arbitrary small positive value (for example 10^{-6}). Then we can define weight w^i_j as:

$$w^i_j = \frac{1}{N^i_w} \cdot \frac{d^i_k}{\max(d^i_j, \epsilon)} \tag{3}$$

It holds, that $\sum_{j=1}^{k} w^i_j = 1$

2) Confidence factor: As we have mentioned above, we want to estimate some factor of confidence, that the similarity results are relevant. We define a function $\alpha(d)$:

$$\alpha(d) = \begin{cases} 0 & \text{for } d > 2T_d; \\ 1 & \text{for } d < T_d; \\ 1 - \frac{d - T_d}{T_d} & \text{for } T_d \leq d \leq 2T_d; \end{cases} \tag{4}$$

which define the confidence that the object class in the database with distance d from query q is relevant also for query image q itself. T_d is a threshold of "absolute confidence". If the distance between an object o and a query q is less then T_d, confidence value is equal to 1. If the distance is in the range $\langle T_d, 2T_d \rangle$ confidence value decreases linearly, and if the distance is greater than $2T_d$, the confidence is equal to 0.

3) Final likelihoods: If we define that $c^i_j = 1$ when the image img^i_j represents road and $c^i_j = 0$ when the image img^i_j represents non-road then we can compute final likelihoods p^i_R and p^i_N using:

$$p^i_R = \sum_{j \in \{x | c^i_x = 1\}} \alpha(d^i_j) \cdot w^i_j \tag{5}$$

$$p^i_N = \sum_{j \in \{x | c^i_x = 0\}} \alpha(d^i_j) \cdot w^i_j \tag{6}$$

With these definitions, numbers p^i_R and p^i_N can have a value only from interval $\langle 0, 1 \rangle$ and it must hold that $p^i_R + p^i_N \leq 1$. The inequality can happen if sample Q_i is not similar enough to any of the samples in the database. These definitions allows us to work with a confidence in similarity search results and are a key part of our approach. Note that these definitions can easily be extended to any number of classes.

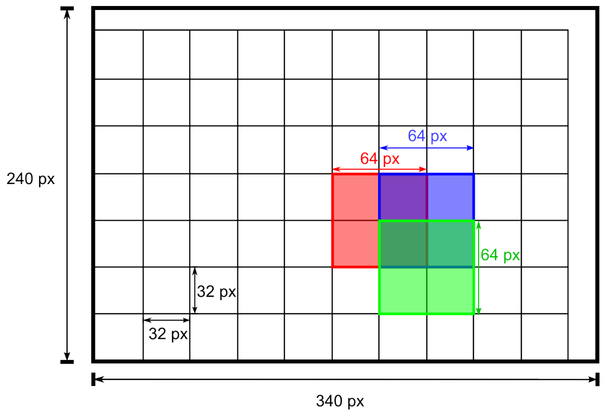


Fig. 2. Segmentation of input image into tiles for similarity search. We used samples of size 64×64 px with overlaps of 32 px.

C. Creation of classification map

From the previous step, we have a set of triples $\{(r^1, p_R^1, p_N^1), \ldots, (r^n, p_R^n, p_N^n)\}$, where r^i is i-th region (corresponding to i-th query image Q_i) and p_R^i and p_N^i are the likelihoods defined above. Because regions from $\{r^1, \ldots, r^n\}$ may overlap, we define a final classification of pixel p in the classification map as the average of classifications of all regions that contain the pixel p.

Fig. 3 shows an example of the final classification map computed by our algorithm. The value of p_R is encoded into the blue color channel, the value of p_N is encoded into the red channel. Dark areas in the image means that the algorithm was unable to reliably determine the classification of that areas, because that areas are not visually similar to any of the known samples in the database (in those areas the sum of $p_R + p_N$ is lower than 1).

IV. Evaluation and Results

A. Test data-sets

We tested our method on videos from a real outside environment recorded in a park[2] in the same way as would be recorded when the camera would be carried by an autonomous robot. The test videos were recorded on 2 different days with different light conditions.

We present results on 4 different video sequences (called "walks"). The first and the second videos (called *walk-01* and *walk-02*) were recorded using Canon XM2 camcorder on an autumn day with an overcast weather. These videos were recorded with resolution of 720×576 px. The third and the fourth videos (called *walk-03* and *walk-04*) were recorded with Sony HDR HC-3 camera on a sunny spring day. The videos were recorded in HD resolution (1920×1080 px), but we worked with downsampled images with resolution 960×540 px.

All videos together had a total length of more than 28 minutes. For the evaluation of classification precision we used 364 frames, which were picked evenly in intervals ranging from 0.8 to 8 seconds for different *walks*.

We defined ground-truth manually for each frame in the testing set. Ground-truth for each frame was created as a mask of road area in the frame. We draw the mask manually using a bitmap editor.

B. Knowledge base

Content of our knowledge base was generated semi-automatically. We picked some frames from our testing set, for which we had defined ground-truth. From these frames we extracted several samples of road and non-road regions in the following way. A computer generated several random positions of the sampling window. Each sample whose domain overlapped with road or non-road area in the ground truth for more than 93% was included into the knowledge base. The threshold of 93% was determined empirically.

We picked 53 frames from videos *walk-01* and *walk-02* and then we generated 50 samples of size 64×64 px from each frame. We have manually discarded samples that contained some image abnormality, e.g., over-exposed regions. After this processing we got 2635 samples. The size of our testing knowledge base turned out to be sufficient in our case. We did not rigorously test the minimum size of the knowledge base and did not study the relation between its size and the environment variability in which the navigation should occur.

From videos *walk-03* and *walk-04* we picked 15 and 11 frames respectively and from each frame we generated 20 samples. Using this process we obtained additional 520 samples.

Some examples of such samples stored in our knowledge base are shown in Fig. 4.

C. Precision Evaluation

We defined several error metrics in order to evaluate precision of our algorithm in a quantitative way. The amount of an error depends on the two factors: size of the area on which we obtained other than expected result; and also on the difference between expected and actual result.

We define two measures: "absolute amount of intensity under the mask" (denoted by S_A) and a "relative amount of intensity under the mask" (denoted by S_R). Both measures are evaluated with respect to the ground-truth image GT (which serve as an mask) and a gray-scale image I. Let GT image be a binary image that contains only values 0 or 1. Let I be a gray-scale image, which contains values from interval $\langle 0, 1 \rangle$. Let both images have the same dimensions over a domain Ω. Expressions $GT(p)$ and $I(p)$ denote intensity value of pixel p within the image GT and I respectively. Let the numbers w and h be the width and the height of the images. Then we can define S_A and S_R using:

$$S_A = \frac{\sum_{p \in \Omega} \min(GT(p), I(p))}{w \cdot h}$$

$$S_R = \frac{\sum_{p \in \Omega} \min(GT(p), I(p))}{\sum_{p \in \Omega} GT(p)}$$

[2]park Lužánky, Brno, Czech Republic

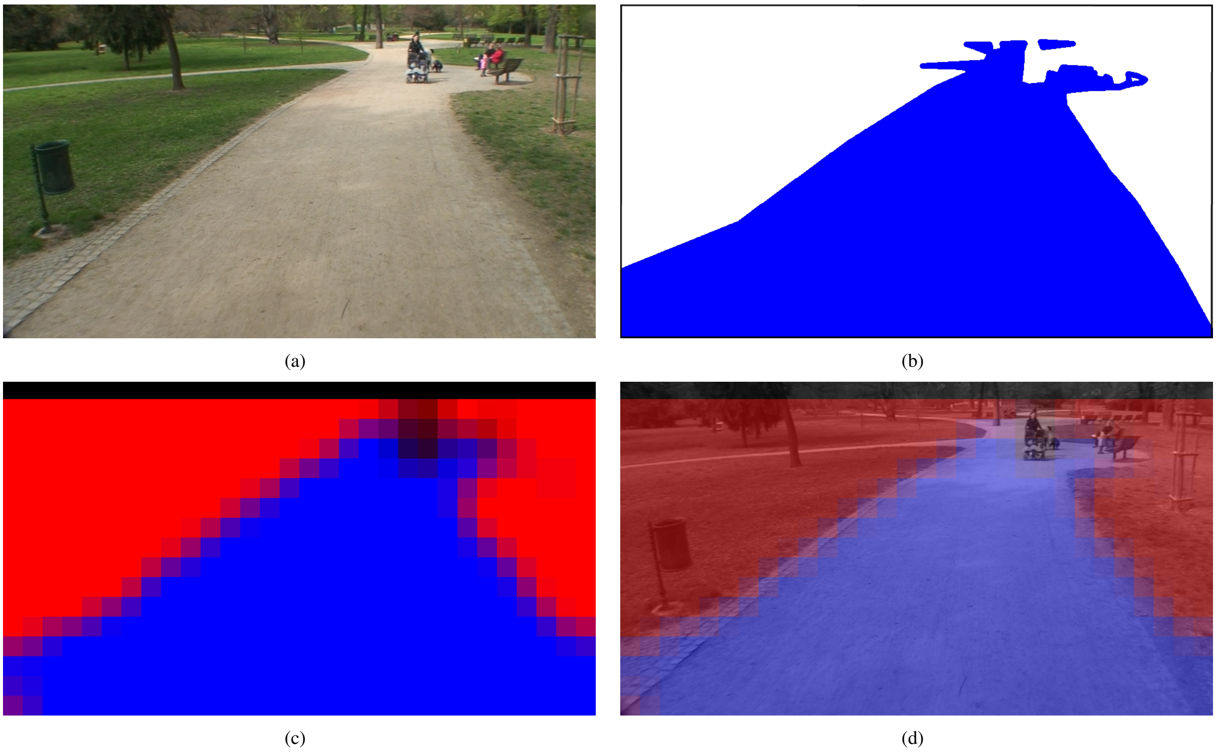


Fig. 3. (a) Input frame from the camera. (b) Manually defined ground-truth for the frame (blue area represents road). (c) Computed classification map. Notice the dark area in the upper part of the classification map—this area contains visually unknown pattern and thus the confidence of the classification is low. Topmost black bar is unclassified margin of the image. (d) Classification map overexposed over input frame.

In these equations a sum of minimal pixel values at corresponding positions in images I and GT are calculated and they are either normalized with respect to the surface of the whole image (in case of S_A) or with respect to the surface of the mask (in case of S_R). Both S_A and S_R have values from the interval $\langle 0, 1 \rangle$.

A value of S_A expresses the ratio between the sum of intensities under the mask and the maximally possible sum of intensities in the whole image; a value of S_R expresses the ratio between the sum of intensities under the mask and the maximally possible sum of intensities under the mask.

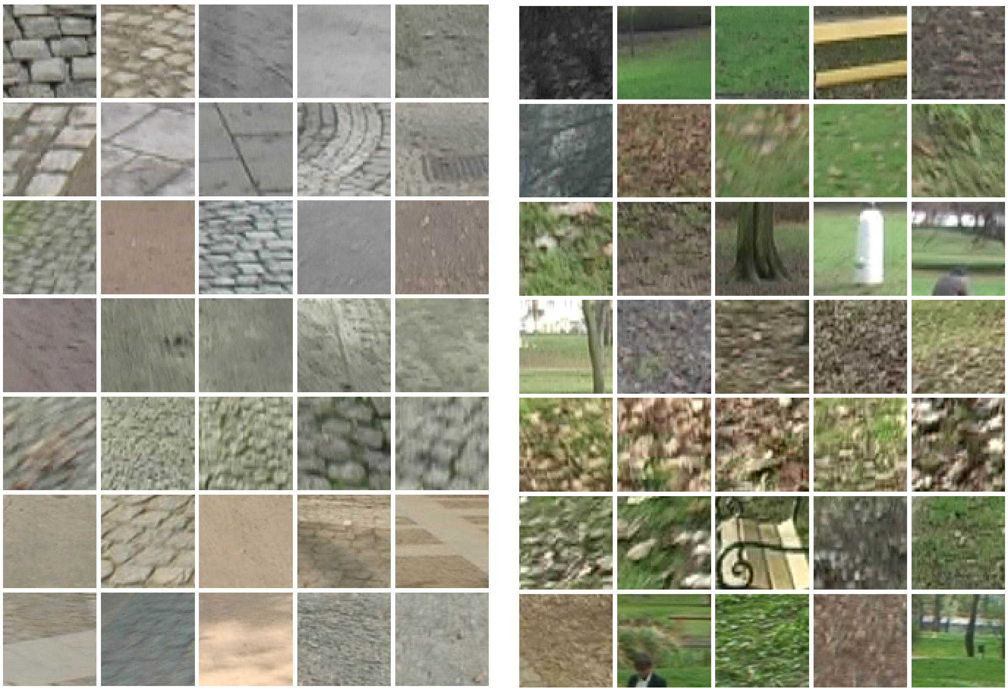
Let I denote the whole classification map encoded as image, I_B denote the blue channel of image I (which contains values of p_R), I_R denote the red channel of image I (which contains values of p_N), GT denote manually defined ground-truth, which contains value 1 for pixels, which represent road and 0 for those, which represent non-road. Let $\overline{X}$ denote complement (i.e., negative) of the image X.

We define several error metrics:

- Error of type FP (*False Positive*) – quantifies the proportion of pixels classified as road within non-road regions Defined as: $FP(I) = S_A(I_R, \overline{GT})$
- Error of type FN (*False Negative*) – quantifies the proportion of pixels classified as non-road within road regions. Defined as: $FN(I) = S_A(I_N, GT)$
- Error of type NP (*Non-Positive*) – quantifies the proportion of pixels not classified as road within road regions. Defined as: $NP(I) = S_A(\overline{I_R}, GT)$
- Error of type NN (*Non-Negative*) – quantifies the proportion of pixels not classified as non-road within non-road regions.
  Defined as: $NN(I) = S_A(\overline{I_N}, \overline{GT})$
- Precision of type PA (*Positive Accuracy*) – quantifies the proportion of pixels that were correctly classified as road regions.
  Defined as: $PA(I) = S_R(I_R, GT)$
- Precision of type NA (*Negative Accuracy*) – quantifies the proportion of pixels that were correctly classified as non-road regions.
  Defined as: $NA(I) = S_R(I_N, \overline{GT})$

D. *Error induced on the road boundary*

It is obvious that there must always be some inaccuracy caused by the used sampling strategy, where we divide the input image into the regular rectangular regions with the smallest possible resolution of 32×32 px. Because of this discretization we cannot precisely classify pixels near the border between road and non-road regions. Therefore we evaluated all error metrics also in a variant which ignores errors on the boundary between road and non-road regions.



(a) (b)

Fig. 4. Examples of images stored in knowledge base: (a) samples of road class and (b) samples of non-road class.

E. Results

Statistics of the achieved results are summarized in Table I. Values in the table are the average values of a particular error metric for all frames from a particular walk.

As seen in the table, an average error of type FP was approximately 3% (only for video *walk-01* reached almost the value of 10%). This can be interpreted in a way that 3% of the image area was classified incorrectly as a road. When we disregard an error induced on the border between road and non-road, all error metrics FP, FN, NP, NN became smaller by approx. 2.5%. Thus, when we ignore errors on the borders borders, we can say that our classification method failed to correctly classify regions in less than 1% of image surface.

Because we allow "unknown" classification in our approach (Section III-B), we also evaluated, how often this "uncertainty" happens. The amount of the "unknown" classification in the road and non-road areas can be computed as the difference NP−FN and NN−FP respectively. We can see from the Table I that this difference is mostly less then 0.5%.

It is also seen, that our road detection method was able to detect more than 85% of road area in the input images, therefore we think it should be possible to easily navigate robot through the real roads based on the result obtained from our algorithm.

Table II shows the results achieved for *walk-03* and *walk-04* which were classified using "knowledge base" based only on samples from *walk-01* and *walk-02*. As we can see that the

TABLE I
EVALUATION OF ERROR METRICS FOR ALL FOUR WALKS. VARIANT I SHOWS ERROR FOR THE WHOLE IMAGE, VARIANT II SHOWS ERROR WITHOUT THE ERROR ON THE BORDER BETWEEN ROAD AND NON-ROAD. ALL VALUES ARE AVERAGE VALUE OF PARTICULAR ERROR METRIC FOR ALL FRAMES OF THAT PARTICULAR WALK.

Walk	walk-01		walk-02	
Number of frames	76		82	
Variant	I	II	I	II
FP	9.86%	5.33%	2.95%	0.39%
FN	1.86%	0.34%	3.19%	0.36%
NP	1.91%	0.34%	3.20%	0.37%
NN	10.08%	5.48%	2.97%	0.40%
PA	93.90%	95.45%	85.46%	90.69%
NA	76.30%	85.42%	93.41%	99.16%
Walk	walk-03		walk-04	
Number of frames	98		108	
Variant	I	II	I	II
FP	2.57%	0.38%	3.19%	0.71%
FN	3.50%	0.28%	0.71%	0.26%
NP	3.68%	0.29%	2.99%	0.30%
NN	3.07%	0.57%	3.83%	1.06%
PA	92.47%	99.42%	93.98%	99.23%
NA	93.86%	98.75%	91.70%	97.18%

TABLE II

EVALUATION OF ERROR METRICS FOR CLASSIFICATION OF *walk-03* AND *walk-04* USING DATABASE GENERATED FROM *walk-01* AND *walk-02*. ALL VALUES ARE AVERAGE VALUE OF PARTICULAR ERROR METRIC FOR ALL FRAMES OF THAT PARTICULAR WALK. THESE RESULTS SHOW THAT THE DATABASE OF SAMPLES CAN BE "PORTABLE" (I.E. IT IS NOT BOUND TO THE PARTICULAR ENVIRONMENT AND THE PARTICULAR CAMERA).

Walk	walk-03		walk-04	
Number of frames	98		108	
Variant	I	II	I	II
FP	1.99%	0.30%	2.76%	0.54%
FN	4.88%	0.90%	3.45%	0.57%
NP	5.11%	0.92%	3.72%	0.68%
NN	2.61%	0.56%	3.48%	0.96%
PA	89.86%	98.00%	92.48%	98.29%
NA	95.10%	98.85%	92.40%	97.45%

results are still very precise. This shows that our method works well also for images that have not been used for building the database and which were taken by a different camera on a day with different weather conditions.

In Fig. 5, there are shown examples of computed classification maps related to the input images. Classification map is encoded as red-blue image and is superimposed over the corresponding frame from the camera for a better illustration. Fig. 6 shows some examples with obstackles, which were correctly classified as non-road.

F. Final remarks

We did not have to introduce any complex preprocessing steps before road detection because the fully automatic modes adjusting exposure time, color balance, etc., that we have used on the camcorders (Canon XM2 and Sony HDR HC-3) worked sufficiently well. Many low-end cameras would not be able to deal with these tasks and their produced images could be degraded in some way. In such case, additional image preprocessing may be necessary to achieve comparable results.

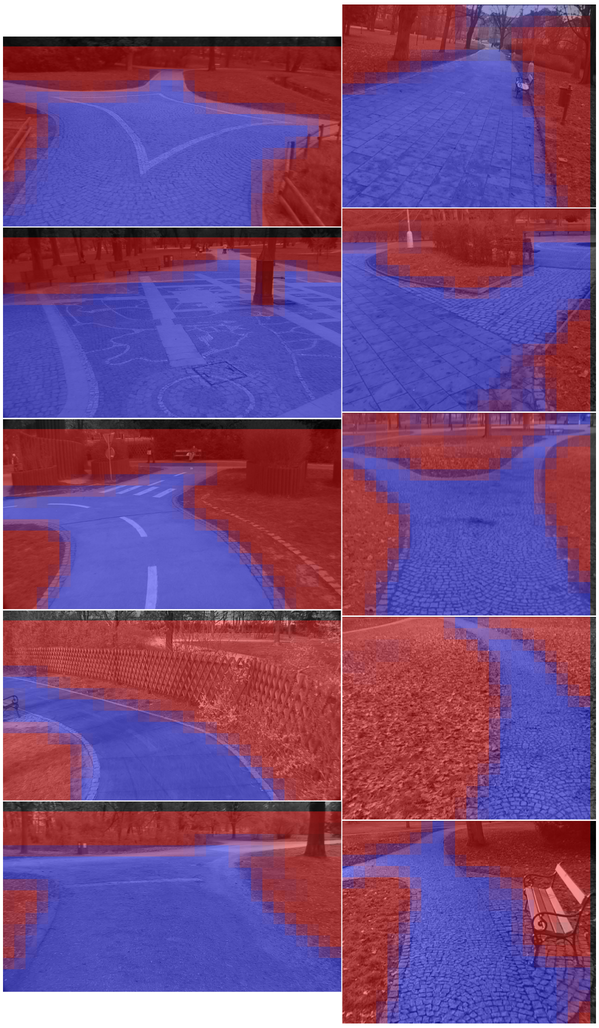


Fig. 5. Examples of classified frames exported from all videos. Frames in the left column are from *walk-03* and *walk-04*, frames in the right column are from *walk-01* and *walk-02*

V. CONCLUSIONS AND FUTURE WORK

We have proposed a new method of road detection for robot navigation in natural environment. We have tested it on real data sets recorded with two different cameras under different conditions. The obtained results indicate that our algorithm could be applicable also for a real robotic implementation. Error in surface classification is less than 1% in average and the average classification error of the whole frame from camera is less then 5%.

The most computationally intensive part of the algorithm is the processing of all samples from input images and searching for visually similar images for each of them. One can easily see, that processing of one sample is independent of the others, so all samples can be processed in parallel.

This method can be easily extended to recognize multiple classes of surfaces.

REFERENCES

[1] T. Krajník, J. Faigl, M. Vonsek, V. Kulich, K. Košnar, and L. Přeučil, "Simple yet stable bearing-only navigation," *J. Field Robot.*, 2010.

[2] T. Krajník and L. Přeučil, "A Simple Visual Navigation System with Convergence Property," in *Proceedings of Workshop 2008*. Praha: Czech Technical University in Prague, 2008, pp. –.

[3] J. Šváb, T. Krajník, J. Faigl, and L. Přeučil, "FPGA-based Speeded Up Robust Features," in *2009 IEEE International Conference on Technologies for Practical Robot Applications*. Boston: IEEE, 2009, pp. 35–41.

[4] J. Iša, T. Roubíček, and J. Roubíček, "Eduro Team," in *Proceedings of the 1st Slovak-Austrian International Conference on Robotics in Education*, Bratislava, 2010, pp. 21–24.

[5] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimson, "Visually-guided obstacle avoidance in unstructured environments," in *IEEE Conference on Intelligent Robots and Systems*, 1997, pp. 373–379.

[6] P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[7] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content*

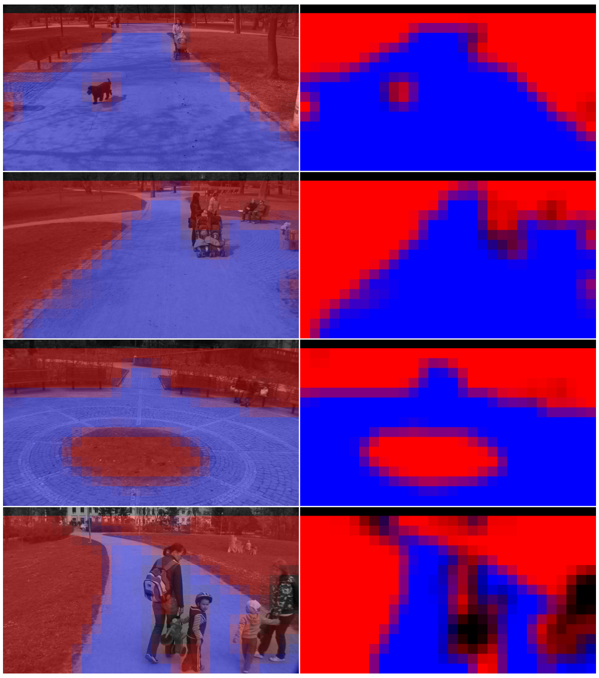


Fig. 6. Examples of classified frames which contain some obstacles. Left column shows frames with classification overlays, right column contains pure classification maps.

Description Interface, B. Manjunath, Ed. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[8] M. Batko, D. Novak, and P. Zezula, "Messif: Metric similarity search implementation framework," in *Digital Libraries: Research and Development*, ser. Lecture Notes in Computer Science, C. Thanos, F. Borri, and L. Candela, Eds. Springer Berlin / Heidelberg, 2007, vol. 4877, pp. 1–10.

Human Machine Interaction using Head Pose Estimation

Simon-Alexander Zerawa, Stefan Kohlhauser, Charlotte Roesener, Andreas Perner

Institute of Computer Technology, Vienna University of Technology, Austria

[zerawa, kohlhauser, roesener, perner]@ict.tuwien.ac.at

***Abstract*— In course of the educational research project XINU (eXcellent Interface for Non-haptic Use) high school students, university students, teachers and researchers developed a novel way to interact with and control elevators. Using standard web cams and state-of-the-art head pose estimation, distinctive and predefined head gestures are interpreted as explicit commands for a mechanical prototyped elevator model. For usability tests of the HMI (human machine interface), a cursor-based application was created to allow a mouse-like selection of possible commands by only moving the head. In this article, the system approach and the project setting is described and the current implementation is presented.**

***Keywords*— head pose estimation, user interface, elevator control, gesture, webcam, cursor;**

I. Introduction

To initiate partnerships between senior high schools and universities (and other research institutions), the Austrian Federal Ministry for Science and Research (BMWF) introduced a research program called "Sparkling Science" in 2007 and following years [1]. The basic idea is that young people shall not only get direct contact to up-to-date research but also be actively involved in research projects. Groups of both domains, research and education, participate in socially founded and cutting-edge technological developments partly situated in the learning environment of higher education. Based on this idea, the 2-year-project XINU (eXcellent Interface for Non-haptic Use) was started in October 2009. The project is conducted within a partnership between the Institute of Computer Technology at the Vienna University of Technology and the School Centre Ungargasse where physically challenged (physically, visually or acoustically impaired and others) and physically fit students are provided with economical or technical education [2][3].

The amount of features offered by today's applications steadily grows but the flexibility of their user interfaces usually does not keep pace. As our society more and more depends on the services of our half or fully automated living environment the usability of systems in our daily life is challenged. In this context, the objective of XINU is to develop a novel method of contactless operation – in this project – of an elevator by using head gestures only. As there is no need to touch/push any buttons or levers for controlling a system, this concept can be beneficial in further scenarios where physical interaction is inconvenient, dangerous or requires unnecessary physical effort the operator lacks. The research presented, offers a new and flexible way to control such systems, focussing on an elevator application and using head gestures as additional mode of user interaction. Introduced in [4] the approach used can not only be applied to elevators but several other applications of the building automation domain as well. Although the flexible model of the system allows alternative types of interfaces as discussed in [5], the authors will focus on visual head pose estimation for purpose of this article.

This article will look at the project from both sides, the educational side highlighting the project approach and the cooperation, and the research side looking at the concepts, implementations and the current types of interaction supported. The authors will conclude with discussing current results and future work.

II. Project Context and State of the Art

A major question in modern society is how to relate and to embrace the next generation which is now educated in schools of today, so that they might form a cultural identity with a technological advanced society that requires a continuous advancement and potential of future scientists. Science Education – a special form of science communication – is an emerging area of research, originating increasing activities e.g. case studies and investigations, that shall allow crossing borders where the students' life-world culture move into the world of science filling gaps between the students worldview and the worldview embraced by the scientific community [6]. Science education shall help with informal settings, shall assist to create a "public awareness and understanding of science" in a positive way. There are three aspects, which shall be emphasized: The understanding of science content, understanding the methods of enquiry and the understanding of science as a social enterprise [7].

The novel idea of the "Sparkling Science" program [1] shall enforce the integration of cutting edge research into educational science that allows the early contact of young people with interesting research projects helping them to be more interested in and better understand their (science based) world around them engaging in the discourse of and about cutting edge science. It shall reduce reservations and other barriers between these domains. This cooperation supports a creative atmosphere, where high school students and university researchers can learn from each other. But scientists shall not only talk about their science and fulfil an educational part in this setting: The fresh minds of young people give researchers the opportunity to get new input and angle of vision. The beauty of this project in focus is that students have the power to test and suggest optimization of the prototyping system. Their participation can help their bodily challenged peers and might prove to be beneficial to the usability aspects of this project.

Growing computational power and optimized algorithms allow creating systems that are able to track movements of humans with high accuracy. The developments of such systems find the way into various application domains. Building automation offers a wide area to install head pose estimation systems for user interaction. Although concepts exist to alleviate the usage of building automation systems, a focus on the assistance of handicapped people is still missing. Project XINU uses head pose estimation to offer new possibilities in exactly this area of research.

Head pose estimation systems are used to help human system interaction for robotic applications [8], [9]. By use of head pose estimation the control-system of the robot is enabled to estimate human gestures or movements in order to react to the human. The systems used to detect the movement of the head can be categorized in approaches which process a whole image and approaches which focus on distinct points [10], [11]. In order to allow a real time processing of the camera-input the second mentioned approach has advantages for control applications like in project XINU. Other approaches like the usage of markers which are placed on the users head are also used in research projects but are not applicable for the project described since not every user can be equipped with identifier points on their heads.

For the usage in real-world environments where light conditions and backgrounds cannot be precisely defined, the use of thermal images proved to be an applicable solution. Another advantage of this approach is the low computational overhead, like outlined in [12].

For systems which cannot be combined with thermal cameras but must work under restricted observable areas the work of [13] offers a promising solution. By the usage of a multi-camera system, partly observable spots can still be used for image processing algorithms.

In case of project XINU the movements of the head are important since based on the movements control commands are interpreted and sent to the elevator. As described in [14] movements of the head can be described with six degree of freedom. This allows the definition of an individual command set for each user in order to respond to certain needs of a specific user.

III. Project Development and Approach

One of the project's major goals is the creation of a prototype system. Therefore, two intervening approaches were established: first, an application based approach targeting the implementation of a model application with a following evaluation phase was conducted. Second, a broader approach focuses on creating a multipurpose platform in order to allow the integration of different user interfaces with multiple applications.

A. Prototype setup

In order to give access to this system for a wide range of different users using all kind of their own equipment, standard off-the-shelf components like notebooks and webcams were chosen to create the actual hardware for the user interface. At the Institute of Computer Technology a physical 1:20 model of an elevator was constructed providing a fieldbus based sensory network and control system using industrial sensors and actuators.

As an incentive, students of the School Centre Ungargasse were directly involved in the early state of solution design of the user interface and the following evaluation steps. To ensure communication and interdisciplinary working atmosphere, several workshops and regular meetings were held in which students and researcher could interact, exchange and evaluate ideas and challenges in a coequal way.

As several methods were evaluated, a commercial head pose estimation product was selected for development (see section IV) and the implementation of an early prototype providing basic functionality.

B. Flexible and distributed system layout

Beside efforts to build and test a prototype of the system, a broader application of the underlying concept was performed. Key components were identified to allow decoupling of the systems components (see IV.C). In a first stage this can be achieved by using client-server network or fieldbus architecture (see [4]). This allows logical separation of the actual user interaction and the specific application. Due to a modular design with standardized interfaces, the potential systems can provide various types of interaction (e.g. buttons, head pose estimation, speech recognition) on one hand and support all forms of applications (e.g. multiple elevators, air conditioning, heater) on the other. Within this precise approach, the detailed concept provides three layers (see [5]):

- A Human Computer Interaction (HCI) layer implementing the actual interface for the user,
- an application layer providing the services of the application,
- and a communication layer ensuring a proper communication between HCI and application layer.

All layers communicate their data via defined interfaces and protocols. Therefore, a distribution on different hardware systems and even mobile devices is possible. To allow the support of multiple user interfaces and different applications at the same time, a registration and administration server enables the storage and management of different user configuration sets and application-specific software. The server is able to provide an initial communication infrastructure and handles the process of establishing a communication between a specific user (with a specific method for user interaction) and the services of a specific application.

IV. System Design and Implementation

The project's objective is the implementation of a system, which provides physically challenged people with additional means to control an elevator without the need of physically strength and mobility required for pressing a button on control panels. There are various ways to achieve this objective. The focus of this project is the human computer interaction via head movement recognition. However, the design and implementation of a face tracking software itself is not in the scope of this project. Based on an extensive requirements analysis for application and user specific demands, a suitable product (API) had to be found which can provide the following criteria:

- **Robust:** The product has to handle various types of faces coping with variations in skin colour, facial deformation, beards, glasses, hairdo etc. Furthermore, occlusions or bad lighting conditions, which are common in real elevators, have to be handled well.
- **Real time:** The product has to be able to convert the movement of a head into coordinates which can then be used to control a graphic user interface (GUI). This conversion has to be done without any delays, in order to provide a smooth user experience. To be able to acquire and detect head movements a frame rate close to or above 30 frames per second is desired.
- **Simple:** Since price is one of the main issues when considering barrier-free services, low cost is an important point to pierce any resistance. This means the system should run on a standard personal computer and use a webcam coping with simple products and existing infrastructure.

Several commercial and non-commercial products have been considered. The one finally selected was the program faceAPI from Seeing Machines. It provides fast and reliable face tracking, makes no difference between varying head shapes, size, and facial features, runs with standard Intel processors supporting standard webcams and offers a well-documented application programming interface (API). It is implemented in C++ and the provided library can easily be integrated and used with other projects. The program provides tracking values of a head along the X-, Y- and Z-axis as well as rotation along these axes and an additional confidence value, which gives an estimation of how accurate the system rates these values. A commercial and a non-commercial version of the faceAPI exist. A licence was acquired, however, the functionality of the non-commercial version proved to be sufficient.

A. Gesture implementation

In the first step, the system was implemented in C++ as a local application on a single computer. It included the face tracking module, the GUI and a simple elevator simulator. It is still used as a test bed to control the mechanical model elevator instead of the software simulator during integration tests. A more detailed description can be found in [5]. This first implementation used head gestures for the human computer interaction. Nodding the head upwards or downwards selects the desired floor and nodding the head to the left commits the command to the system for execution. Using solely three different head gestures, this approach depended strongly on the knowledge of the gestures and the ability to perform them. Gestures/command sets can also be customized and trained on individual users for a mobile setup, where every user has a control unit. However, this command setup is hardly feasible in an environment where the camera is fixed and directly embedded in the elevator.

B. Cursor-based implementation

The second implementation developed considers several new approaches for improvement of user interaction. Predefined commands should be avoided and a new form of feedback was intended, namely the controlling of a cursor on a GUI. The first enhancement was to not recognize complete head gestures as specific commands but instead to interpret the coordinates provided by the faceAPI as cursor position. In this approach, the movement of the head was converted into input for the cursor which was moved over a GUI for the elevator. In this scenario the user had always a direct visual feedback and the process was much more intuitive as with the implementation described before. Therefore there was no need to agree on and learn predefined head movements as with the gesture based solution above. However, this entailed the necessity of an additional display for feedback needed for the GUI designed for the elevator.

In the GUI rectangular buttons represent the different floors for selection and other control commands of the elevator control. Functions can be selected by moving the circular cursor over the button and holding it there for about two seconds. This process is visualized by small sectors increasingly surrounding the floor number while hovering over a button (see Fig. 1). The size and transparency of the cursor represent the distance from the camera to the head and the confidence of the head pose estimation process. Once a floor button is selected the rectangular area gets highlighted until the elevator has served the floor.

An additional objective of the project is to place a mobile solution at the physically impaired student's disposal. Therefore, it should be possible to use a mobile device like a laptop, notepad, tablet PC or a smart phone for command input. The logical modules of the system were separated in order to be able to run them on distributed devices.

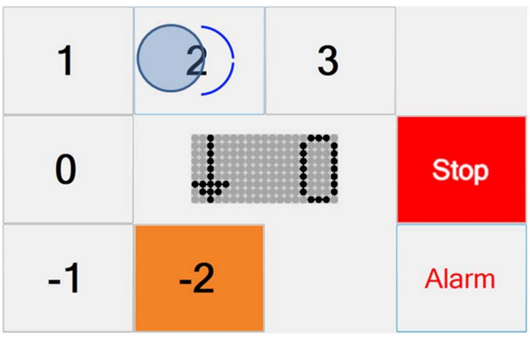


Fig. 1 GUI of the cursor based implementation

The first logical module represents the head pose estimation itself and is running on a PC with a webcam and directly uses the faceAPI framework. This module converts the face movements from the video input from the camera into coordinates, parses them into an XML data structure and provides them over the network.

The second module is the GUI server. It receives the XML-coordinates from the first module and transfers them into cursor movements, considering display size and smoothing algorithms. These cursor movements are then used to control the GUI, as mentioned above. The GUI itself can either be displayed on a screen within an elevator or it can likewise be displayed on the same device (e.g. PC) where the first module is running. By decoupling the modules, either solution is possible.

The third functional module is either the elevator simulator or the mechanical elevator prototype. When a button is selected on the GUI, the command is sent to the elevator which executes the command and sends an updated status back to the GUI. The modules and their interactions are displayed in Fig. 2. Based on this conceptual changes as mentioned above extensive refactoring of the existing code would have been necessary. This led to the idea of a complete redesign and implementation of the GUI server and the simulator as the software part of the model elevator. This step provided several additional possibilities.

The new implementation was done in C# in general. The only module not ported to C# was the actual faceAPI interface itself. For this module only a socket was added to send the coordinates in an XML data structure via the network. Any device can be used as an input for the GUI, independently of its implementation as long as it is able to send coordinates in an XML structure over a TCP (Transmission Control Protocol) socket.

Nonetheless, the GUI module, the simulator and the interface for the model elevator were ported to C#. The first advantage was that it is easier, less error prone and thus faster to write a program in C#. As a corollary it is easier to write reusable code for planed future additions.

Another major benefit of using C# was that it provided the possibility to integrate the high school students better into the actual development process: Most of them were already familiar with the C# programming language, whereas only a few had reasonable knowledge about C++. This lowers the barrier for the students to get in contact not only with the software itself but also with its source code.

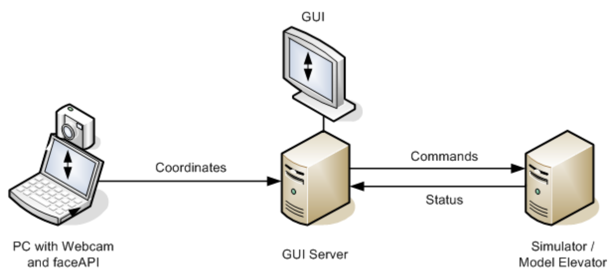


Fig. 2 The XINU modules

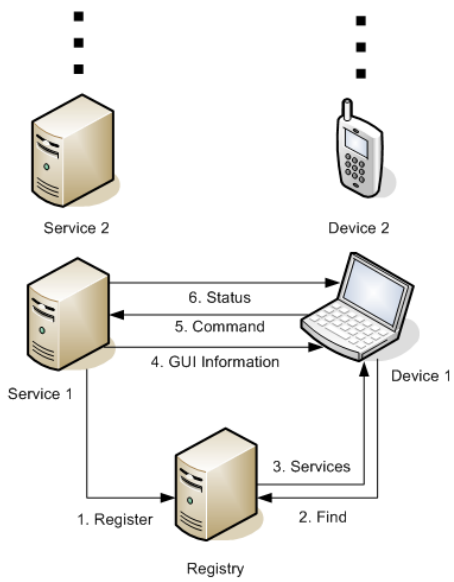


Fig. 3 Dynamic system design

This is important if the students encounter errors and do active bug tracking during evaluation. If the students have a reasonable knowledge about the programming language they can possibly provide in-depth bug reports with a precise error allocation within the source code or even might be able to fix the error themselves entailing a better insight and vast better integration of the high school students in the software engineering process.

In the current implementation the interpretation of the module output as cursor coordinates allows the controlling of any GUI. In the previous implementation only a fixed set of gestures for commands could be defined.

C. Generic service and interface implementation

This section provides an outlook for this project. As already mentioned, one of the major goals of the project is prototyping for an interaction framework. The prototype shall be able to control a wide range of services from a wide range of devices.

How the second part can be achieved was outlined already in the previous section. The goal was the usage of any mobile device, like laptops or smart phones, as an input device for the interaction with the control system. With a modular design and explicit interfaces this can be achieved. E.g. several platforms like Apple's iPhone or Google's Android have head tracking features in development. However, for this approach a novel way how to control a wide range of services has to be reconsidered. An important factor is the fact that several services have specific and diverse interaction possibilities and therefore require a special interface design.

This approach is similar to Service Oriented Architectures (SOA): Several independent services are offered in a network. They send an XML message with their IP-Address (Internet Protocol) and the service name to a registry server via broadcast. When a device joins the network, it can send a broadcast to find the registry server and get a list of available services. The options are inserted into a GUI framework which basically provides only a GUI stub. This stub is modified by the information received from the registry. Once a service is selected, the device can request for a specific service of the GUI from the service server. Again, the GUI stub is modified according to the information provided by the service adding buttons, labels, etc. as needed. By pressing a button a command is submitted to the service where the according action is performed and an updated status can be returned to the device.

The setup and interactions of the approach is shown in Fig. 3. This basic architecture will be implemented by senior researchers and students at the university. As soon as the interfaces for the services are defined, extra functionalities for the system can then easily be implemented. An example would be a web interface so the services can be used by any standard browser.

V. Discussion And Results

In October 2010 a first model setup was presented allowing the operation of the mechanical model elevator using head movements only. Following this stage, students could work directly on implementation tasks during their internships at the institute in 2010. While the crucial implementation tasks were mainly accomplished by researchers and students at the university, the high school students had the chance to use the software and carry out surveys while conducting tests with other students in school addressing different classes and physically challenged students. In several meetings, the students presented their findings to the project team. As the first prototype was set up at the institute and a prove-of-concept was operational, the mechanical elevator model was separated from the head pose estimation control itself. For evaluation and testing purposes preconfigured laptops were given to the students. Containing an elevator software simulation and several versions of webcam control software, evaluations and tests could be performed directly at the school. These were accompanied with surveys to get direct feedback from the participants, mainly peers of the same school who volunteered for testing.

The results were presented and discussed with researchers in regular meetings. It showed that once the users were familiar with the predefined head gestures the time needed to make a selection (a few seconds) was perceived appropriate by most of the participants. Additionally we found that the choice of gestures used in the selection process is very crucial and need to be configurable to cope with the needs and preferences of the individual user.

To provide better visual feedback and a more intuitive control possibility a mouse-like interface could be introduced to the hardware prototype in May 2011. This setup was presented to undergraduate students as part of a lecture at the university. Additional testing with students showed that this allowed reducing the time to make a selection between three and four seconds (including the time needed to hover over the selection).

The first mobile setup was introduced in June 2011 consisting of a Wi-Fi enabled laptop with integrated webcam. Within the range of the wireless-infrastructure, a user could select between two different applications available (the hardware model elevator and an elevator simulation). The selection of the application itself and the related control functions was performed entirely by head movements.

Throughout the project the students of the participating school were generally very eager and motivated to find a new solution to help their physically challenged colleagues with help of this project. It proved to be very beneficial conducting regular meetings both at the university and the school in order to discuss the project status and distribute tasks. The meetings helped the high school students to see their own work within a larger perspective and increase accountability for their tasks. The authors and project members also noticed a very application based mindset focussing at very specific solutions, unfortunately sometimes unrelated to the project goals itself (e.g. voice recognition), but nonetheless emerging interesting ideas. Here it was important to point out methodology and the focus of the project but still welcome and support input from all sides.

With assistance of motivated teachers, it was even possible to integrate parts of the project work in the regular classes at the school and a positive effect on motivation and productivity of the students within those periods was noticeable.

During the project's period 4-week internships at the university could be provided to motivated students during school holidays, in 2010. The interns were given specific programming and testing tasks related to the project. Concluding their assignments they presented their work to the scientific staff.

VI. Future Work

Based on good results and positive feedback from the first group of internships it was to initiate a second period of internships staring in July 2011. Following the main system implementation, smaller project tasks for the interns will be provided. To make sure the task are appropriate, the project members intend to offer topics related to HTML (Hypertext Markup Language) or ASP.NET (Active Server Pages for .NET) and other technologies that are already familiar to the high school students.

This adds to the philosophy of the XINU project that the general scientific and engineering work is accomplished the university. However, smaller tasks with a shorter time frame and well defined goals can be handed over to the high school students. These tasks extend their existing knowledge about techniques used in praxis and leave enough space for them to learn practical skills and deepen their knowledge. Some very time consuming tasks, e.g. extensive testing, give reason to be outsourced to students, who did not participate directly in the implementation and are less prone to following logical traps during evaluation. Additionally, researchers from the university enjoy the contact and fresh ideas of students, who help them to get a new view of their own research.

With regards to research content, the evaluation and development of a mobile solution will be continued. It is intended to provide a web interface to allow operation of the elevator model from any device capable to process internet webpages.

VII. CONCLUSION

The authors presented an educational project using state-of-the-art head pose estimation to control an elevator. Together, high school students, university students, teachers and researchers developed a prototype application consisting of a mechanical elevator and a PC with webcam-based control software. Two possible implementations were presented. First, defined head gestures can be used to trigger elevator control commands without pressing buttons. Second, a mouse-like selection of the floor is possible by only moving the head to position a cursor over a corresponding button on a screen.

ACKNOWLEDGMENT

This study is supported by the research program "Sparkling Science" by the Austrian Federal Ministry for Science and Research (BMWF).

Special thanks to Seeing Machines Inc. for their great support and interest in our research work.

REFERENCES

[1] Internet reference: Sparkling Science. http://www.sparklingscience.at, May 2011

[2] Internet reference: Institute of Computer Technology, Vienna University of Technology. http://www.ict.tuwien.ac.at, May 2011

[3] Internet reference: School Centre Ungargasse. http://www.szu.at, May 2011

[4] C. Roesener, A. Perner, S. Zerawa, and S. Hutter, "Interface for non-haptic control in automation," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, 2010, pp. 961 –966.

[5] S.-A. Zerawa, A. Perner, and C. Roesener, "Non haptic interaction system," in *20th International Symposium on Industrial Electronics, (In Press)*, 2011.

[6] G. S. Aikenhead, "Students' ease in crossing cultural borders into school science," *Science Education*, vol. 85, pp. 180–188, 2001.

[7] T. W. Burns, D. J. O'Connor, and S. M. Stocklmayer, "Science Communication: A Contemporary Definition," *Public Understanding of Science*, vol. 12, no. 2, pp. 183–202, 2003.

[8] M. Baklouti, M. Bruin, V. Guitteny, and E. Monacelli, "A human-machine interface for assistive exoskeleton based on face analysis," in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS EMBS International Conference on*, 2008, pp. 913 –918.

[9] M. Baklouti, S. Couvet, and E. Monacelli, "Intelligent camera interface (ICI): A challenging HMI for disabled people," in *Advances in Computer-Human Interaction, 2008 First International Conference on*, 2008, pp. 21 –25.

[10] V. Graveleau, N. Mirekov, and G. Nikishkov, "A head-controlled user interface," in *Proceedings of the 8th Int. Conf. on Humans and Computers, HC 2005*. Aizu-Wakamatsu, 2005, pp. 306–311.

[11] M. N. Mamatha and S. Ramachandran, "Automatic eyewinks interpretation system using face orientation recognition for human-machine interface," in *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 5, 2009, pp. 155–163.

[12] X. Yu, W. K. Chua, L. Dong, K. E. Hoe, and L. Li, "Head pose estimation in thermal images for human and robot interaction," in *Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on*, vol. 2, May 2010, pp. 698 –701.

[13] M. Voit and R. Stiefelhagen, "A system for probabilistic joint 3D head tracking and pose estimation in low-resolution, multi-view environments," in *Proceedings of the 7th International Conference on Computer Vision Systems: Computer Vision Systems*, ser. ICVS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 415–424.

[14] Y. Matsumoto, N. Sasao, T. Suenaga, and T. Ogasawara, "3D model-based 6-DOF head tracking by a single camera for human-robot interaction," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 3194 –3199.

Mappino - Open Source Robot for Learning about IR Range Scan Matching

Albert Diosi
This paper is the result of independent, self-sponsored research
Samorin
Slovakia
Email: albert.diosi@gmail.com

***Abstract*—Localization and mapping is a basic building block for many mobile robots. Scan matching is one way of implementing such a functionality. Currently a significant commitment of time and financial resources are likely to be necessary to delve into this area of robotics. This paper aims to bring scan matching closer to robotics enthusiasts and students by introducing an inexpensive, open source robot programmable through the accessible Arduino programming environment. In the presented robot, rotating infra red range sensors are used instead of a laser scanner. Simple example code is available for the robot to facilitate a quicker familiarization process. Preliminary experimental results demonstrate the scanning and scan matching capability of the robot.**

I. INTRODUCTION

In the future, mobile robots will likely play an important role in people's lives. Beside being useful, mobile robots can be fun to build and exciting to experiment with. Building and experimentation may encourage youth in a playful manner to learn about programming, maths, physics and engineering in general.

For decades, robotics enthusiast have been building mobile robots. However, their efforts often stopped before creating truly useful ones. Perhaps this has been caused by the lack of mapping and localization capability. It is often easier to solve a practical problem if the robot knows its pose and has access to a map of its environment. As an example let us consider a plant watering robot. If one assumes that the pots holding the plants are stationary (a reasonable assumption to make if the robot has to work only when the owners are on holiday) the problem of making the robot to know where to water can be reduced to the problem of knowing where the robot is and where the pots/plants are.

Perhaps one of the reasons why localization and mapping did not catch on in the circles of robotics enthusiasts is due to the large initial commitment required in time and often in finances to complete a mobile robot capable of self pose awareness. Furthermore the complexities of the maths involved may scare people away from implementing a localization and mapping functionality.

In recent years, several off-the-shelf solutions have appeared for localization. If the modification of the environment is acceptable, then one can use for example the NorthStar system developed by Evolution Robotics[1]. Here a stationary unit projects a pattern of light on the ceiling (if there is one) which is observed by an optical sensor unit mounted on a mobile robot. From the the light pattern observation, the unit tells the robot its estimated pose. A similar localization device called StarGazer has been developed by the company Hagisonic[2]. However, instead of pattern projection, fiducials are placed on the ceiling (if there is one). The markers are then observed by a camera unit (including an illumination source) placed on the robot. Based on the observed appearance of the fiducials, the camera unit tells the robot its pose estimate. Unfortunately none of the mentioned approaches are inexpensive to buy, open source (do not facilitate learning and experimentation to a great extent) or provide information about obstacles and both require the modification of the robot's environment.

Popular solutions to the localization problem whithout requiring the modification of the environment include the use of sonars, cameras (2D and 3D) and laser range finders. Simple sonar systems are inexpensive, however due to wide beamwidth, specular reflections and cross-talk they are not straightforward to use for localization and mapping. An example of open source sonar localization (based on an existing map) can be found in CMU's Carmen robot software package[3].

Cameras seem to be newcomers compared to lasers and sonars, yet they gained rapid popularity in the area of localization and mapping. Cameras can be inexpensive, however they require a considerable amount of processing power for utilization. Furthermore, learning about how to use cameras for localization and mapping may demand a considerable amount of effort from the buddying roboticist. Luckily there are readily available open source software packages as Willow Garage's Robot Operating System (ROS)[4] implementing such localization functionality for stereo and 3D cameras. Just as Carmen, ROS is processing power hungry and the familiarization with ROS may take a considerable amount of effort.

Laser range finders provide a fairly simple and reliable way of localization and mapping. Their price used to be prohibitive for robotics enthusiasts and for classroom use, however as shown in [1], it is possible to build a $30 bill of material (BOM) laser range finder. The mentioned laser

[1]www.evolution.com/products/northstar (Accessed: 28 May 2011.)

[2]www.hagisonic.com (Accessed: 28 May 2011.)
[3]carmen.sourceforge.net (Accessed: 28 May 2011.)
[4]wwww.ros.org (Accessed: 28 May 2011.)

range finder later appeared in the robotic vacuum cleaner of Neato Robotics[5]. Laser scan matching is a popular way of performing localization and mapping. Software modules for performing scan matching are present in both Carmen and ROS. However existing laser scan matching approaches can be processor intensive, and fairly complex.

The goal of the work behind this paper is to develop a robot system minimalistic in size, cost and complexity for educational use, capable of performing localization and mapping in simple environments while running on a hard, flat surface. Initial results aimed at achieving this goal have been obtained by constructing a simple but limited laser range finder replacement embedded into a simple robot base. All processing on the robot is done using boards which can be programmed using the simple and popular Arduino integrated development environment (IDE)[6]. A laser scan matching algorithm, the Polar Scan Matching (PSM) [2] has been adapted to run on the utilized inexpensive 8-bit microcontroller. The software and hardware designs are open source which facilitates easy experimentation and modification. The electronic components of the robot have been designed with re-usability in mind for other projects.

The concept of the range scanner comprising of rotated IR range sensors (PSD - position sensitive device) is not new. There are many examples for it in the literature for example [3]. These sensor have even been used for Simultaneous Localization and Mapping (SLAM) for example through the observation of line segments as in [4].

The contributions of this paper are two fold: an open source platform for learning about localization and mapping for robotics enthusiast and students in high school or university classrooms. There is also an engineering contribution as there does not seem to be a precedence in the literature for performing scan matching on a small, 8 bit microcontroller. All the design files are available under a suitable open source license for free.[7]

The paper is structured as follows. First the robot platform is described followed by a brief description of the scan matching algorithm. Experimental results are shown next followed by discussion and conclusions.

II. Hardware Platform

A. Requirements

Prior to designing the robot the following lax requirements were set solely based on practical considerations of the author:

- Robot should enable easy learning about scan matching, mapping and localization.
- BOM should be below 150 Euros when building a single robot.
- The robot should be able to take 2D scans of simple environments while having controlled lighting conditions.

[5]wwww.neatorobotics.com (Accessed: 28 May 2011.)
[6]wwww.arduino.cc/en/Main/Software (Accessed: 28 May 2011.)
[7]www.diosirobotics.net/mappino.html

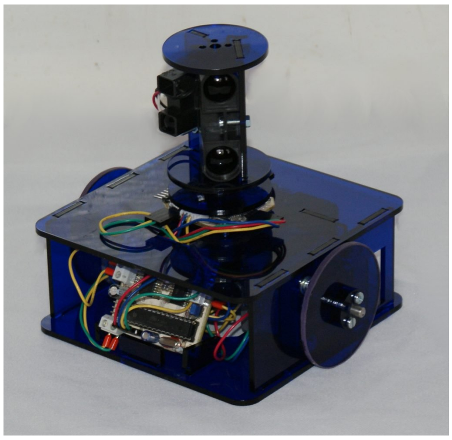


Fig. 1. Mappino v0.1.

TABLE I
Mappino Specifications

Size (WxDxH)	168x144x148mm
Weight	1100g
Motors	3 x Bipolar Stepper
Sensors	Sharp PSDs: GP2Y0A02YK, GP2Y0A710K
Batteries	6xAA 2450mAh NiMh
Estimated Runtime	$\geq$20h on Standby, $\geq$2h while moving
Maximum Speed	$\geq$10cm/s
Estimated Cost	220 euros

- Robot should be able to move indoors on flat hard floors and table tops.
- Robot run time should be longer than the duration of a double class (i.e. 2 hours).
- Minimum speed should be at least 10cm/s.
- Robot should be easily programmable, without learning much about microcontrollers.
- Robot should come with basic demos which provide a starting point for students.
- Individual robot parts should be modular, reusable and modifiable.
- Robot should be small and light for easy transportability.
- Robot should use easily available batteries.
- Electronics parts should be easily obtainable using a popular component distribution company such as Farnell.
- Mechanical parts should be easily obtainable by rapid prototyping services such as Ponoko.
- All software used in the design and for programming the robot should be freely obtainable, preferably open source. This is to ensure that users easily can modify the design.
- Robot should be programmable from three of the major operating systems (Linux/Mac/Windows).

B. The Robot Mechanics

The implemented solution (see fig. 1,2) consist of a robot base into which a rotating range finder (see fig. 3) module is embedded. The components of Mappino can be seen in fig. 4. The specifications of Mappino are shown in tab. I.

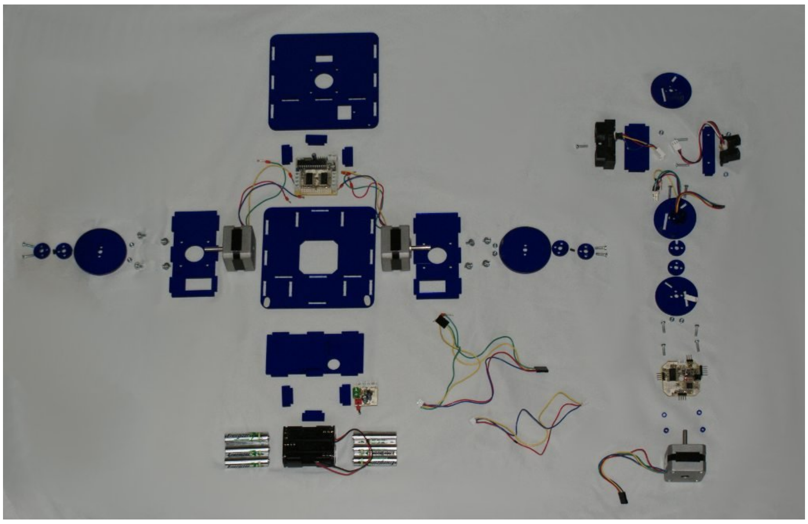


Fig. 4. Most of Mappino's components laid out as they are assembled.

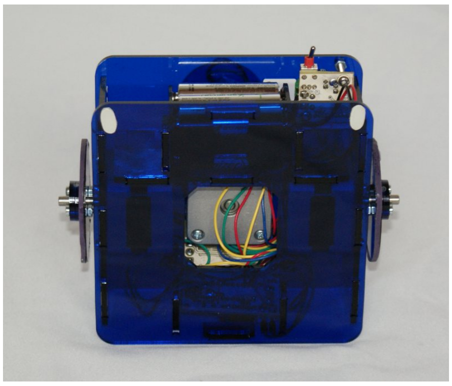


Fig. 2. Mappino viewed from underneath. Notice the power board and the mouse skids.

Fig. 3. The range scanning module.

Most mechanical parts except nuts, bolts and motors are laser cut from 3mm thick blue acrylic. Through careful design, most parts just snap together. However if extra strength is necessary one can apply glue as well. To improve traction, rubber bands are glued onto the wheels. This may be changed to O-rings in the future to ease assembly. The back of the robot is weighted down by the 6 AA batteries used as a power source. Two mouse skids mounted on the bottom of the robot under the batteries ensure easy gliding on smooth surfaces.

The robot is driven by two bipolar stepper motors micro-stepped to 800 steps per revolution. Stepper motors were chosen to simplify the control of the robot and to increase the odometry accuracy.

In the odometry center of the robot a pair of rotating range sensors is mounted. The range sensors are rotated by the same type of stepper motors as used for propelling the robot. As range sensors a long range (100-550cm) Sharp distance sensor (PSD) is used in conjunction with a shorter range (20-150cm) PSD. The shorter range PSD is offset by 45°. The purpose of the shorter range PSD is to provide distance measurements to objects closer than the minimum range of the long range sensor (100cm).

To reduce cost and to simplify the design the robot does not have a bump sensor. Instead the range sensors are to be used to detect close obstacles. However, users may add their own bump sensors if necessary.

The laser cut acrylic used in the robot providing the structural elements may be replaced in the next revision with a less brittle material such as Lexan or laser cut aluminium to increase the life expectancy of the base. Given the considerable weight of the robot due to the 3 stepper motors and the batteries, the robot can not be handled roughly without risking breaking the acrylic.

The next version of the robot may entail a reduction of size. Grub screws are also considered to enable solid mounting of the wheels and the sensor head to the motor shafts. Currently, frequent mounting and removing of the wheels and sensor head may loosen the tight fit of the plastics and cause slipping. There are other minor tweaks considered as extra holes for mounting, cabling and for making some of the screws easier

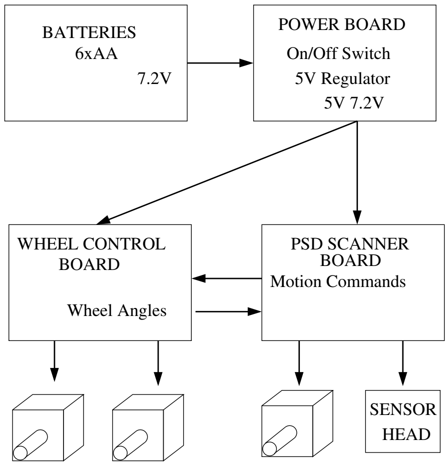


Fig. 5. Overview of the electronics. The wheel control and scanner boards can be programmed through the Arduino IDE.

to access with a screw driver.

C. Electronics

The electrical connection of the robot can be seen in fig. 5. As already mentioned, the robot is powered by 6 AA, preferably NiMh rechargeable batteries. Power from the batteries is lead to a power distribution board which hosts a resettable fuse, an on/off switch and a 5V regulator. The battery voltage and the regulated 5V voltages are connected to the dual bipolar stepper motor control board and the PSD scanner board.

The motor control board accepts serial line commands from the PSD scanner board and reports back motor orientations. The board can micro-step two bipolar stepper motors. The current into the motor windings can be set by software.

The PSD scanner electronics is mounted on top of the stepper motor, just under the sensing head. The board controls the panning stepper motor, evaluates a position reference signal and samples the analog signals provided by the PSDs. It is also the job of the PSD scanner board to generate the scans, perform scan matching and to generate the desired commands for the motion control board.

To enable energy consumption control, motors and sensors can be turned off by software. The current consumption while having the motors and sensors turned off is 60mA. By turning on the sensors and the panning motor, this current goes up to 200mA. In motion, the current consumption increases to 680mA. The current consumption is the highest with 870mA when the traction motors are activated but stationary. In the demo code supplied, motors and sensors are turned off when not in use, thus one can have the robot sitting on a desk in the ”on” state for tens of hours when using 2450mA capacity batteries. In motion, the batteries will hopefully last for more than two hours even at the end of their life cycle.

In the spirit of a reusable/modifiable design, the PSD scanner can be used as a standalone unit, independent of the robot base. Furthermore the sensing head can be easily replaced with other sensors. The motor controller and the motors can be be used on other projects as well. There are prototyping areas on the motor control and PSD scanner boards to allow adding extra components.

TABLE II
APPROXIMATE COST BREAKDOWN IN EUROS.

Acrylic	30
PCBs	54
Motors	63
Sensors	55
Batteries	13
Other	5
Total	220

As the parts have been designed using freely available software as QCad[8] for the mechanics and GEDA’s gSchem[9] and PCB for the electronics, users can customize the design of the robot. The author used Linux throughout all of the design and programming process, however there are reports of the tools working on other operating systems as well.

D. Assembly

Given pre-made boards and wiring harness, the robot can be assembled in few tens of minutes. However, making the wiring harness and assembling the boards requires skills and special tools. To reduce size and cost, surface mount parts were used.

E. Programming Environment

Until recently the programming of 8 bit microcontrollers using freely available software was often not an easy task. However, all has changed with the arrival of the Arduino boards and their programming environment (based on Wiring[10]). Arduino has enabled people with non-technical backgrounds to rapidly create embedded system prototypes. The programming language of the Arduino boards is a well documented, easy to use, simplified version of C/C++. There is also a vibrant and eager community of Arduino users all over the world. The only disadvantage is that the Arduino IDE does not support on-board debugging. Due to the virtues of the Arduino concept, the motion control board and the PSD scanner board can be programmed through the Arduino IDE and application programming interface (API).

F. Cost

When constructing only one robot, the BOM cost including value added tax and delivery fees is approximately 220 euros as can be seen in tab. II. The target price of 150 euros has been grossly overshot, however a significant portion of the cost are delivery fees. Furthermore the price includes the manufacture of 10 bare boards of each of the 3 boards. Educational institutions of some countries would not have to pay tax on the parts which considerably reduces the cost.

[8]www.qcad.org (Accessed: 1st of Aug. 2011.)
[9]www.gpleda.org (Accessed: 1st of Aug. 2011.)
[10]wiring.org.co (Accessed: 1st of June 2011.)

III. Scan Matching

Scan matching is a tool enabling localization and map building. 2D scans contain range measurements in a plane at given bearings. In scan matching an alignment of a current scan is sought which maximizes the overlap with a reference scan. The result of this alignment process can be used for localization as it provides the relative pose of the laser scanner when the current scan was taken with respect to the reference scan's pose. Even though scan matching is usually applied to data from laser range finders, it will be shown that this technique can be applied to PSD scans as well.

Scan matching has been around for a few decades starting with the work of [5]. However, in this paper a simplified version of the Polar Scan Matching (PSM) [2] approach is used as it is fairly simple and quick. However, there will be simplifications made which will preclude the reliable use in general indoor environments. The simplified scan matching will still demonstrate the principles behind laser scan matching in simple controlled environments.

Without delving into much detail for PSM (for more see [2]), the scan pre-processing steps involved for the 8-bit version are the following:

1) The PSD sensors are sampled over 267 steps of the full 360 degrees revolution of the sensing head.
2) The long range and short range measurements are combined together into one scan.
3) A median filter of width 3 is applied to the measurements to remove spurious readings.
4) The 267 readings sized scan is interpolated into a 256 readings scan. Having scans with 256 elements grossly simplifies the programming effort and processor load for matching.

During scan matching the following steps are executed:

1) A virtual scan is taken from the current scan as if the scanner was placed where the reference scan was taken. This step is called scan projection.
2) The projected scan's orientation is sought by searching for a rotation which minimizes the sum of absolute range difference between the projected scan and the reference scan.
3) The projected scan rotated by the estimated orientation.
4) The position of the current scan is estimated by minimizing the sum of square range residuals between the projected and reference scan's range readings. It is assumed that range readings in the projected scan describe the same planar points as the reference scan points at the same bearings.
5) Jump to 1) unless the changes are small, or a maximum number of steps have been achieved.

To enable a reasonable run time for scan matching on an 8-bit microcontroller with a very limited resources, the following compromises were made:

- Range readings are represented with 8-bits.
- Angles are represented with 8-bits.
- The unit range is 2cm.

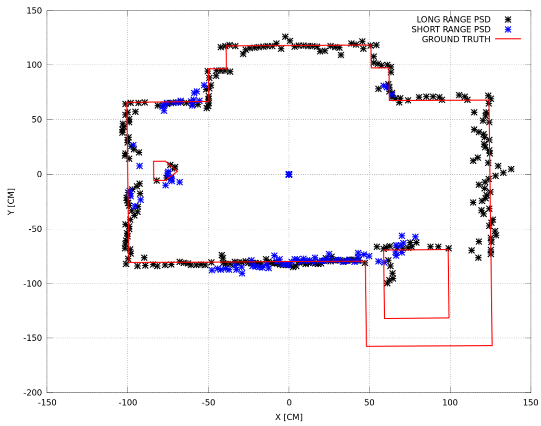


Fig. 6. A raw scan of a bathroom overlaid on the hand measured drawing (red) of the room. Points measured with the long range PSD are shown with black. Short range PSD points are shown with blue. Grid size is 50x50cm.

- Sine and Cosine look-up tables are 256 element large with 8bit resolution.
- In scan projection all readings are assumed to be good ones (bad ones are replaced with interpolated ones).

IV. Experimental Results

To show the basic capabilities of the robot, three experimental results are shown. In the first one a scan is compared to ground truth measurements, followed by a scan matching experiment. At last a series of scans are shown as collected by the moving robot. As the robot is still a work in progress, all the results are preliminary.

A. Scanning

In this experiment a raw scan (fig. 6) was taken by Mappino in a bathroom. The scan is compared to the ground truth outline of the room. The long and short range PSD voltage readings were downloaded into a PC, aligned, converted into distances and are shown separately.

Most surfaces in the bathroom were white tiles and reflected light back well. There were parts of the room which were only measured with just one of the range finders. All objects were within the maximum range of the long range PSD. The time needed to take the 360 degree scan was 4s. There were an additional 4 seconds used for determining the 0 orientation of the scanner.

The ideal surfaces resulted in a fairly good match of the scan with the ground truth.

B. Scan Matching

In this experiment on-board scan matching is demonstrated using scans taken in the same room as in the previous experiment. The robot was put down on the floor where it took a scan and stored it in its EEPROM. Then the robot moved a few tens of centimeters and took a second scan which

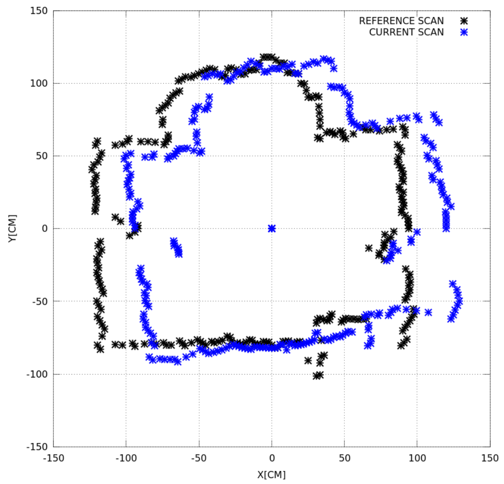


Fig. 7. Reference (black) and current (blue) scans prior matching. Grid size is 50x50cm.

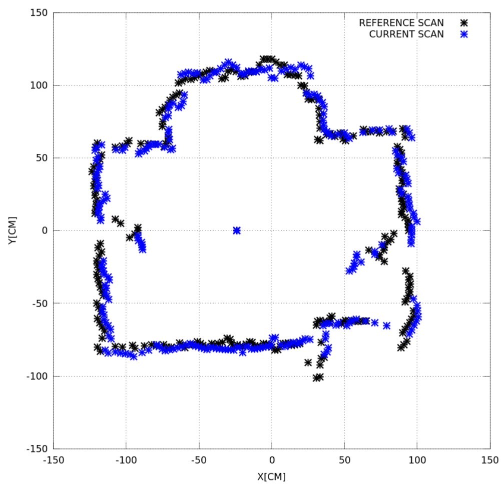


Fig. 8. Reference (black) and current (blue) scans aligned by the microcontroller performing scan matching. Grid size is 50x50cm.

was saved too. After matching these two scans, the scans and results were uploaded to a laptop through a serial line.

Scan matching was initialized (see fig. 7) with the initial pose of (0cm,0cm,0°) of the current scan. The current scan's pose converged in two steps to (-24cm,0cm,-3°) as can be seen in fig. 9. The five iterations performed took just 154ms to calculate on the microcontroller even though the compilation of the code was optimized for size and not for speed. By visually observing the resulting overlap of the scans (fig. 8) one can only see a small angular misalignment. This is not surprising as angles are quantized at 1.4° increments.

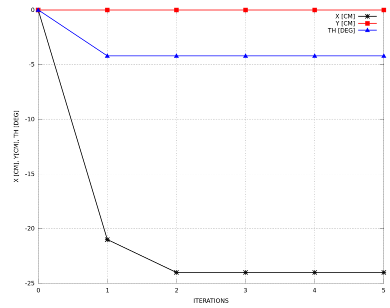


Fig. 9. The evolution of pose while matching the scan shown in fig. 7. X coordinate is shown with black, Y with red and orientation with blue. Grid size is 1x5.

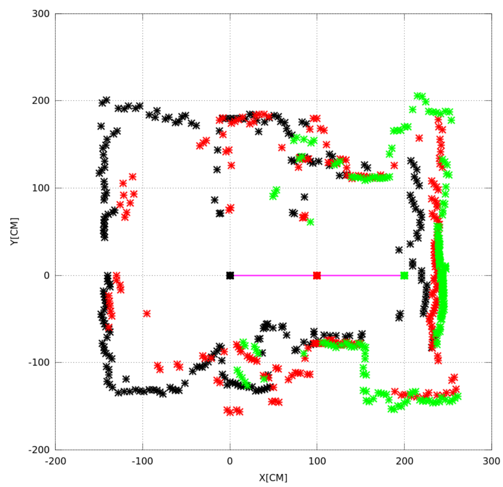


Fig. 10. Three scans collected approximately 100cm apart. Each scan is shown at the commanded robot pose with different colours. The commanded robot path is shown with magenta. Grid size is 50x50cm.

C. Sequence of Scans

To show what can one expect in less favorable conditions, the robot was commanded to travel 200cm in a kitchen while saving 3 scans into its EEPROM. The scans are cluttered with chair and table legs (see fig. 10 above the left half of the magenta line). Furthermore, horizontal edges as the bottom parts of radiators and cupboards which are aligned with the sensing axis of the sensors generate a lot of spurious measurements. As long range sensor is slightly tilted up to avoid getting measurements from the floor, the horizontal edges do not have to be at the same level as the sensor to influence measurements. Once the robot finished, it was hooked up to a PC and the scans were downloaded for visualization. The scans were plotted at the commanded robot poses which may differ from real poses.

V. Discussion and Classroom Use

The shown experimental results for scanning and scan matching indicate that Mappino has the potential to perform localization and mapping under laboratory conditions. The scan matching experiment has shown a surprisingly short scan matching time of 154ms on the employed 8 bit microcontroller on scans consisting of 256 range measurements.

The conservative estimates of Mappino's 20h standby and 2h of runtime battery life likely makes Mappino usable in classrooms.

However, there are limitations on the use of the robot. The internal EEPROM memory of the robot can only hold 4 scans. More memory can be hooked up through the exposed SPI port of the PSD scanner. The extra memory could also complement the 2KB RAM of the ATMEGA328 as it is barely enough for scan matching. Occupancy grid operations and path planning will likely not run at the same time as scan matching due to this memory limitation. Beside the use of external RAM, the upgrade of the ATMEGA328 is considered to a processor with 8KB of memory.

Even though there is a shortage of RAM, program memory seems abundant. In the current state of the robot only 12KB are taken up from the total of 32KB.

There are limitations on the environment as well due to the PSDs used for measuring range. These sensors do not work in all lighting conditions. One can expect interference from the sun and other light sources. Black surfaces may not reflect back enough light either for the sensor to work properly. Non-smooth surfaces may generate false measurements as well. The users therefore should start in simple, controlled environments, and as they advance their knowledge, they may be able to handle more and more difficult environments.

The stepper motors and wheel geometry enable accurate odometry, but the low motor torque, low robot clearance and low wheel traction only work well on hard, smooth and level surfaces. Beside table tops and linoleum flooring, the robot is expected to work on hard floor and tiles, but not on carpet. As the odometry of the robot is fairly accurate, scans could be taken during motion and then converted into virtual scans taken from one spot for matching purpose. This exercise, however is left for the user.

Regarding the educational value of the presented robot, there are several topics high school/university students or robotics enthusiasts can learn by using Mappino. The basic demo code provided shows the possibilities of platform and provides motivation to start working with the robot. After familiarization, they may decide to start by adding simple code for avoiding obstacles or following a person. One may continue by generating occupancy grids from a few scans. Once having an occupancy grid, path planning can provide heaps of fun. As users learn more and more about robotics, they may decide to improve odometry estimation and motion control. After grasping how scan matching works, they may get scan matching running on a moving robot as well. The modification of the provided scan matching code to work with bad readings could be also rewarding. Of course, one does not need to rely on points-based scan matching for localization. It is also possible for the users to learn about feature based localization and mapping in structured environments by extracting lines and corners from scans. Users may make hardware modifications as well. Extra electronics can be added to the prototyping areas. One can also redesign the mechanical bits of the robot and then re-use the electronics.

For younger kids the robot can be quite easily converted into a drawing robot. The PSD scanner can be replaced with a pen pointed at the floor. The motor and control electronics of the scanner can be converted into a pen lifting mechanism.

There are other possibilities regarding the robot. The programming serial cable can be replaced with a wireless link which would enable remote monitoring of the robot from a PC. The next step could involve familiarization with ROS by writing drivers for the robot and having ROS controlling the robot from a PC.

VI. Conclusion

In this paper the open source mobile robot Mappino was presented. Mappino is being developed to provide a fairly easy entry into the realms of robot localization and mapping using scan matching. Mappino achieves this goal by sporting two rotating IR range sensors as an inexpensive replacement for a laser range finder. Mappino can be programmed through the easy to use Arduino environment. The laser cut acrylic body is propelled forward by a pair of stepper motors powered by 6 AA batteries.

The shown experimental result indicates that the quality of scans taken by the robot is adequate for mapping and localization in controlled, well chosen environments. It has been shown that it is possible to perform scan matching on an 8-bit microcontroller.

Future work involves refining the mechanical design, addressing the problem of having inadequate amount of available RAM and creating more demos as proper on-the-fly localization using scan matching. The addition of a wireless communication link is also considered.

Acknowledgments

The author is very grateful for the reviewers' comments. Geoffrey Taylor's comments on the draft are gratefully acknowledged. The Taylor family's help of non-technical character during part of the development process is appreciated.

References

[1] K. Konolige, J. Augenbraun, N. Donaldson, C. Fiebig, and P. Shah, "A low-cost laser distance sensor," in *ICRA'08*, 2008, pp. 3002 – 3008.
[2] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *The International Journal of Robotics Research*, vol. 26(10), pp. 1125–1153, 2007.
[3] C. Joo and Y. Ryoo, "Self localization of mobile robot using multiple scanning infrared range system," in *ISIS'07*, 2007.
[4] Y.-H. Choi, T.-K. Lee, and S.-Y. Oh, "A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot," *Autonomous Robots*, vol. 24, pp. 13–27, 2008.
[5] I. J. Cox, "Blanche–an experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 193–203, april 1991.

A Low-Cost Real-Time Mobile Robot Platform (ArEduBot) to support Project-Based Learning in Robotics & Mechatronics

Ilya B. Gartseev[#1], Leng-Feng Lee[*2], and Venkat N. Krovi[*3]

[#]Moscow State Technical University of Radioengineering, Electronics, and Automation, Moscow, Russia

[1]gartseev@gmail.com

[*] State University of New York at Buffalo, NY 14260 USA

[2]llee3@buffalo.edu

[3]vkrovi@buffalo.edu

Abstract— **We discuss aspects of development of a low-cost real-time mobile-robot platform – ArEduBot - for educational experiments. Our framework leverages ease-of- programming in block-diagrammatic form within the MATLAB/Simulink environment, together with several special-blocks developed within our Arduino-Simulink Toolbox [1]. The executable, compiled using the Real-Time-Workshop toolchain, can be downloaded for standalone real-time execution on an Arduino controller board interfaced to an iRobot Create mobile base. Our goal is to deploy this framework in introductory robotics and mechatronics classes, to complement the lecture and to support project-based learning. From this perspective, we compare the ease-of-use of multiple deployment architectures, describe our block-implementation within the Arduino Simulink Toolbox, and present several example experiments created using this framework.**

Keywords— **project-based learning, mechatronics, robotics, arduino, simulink, matlab, educational platforms, mechatronics-enabled teaching and training; software design for system integration**

I. Introduction

Project-based learning encourages team-work, allowing students to learn from each other and preparing them for a real-work environments [1]. Our goal is to promote such project-based learning in various mechatronics/robotics courses (suitable for undergraduate seniors and incoming graduate students). To this end, we seek to create a flexible, open-ended, and easily programmable robotic kit to: (i) support students at various academic levels and with varying level of programming knowledge, (ii) support greater variety of algorithms and sensors, including the more complex ones in a deterministic real-time setting, while (iii) retaining the ease-of- programming in a high-level block-diagrammatic language.

Oftentimes, the selection of experimental test-platforms devolves to: (i) completely commercial-off-the-shelf (COTS) ‘robotic kits’[2, 3]; (ii) ‘robotic kit’ designed completely by the instructor [4]; or (iii) using COTS components to assemble a customized robotic kit [5-7]. Examples of COTS robotic kits include Lego MindStorms by the Lego Group [8], iRobot products by iRobot Corporation [9], URTK robots by RosUchPribor company [10], Boe-Bot Kit by Parallax, Inc. [11], Make Controller Kit by Makezine [12] teamed up with MakingThings and Arduino-family boards [13]. However, they come with vast differences in terms of ease-of-use, cost, as well as underlying performance which tends to complicate selection. While completely COTS robotic kits facilitate immediate-use, often with considerable supporting tutorial and project materials, they tend to be expensive. In addition, such COTS kits focus more on the hardware-interfaces and thus often feature relatively simple/restricted software interfaces.

At the other end of the spectrum, developing a robotic kit completely in-house, takes significant investment of instructor-time –for kit design, kit maintenance (controller, chips, or software) as well as developing the supporting pedagogic material. Nevertheless, the instructor has the freedom to tailor the projects to align well with lecture material at a relatively low cost.

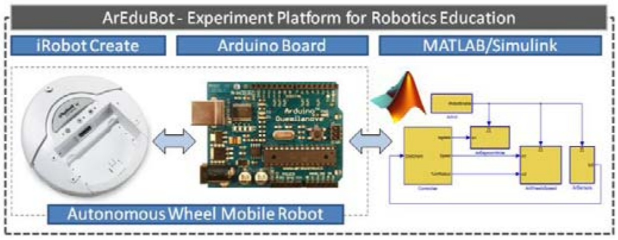


Figure 1: ArEduBot experiment platform consist of iRobot Create, Arduino Board, and MATLAB/ Simulink programming environment.

Hence, we sought an intermediary solution of building upon existing COTS components and adding to functionalities of basic robotic kit to create the ArEduBot. We chose the iRobot Create hardware (as a robust, inexpensive and well-supported platform), Arduino Board as our controller board (given its interrupt-based real-time capabilities, significant user- and code-base), and MATLAB/Simulink as the programming environment (given its wide usage for control courses). We propose to deploy the kit to support 3 robotics/controls courses: Mathematical Methods in Robotics

[1] The iRobot-Arduino-Simulink toolbox is available for downloading at http://www.gartseev.ru/projects/aredubot

(MAE 493/593) and Robotic Mobility and Manipulation (MAE 413/513) are senior/ graduate level course offered in the department of Mechanical & Aerospace Engineering at the University at Buffalo, and Automatic Control Systems in Mechatronics a undergraduate-senior level course offered in the department of Cybernetics in University 'MIREA' in Moscow, Russia. These three courses currently provide the theoretical flavor but lack a suitable easy-to-use platform for deployment. Most students came to the class with little or no prior knowledge in mechatronics, but are relatively comfortable with the MATLAB programming environment.

A. Deterministic Real Time Operating Systems (RTOS)

Deterministic real-time execution forms the bedrock of development of various estimation methods (e.g. velocity by finite differencing) or control methodologies (e.g. digital control). In the past, these requirements often restricted implementation to x86 or 68HC11 architecture systems that could facilitate low-jitter interrupt-based real-time code-execution. However, an often overlooked factor remains the ability to program such deterministic real-time algorithms within a user-friendly programming environment. Past deployments have required either direct assembly programming or at least C level programming. The auto-code-generation capability within a MATLAB/Simulink environment also placed minimal requirements for implementing interrupt based code – thereby effectively restricted deployment platforms to at least ARM-based processors (e.g. GumStix) or MPC555 (and their derivatives) .

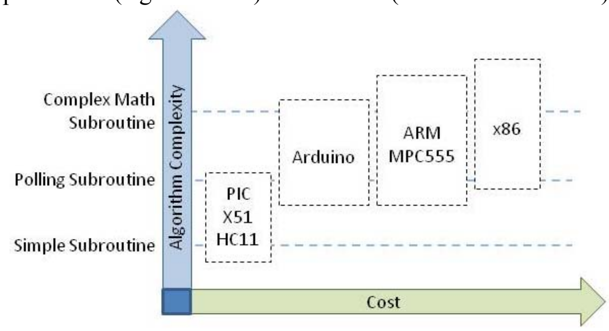


Figure 2: Using Arduino board fills the gap of handling complex algorithm at a relatively cost.

In recent times, the Arduino board has emerged as a viable processor-alternative to fill the gap between low-end microcontroller (PIC, X51, HC11) and the higher end processors (ARM, MPC555 and x86). However, there is a need for a framework to systematically develop deterministic real-time code. In this paper, we seek to remedy the absence of a suitable high-level block-library for the MATLAB Simulink, which can be used for block programming of the system.

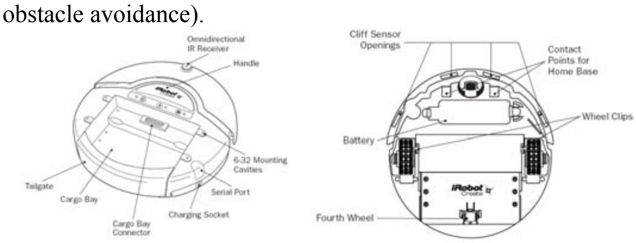
II. AREDUBOT COMPONENTS

It was crucial to develop an overall framework that allows students to easily test their algorithms on an inexpensive, real-time experimental testbed, within a relatively short period of time.

A. iRobot Create

The iRobot Create (shown in Figure 3) is a reprogrammable version of the Roomba robot vacuum cleaner for robotics hobbyists, educators, and researchers. This mobile platform is very actively used across the world to support research [14-16] and educational [17-19] activities in robotics and mechatronics.

The basic iRobot Create hardware, consists of two differentially-driven wheels, speakers, LEDs, cliff sensors, IR Receiver, serial port. iRobot Create features a command line API allowing for scripting from a command line interface via serial communication and can be programmed directly from PC. However, it is not always enough for the educational purposes and will be discussed further later in Section III. Nevertheless, this Create platform has been successfully employed to support a range of projects from easy (individually controlling wheels or reading sensors) to more complicated (trajectory planning with map construction and obstacle avoidance).



(a) **(b)**

Figure 3: (a) Top and (b) Bottom view of an iRobot Create Platform.

B. Arduino Board

The Arduino Board is a single-board open hardware design microcontroller, with embedded I/O support and a standard programming language [13]. While, there are variants of Arduino boards, we use the Arduino Duemilanove board [20] which retails for about $20. The ease of use of these boards is the reason of the wide spread usage for research [21, 22] as well as in education [23, 24]. Figure 4 depicts a sample Arduino board and contents of Arduino board kit while the core technical specifications are shown in Table 1.

(a) **(b)**

Figure 4: (a) Arduino Duemilanove board; and (b) A typical Arduino kit.

Arduino boards allows gathering information directly from digital sensors as well as by converting data from analog sensors using a built-in ADC chip. They also have the capability and necessary hardware to drive different kind of motors with PWM outputs, including DC motors, servos, stepper-motors. Together with synchronous- and asynchronous-serial communication capability, this simplifies the implementation of closed-loop servo-control for a variety of mechatronic systems. The compact footprint promotes in-situ embedding of these devices in-situ, connected to a computer solely via a USB board interface. The Atmega328 microcontroller at the heart of the board can be re-programmed with AVR studio or any other compatible programmer with code developed in assembler and C language. Moreover, programs can be stored to EEPROM for the further autonomous device functioning. In addition, processor emulation capability with software and electrical circuits jointly (e.g. using Proteus [25]) offers an invaluable opportunity for the student projects.

TABLE I
SPECIFICATION FOR ARDUINO DUEMILANOVE BOARD

Microcontroller	ATmega328
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
Flash Memory	32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

C. *MATLAB / Simulink*

MATLAB/Simulink provide a user-friendly programming environment for students to develop code and subsequently download the compile executable for standalone operation to Arduino board. We focus on creating an Arduino Simulink Toolbox, which facilitates the above process. At this stage, we would also like to draw a clear distinction with respect to the MATLAB Toolbox for Create (MTIC) [26]. The MTIC provides a wrapper for Create Open Interface and this permits direct serial-communication from MATLAB to the iRobot Create. All code is developed, stored and executed on the host PC/base station. In contrast, our effort, we focus on developing standalone downloadable executables for Arduino board from within Simulink. This distinction will be clear in Section III, as we discuss some of the commonly encountered interaction architectures for the iRobot Create.

III. EXTANT & PROPOSED INTERACTION ARCHITECTURES

We discuss the various interaction-architectures for the iRobot Create and their potential advantages and disadvantages, within an educational setting.

A. *Serial Port Communication (via API)*

1) *Terminal Program*

The first option is connect the iRobot Create with PC through serial port, as shown on Figure 5. The solid circle denotes the hardware iRobot Create while the solid rectangles indicate software components, solid diamonds for hardware components, dashed rectangles for full-systems and dotted rectangles for functioning run-time system components.

The iRobot Create can be controlled from PC by sending of the numerical control sequences over a serial connection. Any computer with a serial interface may be used as terminal, e.g. realterm. This option is very useful for early-stage interactions and initial testing with iRobot but requires a wired run-time connection to the computer. While the limitation of wired connection can be overcome by using the Bluetooth Adapter Module (BAM) [27] or by using a RF based serial interface (e.g. Super Screamer), this increases the cost of the whole system significantly. Further, the interactive line-by-line interpreted mode of operation using an unintuitive and very limited serial script language is restrictive.

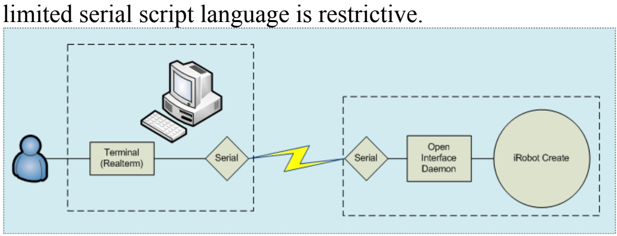


Figure 5: Control scheme for iRobot with terminal program

2) *MATLAB/ serial port*

The MATLAB Toolbox for the iRobot Create (MTIC) [26], shown in Figure 6, can be viewed as an extension of the above scripting process. It utilizes all of MATLAB computing power, to develop algorithms for iRobot Create. The toolbox function allow for translation of the high level commands (e.g. move forward) into the corresponding Open Toolkit API command directives (e.g. 134). In giving such wrapped function access to the iRobot Create it allows a programmer to write programs (instead of control with cryptic numerical sequences). However, from the hardware point of view, this option is equivalent to the PC terminal control, albeit with a slightly more capable software interface (MATLAB).

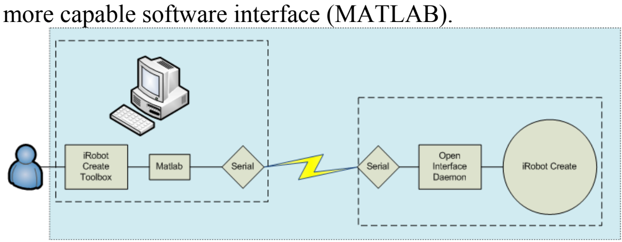


Figure 6: Control scheme for iRobot with MATLAB.

B. *Dedicated Micro Controller*

The Command Module Unit (CMU) for iRobot Create [28] is based on 8-bit, 18MHz Atmel Atmega 168 microcontroller, and can be powered by iRobot's battery. It enables full programmability of iRobot Create's motors, lights, sounds, and sensor sweeps and autonomous functioning. While permitting C programming, the CMU uses the Open Serial Interface to interact with the iRobot Create thus precluding direct control actuators or sensors. In Figure 7, the PC and CMU are connected via an USB interface for programming the flash-memory after which the connection may be removed.

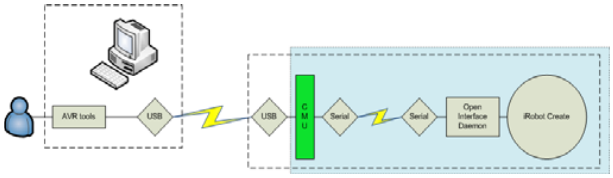


Figure 7: Control scheme for iRobot with CMU.

C. *Embedded PCs*

With embedded computers (x86 PC/104 system/ GumStix/ other ARM processor), it is possible to create a compiled executable using the MATLAB Real Time Workshop toolchain that can be downloaded to the memory of the embedded PC. The range of tasks that can be solved is very broad, by virtue of the greater computing power, but the cost is relatively high. The computational power permits more complicated and intensive computations (such as online trajectory planning). Moreover, most of the state-of-the-art embedded PCs, with wireless WiFi or Bluetooth interfaces, allowing the creation of wireless semi-autonomous/human-in-the-loop remote-controlled systems.

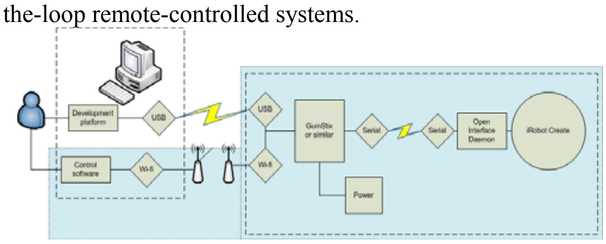


Figure 8: Control scheme for iRobot with embedded PC.

Figure 9 shows two ways of the running iRobot with microcontroller board. The short dotted rectangle depicts situation when iRobot and microcontroller are used in an autonomous mode. In this case, the development computer (high or low level) is used for compiling code and downloading onto the microcontroller board, while the system functioning in fully-autonomous mode at runtime. As a second alternative, highlighted by long narrow rectangle, the microcontroller board and controlling computer continue to use the Bluetooth or WiFi, interface as a data-logging /semi-autonomous control channel during runtime.

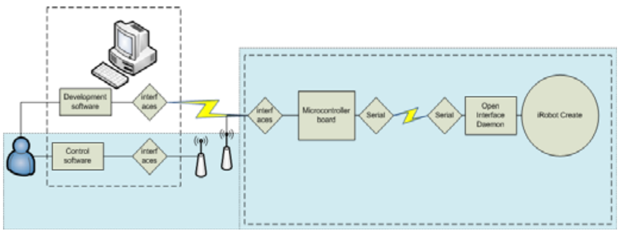


Figure 9: Control scheme for iRobot with microcontroller board.

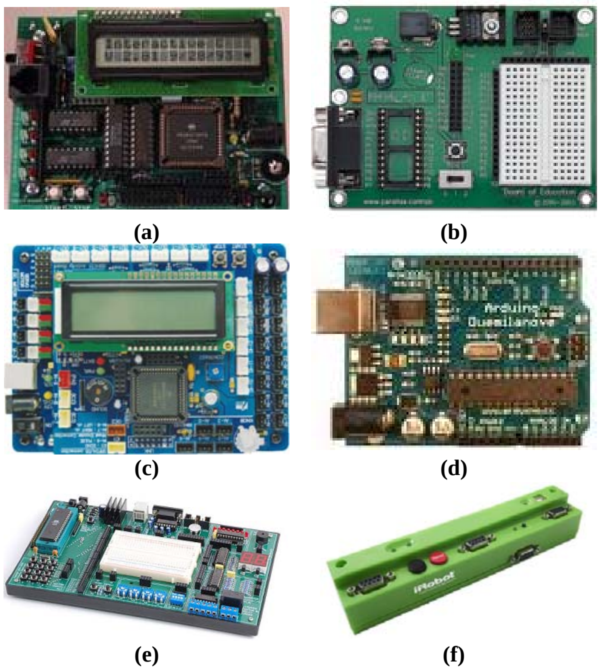


Figure 10: Microcontroller boards. (a) The Handy Board [29], (b) Parallax's Board of Education [11], (c) AX-11 Activity board [30], (d) Arduino Duemilanove [13], (e) SKIT-1RD2 V3 Experiment System [31], and (f) the Command Module Unit (CMU) from iRobot.

D. *ArEduBot system*

We take advantage of capability similar to that discussed in Section III.C, but with the Arduino Board mounted on the iRobot Create as shown on Figure 11. The Arduino Target tool [32] allows development of deterministic interrupt-based real-time code deployments for the Arduino platform right from Simulink. The target includes blocks to interface with the I/O ports on the Arduino board as well as a target file that automatically compiles and downloads the application onto the board directly from Simulink. We augment this process by developing a set of Level-2 S-function blocks encapsulated in a Simulink ArEduBot block library, discussed in Section IV. The MATLAB RTW toolchain can now be used to create executable machine language code directly from Simulink (unlike the MTIC [26] which also uses Simulink but only creates command level directives).

Figure 11: Hardware of ArEduBot system.

Figure 12 depicts logical structure of the ArEduBot system with three different ways of interacting with the ArEduBot: (i) low level programming; (ii) C level programming; or (iii) using the Arduino Target tool with our Simulink ArEduBot library for block-level programming of the Arduino board.

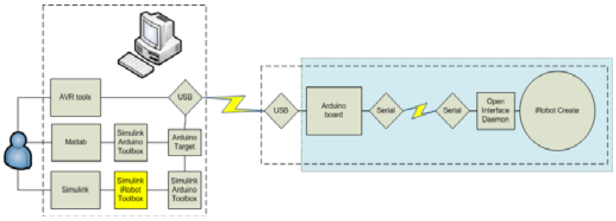


Figure 12: Control scheme for iRobot with Arduino board.

IV. SIMULINK AREDUBOT LIBRARY

A control process for iRobot Create consists of interaction with different hardware parts of this device: wheels motors, sensors, sound source, LEDs, battery and, possibly, external devices. Consequently, the structure of the ArEduBot library should reflect control of all of these parts as shown in Figure 13, and library by itself should consist of blocks that realize all respective actions. Accordingly to aforementioned structure, a set of blocks in ArEduBot library is shown on Figure 14.

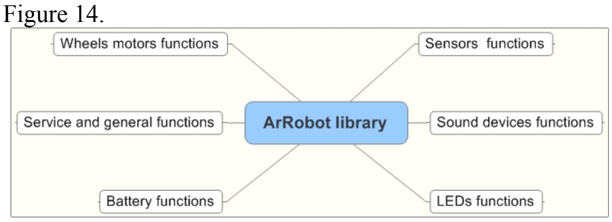


Figure 13: ArEduBot library structure.

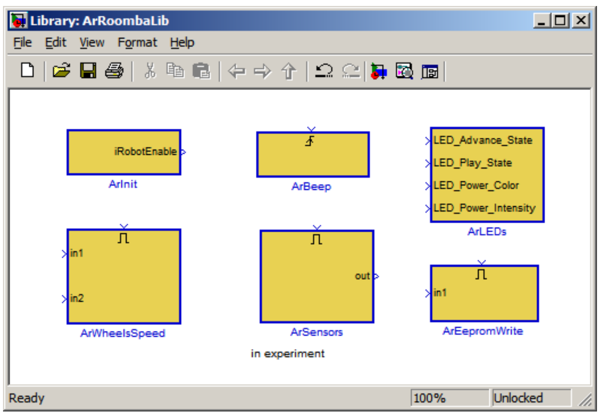


Figure 14: Blocks inside ArEduBot library.

A. ArInit block

This block initializes serial port for Arduino-iRobot connection and performs startup check of LEDs and beeper of iRobot Create. This block must be present in all models. The active level of output signal *iRobotEnable* of this block indicates that iRobot is initialized and ready to use.

This block has three parameters shown on Figure 15. *PauseBeforeInit* sets time before sending initializing sequence to the iRobot. *PauseAfterInit* sets time between sending initializing sequence to the iRobot and setting output signal to the active level. *InitType* sets one of initialization type: switching iRobot to the FULL operation mode only [33]; switching and lighting up LED indicators; switching, lighting up, and making a sound with beeper.

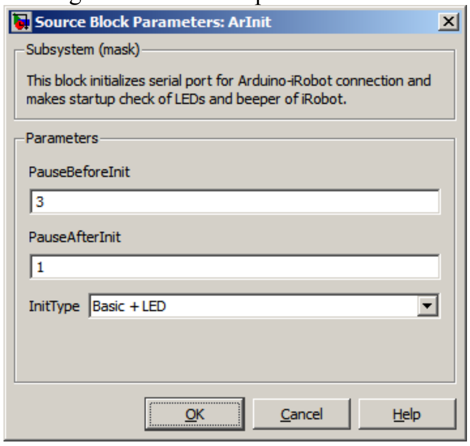


Figure 15: Parameters of ArInit block.

B. ArBeep block

This block initializes commands iRobot Create plays a sound that can be changed with block parameter *Song*. The parameter dialog is shown at Figure 16. Possible values for different sounds are discussed in [33].

Figure 16: Parameters of ArBeep block.

C. ArLEDs block

This block controls state of the LEDs indicators on the iRobot Create. Input signals *LED_Advance_State* and *LED_Play_State* can be 0 (LED is off) or 1 (LED is on) only. Signals *LED_Power_Color* and *LED_Power_Intensity* can be in the range [0..255]. The value 0 of *LED_Power_Color* signal corresponds to green color of Power LED, 255 corresponds to the red color. In the same way, the value 0 of LED_Power_Intensity signal corresponds to minimal intensity of the LED, 255 corresponds to maximum level of the intensity. This block sends new value through serial connection only when one of the input signals is changed.

D. ArWheelsSpeed block

This block control the forward and backward motion of the ArEduBot wheels in various ways. The window with the block parameters is shown on Figure 17. The parameter *Mode* allows choosing between control options. The predefined value *'Direct wheels speed'* allows using of independent control values for both wheels in range of [-500 mm/s; 500 mm/s]. With choosing *'Angular velocity'*, it becomes possible to use angular speed in rad/s as inputs of this block. Last mode *'Forward velocity and turn radius'* allows the control of both linear and turn motion of ArEduBot simultaneously. First input of the block sets linear speed in mm/s. Second input sets turn speed of the robot in rad/s.

This block sends new values through serial connection only when one of the input signals is changed and when Enable signal for the block has active level (more than 0). Commonly, Enable input should be connected to the Enable signal, generated by ArInit block; however, it can be used in ways that are more sophisticated.

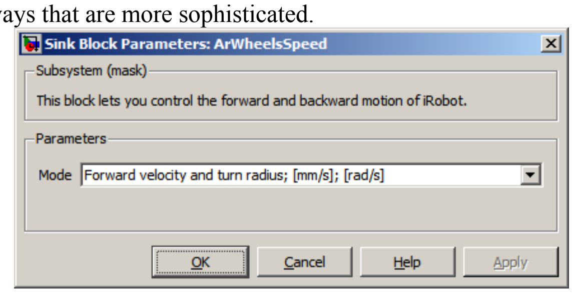


Figure 17: Parameters of ArWheelsSpeed block.

E. ArSensors block

This block creates an interface to the sensors of the iRobot Create. By means of parameter *Sensor Packet* user can get information from a sensor of interest. The value of this parameter is integer number in the range from 0 to 42 and it corresponds to the sensors packages predefined by iRobot Open Interface documentation. For example, if Sensor Packet parameter has value of 39, output signal of block will have vector with two values of current velocity of left and right wheels.

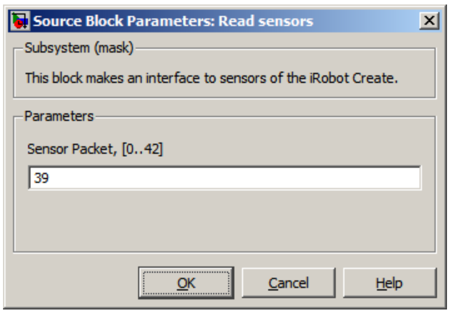


Figure 18: Parameters of ArSensors block.

F. ArEepromWrite block

This block writes input values to the EEPROM memory of Arduino board sequentially. The input signal should have the type of 8-bit unsigned integer. The meaning of input values can be any. The block has input Enable, which allows writing when stat is active.

V. Application Example

The combination of the above blocks allows one to create basic operations on the iRobot Crete through MATLAB/Simulink interface. Two such operations are shown here.

A. Case 1: Moving forward and backward

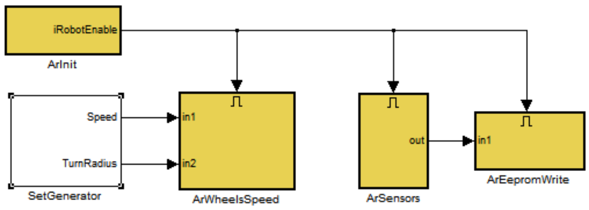


Figure 19: Application of ArEduBot library: moving forward and backward

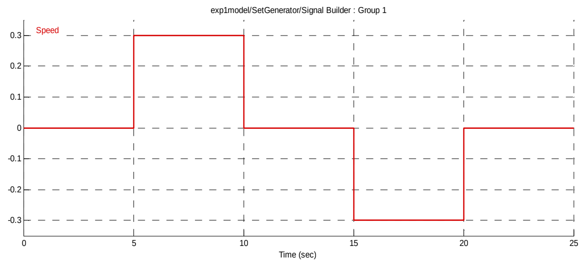


Figure 20: Desire speed input to the iRobot to move forward and backward.

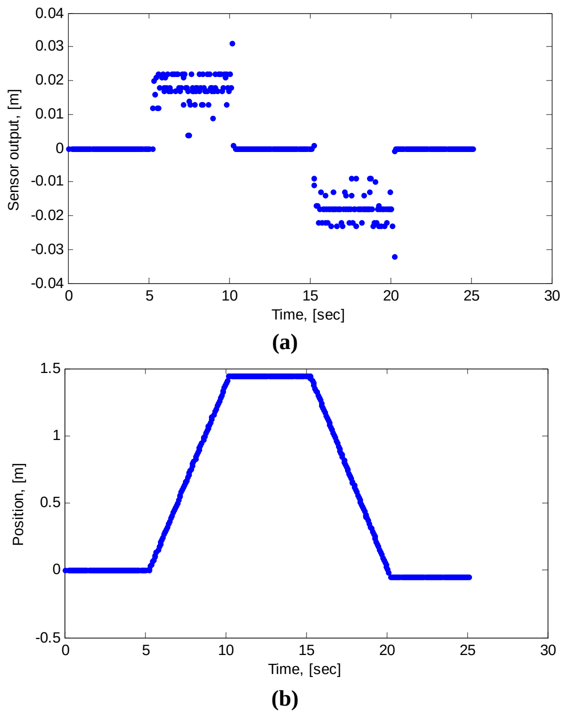


Figure 21: (a) Position differences output from the sensor; and (b) the position of the iRobot traveled by integrating the position difference in (a).

The typical application with ArEduBot library is shown on Figure 19. The block ArInit runs before anything else and generates active level of its output signal after successful initialization of the ArEduBot system. The block ArWheelsSpeed is working in the mode 'Forward velocity and turn radius'. The model gets desired speed and turn radius of iRobot from the block SetGenerator. Generally, they can be simply constants or more complicated time-dependent signals. Block ArSensor provides value of distance difference and orientation difference between current time step and previous time step. These values packed into two-components vector. Block ArEepromWrite stores values to the Arduino's EEPROM.

The input signal to the iRobot is shown in Figure 20, where we provide a forward speed input of 0.3m/s from time = 5 sec to 10 sec. Stop for 5 sec, and move backward at 0.3m/s for another 5 sec. Figure 21(a) shows the position difference reading from the iRobot sensor, and by integrate these sensor reading, one can obtain the position traveled by the iRobot shown in Figure 21(a).

B. *Case 2: Turning*

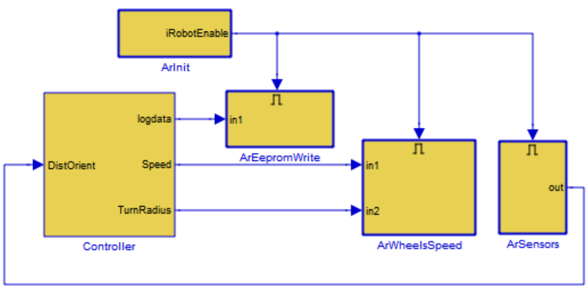


Figure 22: Parameters of ArSensors block.

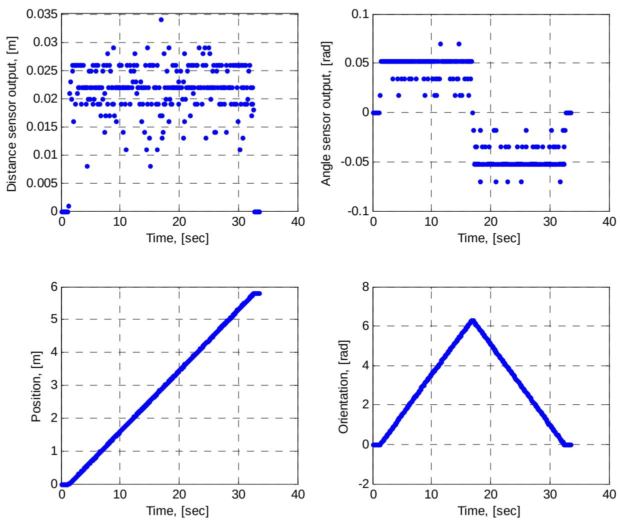


Figure 23: Position (top left) and angular (top right) differences output from the sensor; and the position (lower left) and angle (lower right) of the iRobot by integrating the position difference.

This example demonstrates a simple close loop control for the iRobot. In this case, the desired trajectory is a "∞" trajectory. The block Controller controls output signals to make iRobot perform motion along the two conjugate circles. The feedback from the sensors allows switching from the one circle to another. The signals from sensors and calculated robot' position and orientation are shown on Figure 23. The robot' trajectory is shown on Figure 24.

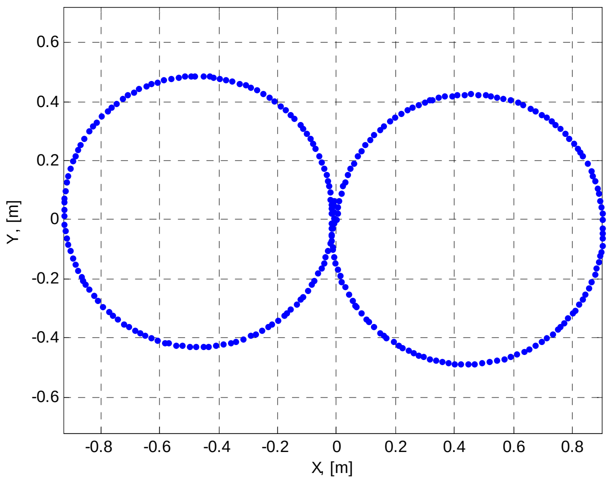


Figure 24: The robot traveled trajectory.

VI. Conclusion

We presented a Simulink Toolbox for ArEduBot, that uses iRobot Create as the mobile platform, Arduino board as the controller, and MATLAB/Simulink as the programming interface, to serve as an experimental platform for robotic courses. The ArEduBot has the advantage of the ability to utilize many of the MATLAB/Simulink blocks and its convenient programming environment to create executables for standalone deterministic real-time execution on the Arduino board interfaced with the iRobot Create, at relatively low cost. This Toolbox is intended for use in introductory robotics class where student have limited background in mechatronics but want to create a robotic system to implement what they have learned in lecture. We are in the process of deploying this Toolbox in multiple robotic classes to support the associated group projects.

References

[1] J. Chizmar and D. Williams, "What Do Faculty Want?," *EDUCAUSE Quarterly*, vol. 24, pp. 18-24, 2001.

[2] F. Klassner, "A Case Study of LEGO Mindstorms'™ Suitability for Artificial Intelligence and Robotics Courses at the College Level," in *33rd SIGCSE technical symposium on Computer science education*, Cincinnati, Kentucky 2002.

[3] S. Parsons and E. Sklar, "Teaching AI using LEGO Mindstorms," in *AAAI Spring Sympos on Accessible Hands-on Artificial Intelligence and Robotics Education*, 2004.

[4] J. Fiene. (2010). *The MaEvArM Project*. Available: http://alliance.seas.upenn.edu/~medesign/wiki/index.php/Guides/MaEvArM

[5] I. B. Gartseev and E. N. Kryuchenkov, "Hardware Development for Autonomous Mobile Robot," in *State-of-the-Art Information Technologies*, Sudak, Ukraine, 2006.
[6] I. B. Gartseev, *et al.*, *The Training Module Robot UMR-2: Datasheet and Manual*. Moscow: RosUchPribor-MIREA, 2002.
[7] I. B. Gartseev, *et al.*, "Unified Control System for Mechatronic Modular Robots," in *State-of-the-art Technologies in Control, Automatics, and Computer Science*, Ukraine, 2003.
[8] LEGO. (2010). *Lego MindStorms*. Available: http://www.lego.com/eng/info/
[9] iRobot. (2010). *iRobot Create*. Available: http://www.irobot.com/
[10] RosUchPribor. *URTK Robots*. Available: http://www.rosuchpribor.ru/
[11] Parallax. *Boe-Bot Kit*. Available: http://www.parallax.com/
[12] O'Reilly. *The MAKE team*. Available: http://makezine.com/about/
[13] Wiki-resource. *Arduino*. Available: http://www.arduino.cc/
[14] D. Smith and Z. Dodds, "Visual navigation: Image profiles for odometry and control," ed, 2009, pp. 1188-1189.
[15] N. Correll, *et al.*, "Ad-hoc wireless network coverage with networked robots that cannot localize," ed, 2009, p. 3878.
[16] Z. Koziol, *et al.*, "A vision for spatial-reasoning commodity robots," ed, 2009, p. 195.
[17] Z. Dodds and B. Tribelhorn, "Erdos: cost-effective peripheral robotics for AI education," ed, 2007, pp. 1966-7.
[18] P. Mawhorter, *et al.*, "A Tale of Two Platforms: Low-Cost Robotics in the CS Curriculum," vol. 24, ed, 2009, pp. 180-188.
[19] I. O. Richard Weiss, "Finding your bot-mate: criteria for evaluating robot kits for use in undergraduate computer science education," vol. 24, ed, 2008.
[20] D. Jolliffe. (2006). *Arduino Fever*. Available: http://cdn.makezine.com/make/arduinoMAKE07.pdf
[21] B. Batagelj, *et al.*, "Digital airbrush," in *ELMAR-2009 - 51st International Symposium ELMAR-2009, September 28, 2009 - September 30, 2009*, 2009, pp. 306-308.
[22] D. Bri, *et al.*, "A multisensor proposal for wireless sensor networks," presented at the 2nd International Conference on Sensor Technologies and Applications, SENSORCOMM 2008, Cap Esterel, France, 2008.
[23] J. D. Brock, *et al.*, "Using Arduino for Introductory Programming Courses," vol. 25, ed, 2009.
[24] H. Blemings and J. Oxer, *Practical Arduino: Cool Projects for Open Source Hardware*, 1 ed.: Apress, 2009.
[25] L. E. Ltd. *Proteus PCB Design Software*. Available: http://www.labcenter.co.uk/index.cfm
[26] J. M. Esposito, *et al.* (2008). *Matlab Toolbox for the Create Robot*. Available: www.usna.edu/Users/weapsys/esposito/roomba.matlab/
[27] I. Element Products. (2007). *BAM for the iRobot® Create™. User Manual*. Available: http://www.elementdirect.com/files/10542B.pdf
[28] i. Corporation. (2007). *iRobot® Command Module. Owners manual*. Available: http://www.irobot.com/filelibrary/create/Command%20Module%20Manual_v2.pdf
[29] *The Handy Board*. Available: http://handyboard.com/hb/
[30] V. R. Kit. (2008). *VegaROBOKit*. Available: http://vegarobokit.com/
[31] P. Limited. (2009). *SKIT-51RD2 V3 Experiment System*. Available: http://www.paltronix.com/skit-51rd2v3.htm
[32] D. Eastman. (2009). *Arduino Target*. Available: http://www.mathworks.com/matlabcentral/fileexchange/24675
[33] iRobot. (2006). *iRobot Create Open Interface Specification*. Available: http://www.irobot.com/

Electronic Platform for Small Robots in Education

Raimund Edlinger[#1], Michael Zauner[*2], Walter Rokitansky[*3]

[#]FH OOE Forschungs & Entwicklungs GmbH
Stelzhamerstr. 23, 4600 Wels, Austria
[1]raimund.edlinger@fh-wels.at

[*]FH OOE Studienbetriebs GmbH
Stelzhamerstr. 23, 4600 Wels, Austria
[2]michael.zauner@fh-wels.at
[3]walter.rokitansky@fh-wels.at

***Abstract*— Robotics is a scientific discipline which needs a high level knowledge in the fields of computer science, as well as electrical and mechanical engineering and high-performance modular control systems. This paper reports the first results of a project that uses a previously developed very small Electronic Platform (Miniboard) for small and simple autonomous mobile robots. This board is used in the education program with bachelor students in their third and fourth semester. It should help them to build robots, like Sumos or Line Followers, so they can get first experiences in robotic. We demonstrate the robustness of this approach in controlling indoor mobile robots for the RobotChallenge in Vienna.**

***Keywords*— wheeled mobile robots, WMR, controlling system, robotic hard and software architectures, RobotChallenge**

I. Introduction

Robotics is a scientific discipline which needs a high level knowledge in the fields of computer science, as well as electrical and mechanical engineering and high-performance modular control systems. This paper reports the first results of a project that uses a previously developed very small Electronic Platform for small and simple autonomous mobile robots. Put simply, the Miniboard consists of a single-processor system which handles all tasks by itself.

To easy the students the entrance into the robotic, a full driver library is supported, e.g. for ADC or Motor Control. In bigger robots, like Line Follower or Mega Sumos the Miniboard can be implemented, as well as in very small robots like the Mini or Micro Sumos.

II. Related Work

Much of the recent work in robotics has used embedded systems such as PC 104 [9], Mini-ITX [8] or RNFBFRA-Board [7]. In contrast to the proposed concept, these systems are very big and have more computing power than needed for small robots. To control actuators or sensors, additional boards are often needed, this also increases the necessary space for the electronic.

These are facts to complicate the design of robots for the students or make it impossible, e.g. if the board has bigger dimensions than the robot is allowed to have.

III. Hardware Design

At the beginning of the development of the Miniboard (as shown in Fig. 1 and Fig. 2) was one main point to design a board with dimensions fewer than 5 cm by 5 cm. Because one goal was to implement the board into Micro Sumo robots and these robots have a dimension of 5 cm by 5cm by 5 cm.

On the board the micro controller ATMega644 [6] is used. Because this kind of micro controller is also used at the Modular Electronic System [1] and so the students get to know the ATMega644. The board is designed to be the stand alone control unit in a small robot.

It can deal with:

- up to two DC brushed motors
- up to 14 digital Inputs/Outputs
- up to four analogue Inputs

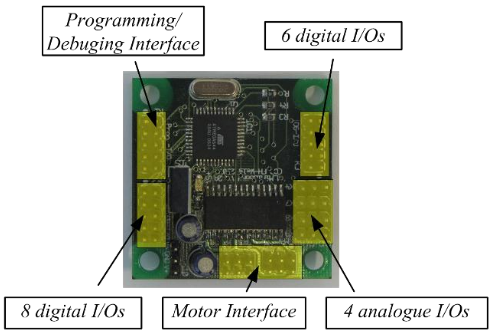


Fig. 1 Miniboard Topview

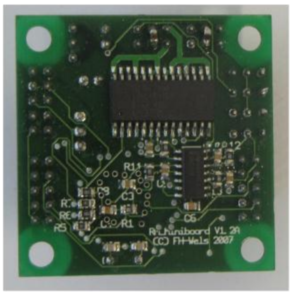


Fig. 2 Miniboard Bottomview

A. Motor Control

The motor board has two full H-Bridges BTS7750G [14] from Infineon to control two DC motors. The BTS 7750G is part of the TrilithIC family and has the following features:

- low RDSon: 70 mΩ high-side switch, 45 mΩ low-side switch
- maximum peak current of 12 A
- full short-circuit protection
- operates from 3.5 V to 42 V
- PWM frequencies up to 1 kHz

Part of the motor control is to manage the speed, the position and the odometric navigation system.

B. Digital Inputs/Outputs

The board supports up to 14 digital Input/Outputs. To these I/Os can be connected various sensors, e.g. infrared sensors to detect a black line in Line Following competition or to become aware of an obstacle or an opponent robot in Sumo competition. Also output interfaces, e.g. LCD interface can be connected.

C. Analogue Inputs

To the board also can be connected up to four analogue input signals. The inputs have a voltage range from 0 V to 5 V and the ADC has a 10 bit resolution. They can be used for measuring the distance to an obstacle or an opponent robot.

These inputs can also be used as digital input or output, if needed.

D. Power Supply

To stabilize the 5 V for the micro controller a LM7805 is used. This kind of voltage regulator can handle an input voltage up to 20 V. The drawback is, at a high difference between input and output voltage the power dissipation is also high.

This is the reason why an additional board was developed. On this board is a step down converter witch can handle input voltages up to 42 V with a power factor from 90 % over the whole input voltage range from 8 V to 42 V. This board can be soldered instead the LM7805.

Usually LiPo accumulators with 7.4 V or 11.1 V are used.

IV. Software Driver

When the students start with there first project, they can use a software library, which set up the whole hardware modules of the ATMEGA644, e.g. the ADC or the timers.

Additional to these low level drivers also high level drivers was developed, e.g. motor control or a cooperative multitasking system. So it is possible to develop software for the main task without dealing a long time to setup the processor or a LCD interface.

A. Motor Driver

The motor driver is divided into different tasks. The first task is to handle the increment encoder and calculate the actual speed and position. Afterwards the software controls the speed and position with a PI-regulator.

To set up the motor controller only the speed and the position have to be set.

B. Multitasking System

To perform more jobs at one time a high performance and self developed cooperative multitasking system [2] is implemented. It provides:

- only 320 byte of code
- only 26 byte RAM per task
- event-triggered tasks
- cyclic tasks
- communication between the tasks with mailboxes

V. Experimental Results

The Miniboard is used in a lot of robots, e.g. Line Followers or Mini Sumos. In this chapter three of them will be introduced.

A. Arrow [3]

Arrow (see Fig. 3) is a Line Follower and participated in 2009 at the RobotChallenge in Vienna.

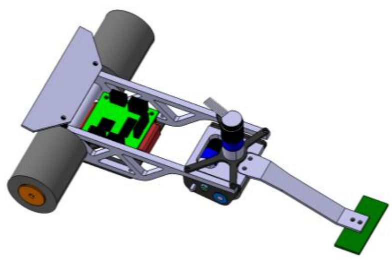


Fig. 3 Line Following robot "Arrow"

It uses both motor ports, one to direct and one to actuate the robot. In the front of the robot is a Printed Circuit Board (PCB) with five infrared sensors to detect the black line. This board is connected to the digital inputs.

B. Dark Knight [4]

Dark Knight (see Fig. 4) is a Mini Sumo robot and participated in 2010 at the RobotChallenge in Vienna.

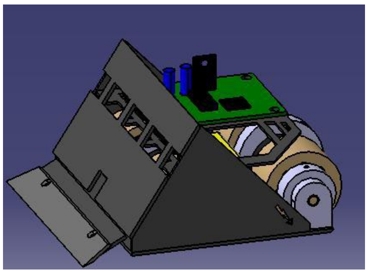


Fig. 4 Mini Sumo robot “Dark Knight”

It uses both motor ports to drive the robot. In the front of the robot are four infrared sensors to detect the opponent robot. These sensors are connected to the digital inputs. The students place a DIP switch on the robot to choose different start scenarios as shown in Fig. 5.

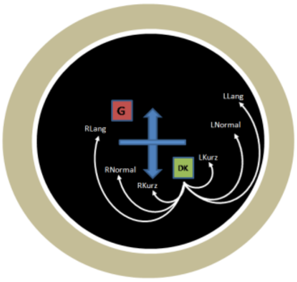


Fig. 5 different start scenarios

C. Dozer [5]

Dozer (see Fig. 5) is a Mini Sumo robot and participated in 2011 at the RobotChallenge in Vienna.

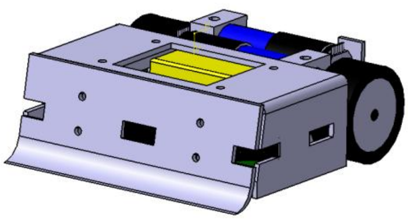


Fig. 6 Mini Sumo robot “Dozer”

It uses both motor ports to drive the robot. In the front of the robot are three and on each side is one infrared sensor to detect the opponent robot. These sensors are connected to the digital inputs.

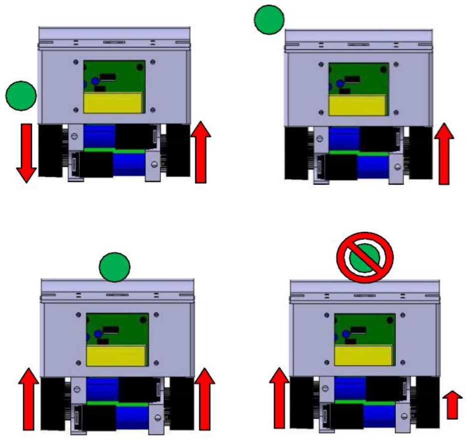


Fig. 7 searching algorithm

Depending on which sensor is detecting an opponent robot, “Dozer” moves hard left, left, forward or right, as shown in Fig. 7.

VI. Further Work

The board works very well and the students were very successful with there robots. A big advantage is the small size of the board, so it can even easy implemented in small robots.

It was developed in year 2007 to participate at the Robot Challenge [11] in Vienna. Students built a Mini Sumo robot that won the first price in Mini Sumo competition and in Slalom Enhanced competition.

Since these time a lot of new electronically parts have been launched, e.g. more efficient micro controller or motor driver.

In the next generation it is planned to use the new micro controller Xmega [12] from Atmel. This controller has 32 MIPS instead of 20 MIPS the ATMega644 has. It also has three full QDEC implemented and lots of other features. It is also planned to use a new motor driver, because the BTS7750G has a maximum switching frequency from 1 kHz. The new type has a frequency from up to 100 kHz.

VII. Conclusions

The Miniboard for the construction of small autonomous robots has turned out to be very useful with a lot of I/Os to connect sensors, actuators or LCD module. With this system it is possible to process tasks at the same time. A LCD module enables the user to carry out changes in the robot, without changing the program and so a connection to an operator station is not necessary. Moreover, important information about the state of the robot is displayed on such a module. The introduced system offers an amount in possibilities to build robots in short time with wide activity field offers for adaptations.

ACKNOWLEDGEMENTS

The authors grateful acknowledge the support of the Upper Austria University of Applied Sciences, Elra Motors, Beta Layout and ifm for setting up the Robo Racing Team.

REFERENCES

[1] R. Edlinger and M. Zauner, "A Modualr and Scalable Electronic System for Mobile and Autonoumous Robots" in *EUROBOT conference*, 2010, paper.

[2] T. Bräunl, "Embeded Robotics", Springer Verlag Berlin Heidelberg, 2003

[3] C. Mittendorfer, B. Muckenhumer and M. Pautzenberger, "Projektbericht Linienverfolger Arrow", 2009, technical report,

[4] G. Sandberger, W. Schinnerl and L. Schlossinger, "MINISUMO [Dark Knight]", 2010, technical report, http://rrt.fh-wels.at/publications/technical_papers/FUP_2010_Minisumo_Dark_Knight.pdf

[5] T. Krößwang, S. Meisinger and M. Zangenfeind, "Projektbericht Minisumo-Roboter Dozer", 2010, technical report, http://rrt.fh-wels.at/publications/technical_papers/FUP_2010_Minisumo_Dozer.pdf

[6] Atmel, http://www.atmel.com/products/avr/default.asp, 2006

[7] RoboterNETZ, http://rn-wissen.de/index.php/RNBFRA-Board , 2010.

[8] Mini-ITX, http://de.viatech.com/de/initiatives/spearhead/mini-itx_2.0/, 2010.

[9] PC104, http://www.pc104.org/, 2010.

[10] Infineon, http://www.infineon.com/cms/en/product/findProductTypeByName.html?q=BTS7750, 2010.

[11] RobotChallenge, http://www.robotchallenge.org/

[12] Xmega, http://www.atmel.com/dyn/products/devices.asp?category_id=163&family_id=607&subfamily_id=1965&source=left_nav

IMU Platform for Workshops

Lukáš Palkovič[*1], Jozef Rodina[*2], Peter Hubinský[*3]

[*]Institute of Control and Industrial Informatics
Faculty of Electrical Engineering, Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava, Slovakia

[1]lukas.palkovic@stuba.sk
[2]jozef.rodina@stuba.sk
[3]peter.hubinsky@stuba.sk

***Abstract*— This document describes a platform planned to use in our workshops on Robotics. It will help the students understand the way how accelerometers and gyroscopes are used for the inertial measurements. Students can train various methodologies of processing the signal coming from these sensors.**

***Keywords*— MEMS, Inertial Measurement Unit, Accelerometer, Gyroscope, Sensor Fusion**

I. Introduction

MEMS components are widely used in the field of robotics. Their usage in some applications is quiet simple. However, more complicated applications such as inertial measurement units need more complex approach to the processing of their output.

Inertial measuring unit measures inertial state variables of an object in space, for example orientation, velocity and gravitational forces. It can be aircraft, space satellite or ground robot (for example Segway®). Usually accelerometers and gyroscopes are used as their basic sensors. MEMS versions of these sensors are nowadays gaining more and more importance.

MEMS sensors have several drawbacks. So combinations of more sensor types are often used to compensate the drawbacks of each other and ensure much better properties of the whole system.

There are many ways [4] how to combine signal from these sensors. It is called sensor data fusion. For example complementary or Kalman filters can be used.

Since it is more interesting to work with real devices than just with simulations and our workshops on service robotics are practically oriented a modular platform with a sensor unit was created to implement these techniques and investigate the properties of the sensors.

II. Mechanical Design

To avoid pure simulation and make the work for students more practical and interesting the design of the platform had to deal with real hardware. It consists of a mechanical part providing some pre-defined movements and an electronic part with sensors and signal processing hardware.

Fig. 1 is a simplified sketch of the mechanical construction. There is a position servo motor mounted on a vertical wall with a drawn protractor. Shaft of the motor is oriented horizontally and holds a deck with the printed circuit boards.

There are two boards. One of them contains the digital signal controller and the other one is the sensor board. Its advantage is the possibility to change the sensor board and test various sensors.

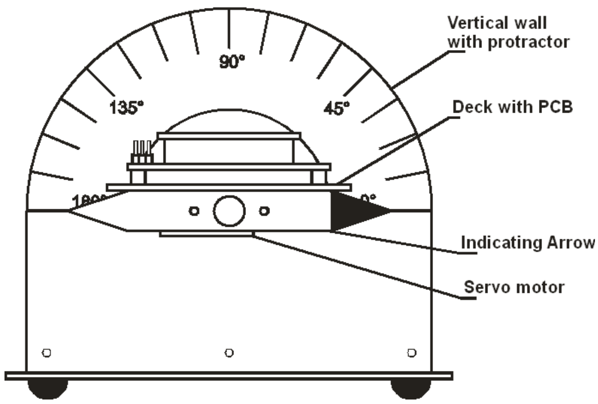


Fig. 1 Mechanical construction of the stand

The angle between the horizontal plane and the plane of the sensor board is indicated by an arrow and it can be easily visually measured on the protractor.

When moving the motor this angle changes and the measuring unit consisting of the sensors and the controller should be able to measure this angle and the angular rate of the rotation just from the information based on sensed gravity projection and the gyroscopic moment.

If an additional mechanical arm is used, the sensor board can be located further from the ax of the rotation and the signal can be more influenced by centrifugal forces.

The goal of this application is to let students process the signal from the sensors, compare the results with the precise angle and develop a method obtaining the best results.

III. Electronics

The block diagram of the schematic is shown in Fig. 2.

The most important parts of the circuit are the MEMS sensors located on the sensor board. Our sensor board contains 3-axis analog output accelerometer MMA7361 by Freescale, dual gyroscopes LPY530AL for measuring in X and Y axis and LPR530A by ST Microelectronics.

The signal from the sensors is then converted and processed by a 16 bit digital signal controller dsPIC33FJ64GP306A by Microchip. It is programmed and debugged via ICSP connector and PICKIT microcontroller.

The servo motor HexTronik HXT500 is controlled by standard PWM signal with frequency 50Hz and impulse duration 1.5ms. The motor was modified in order to obtain its internal position feedback, which is usually not available. The position is measured by an internal potentiometer. The signal from it is also measured by the digital signal controller and used as the real value of the angle and is compared to the angle obtained by the inertial sensors.

The controller is connected to a PC via serial port. USB/Serial port adapter is used.

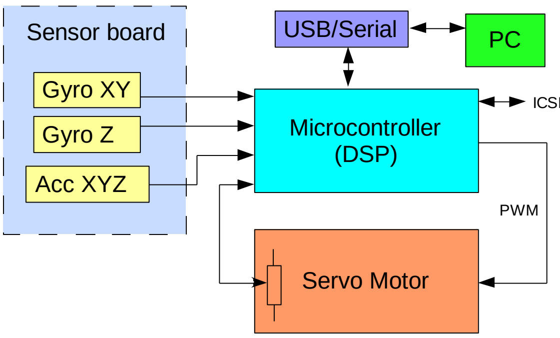


Fig. 2 Block diagram of the circuitry of the platform

IV. Software

There is a pre-prepared application for the digital signal controller. It is programmed in C language and MPLAB IDE is used for programming and debugging during the workshops.

The pre-prepared application has all the basic functionality prepared to work with the signal from the sensors and to drive the servo motor. Things such as interrupts analog/digital conversion, serial port reception and transmit are ready to work with. The assistant will show students how to access the peripherals and they can work on their own modifications.

User interface is very simple and it is based on serial port communication. There are some basic commands implemented, for example:

- go to vertical position
- go to horizontal position

Output is also solved this way. Measured data are sent to the serial port and shown in the following format.

The values are Tab-separated. Period of the data is 10ms. Lines are separated by carriage return character. The values are printed in this order (the meaning is explained in section V):

- Time [ms]
- Angle from potentiometer [degree]
- Angle from filtered gyroscope [degree]
- Angle from filtered accelerometer [degree]
- Angle directly from gyroscope [degree]
- Angle directly from accelerometer [degree]
- Angle from complementary filter [degree]

Since the transfer is in text format, it is easy to use a terminal program to communicate with the stand. Received data can be logged into a file and evaluated in another program.

The students can feel free to modify the output according to their requirements.

We are considering writing a special program or integrate the system to Matlab Simulink too. It might move the complexity of the algorithms from C in controller to sophisticated environment in PC.

V. Algorithms

A. MEMS Accelerometer and Gyroscope Properties

MEMS accelerometers are good sensor especially for static measurements as an inclinometer in cell phones etc. Their output contains high frequency noise and it is sensitive to centrifugal acceleration which can negatively affect the inertial measurement.

In the other hand, MEMS gyroscopes measure angular rate naturally. Hence they have good high frequency response, while integrating of little deviations in steady state modes causes a drift in the output.

It would be useful to find a way how to use just good properties of both types of sensor.

B. Complementary Filter

Complementary filter is a technique used to combine noisy signals. These noises have complementary spectral characteristics.

Assume two measurements y_1 and y_2 of the signal x where μ_1 is high frequency noise and μ_2 are low frequency disturbances.

$$\begin{aligned} y_1 &= x + \mu_1 \\ y_2 &= x + \mu_2 \end{aligned} \quad (1)$$

Assume two complementary transfer functions $F_1(s)$ and $F_2(s)$ which create our complementary filter.

$$F_1(s) + F_2(s) = 1 \quad (2)$$

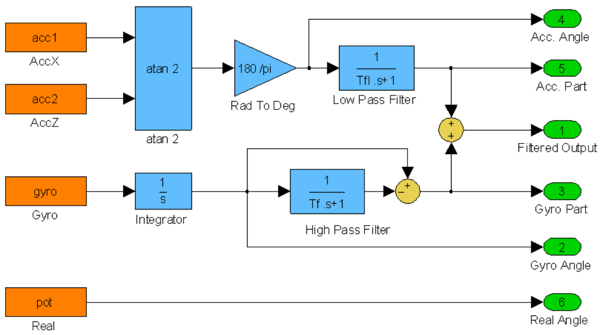


Fig. 3 Block schematic of complementary filter

$F_1(s)$ is a transfer function of low pass filter and $F_2(s)$ is a transfer function of high pass filter.

It implies that our complementary filter passes whole frequency range; therefore whole frequency range of the signal x. Low pass filter $F_1(s)$ is designed to suppress the noise μ_1 and filter $F_2(s)$ should filter the slowly changing disturbance μ_2.

$$\hat{X}(s) = F_1(s)Y_1(s) + F_2(s)Y_2(s) = \\ = X(s) + F_1(s)\mu_1(s) + F_2(s)\mu_2(s) \qquad (3)$$

Fig. 3 shows a block schematic of complementary filter used at our platform.

C. Kalman Filter

Kalman filter is a widely used state estimator. Like other methods, it takes into account model of the controlled system. Unlike the classical approaches, it takes into account also the properties of the measurement (noise, other disturbances). Kalman filter requires the state space model of the system and is usually applied in its discrete version.

Principle of Kalman filter is more difficult to understand than complementary filter and it is also more computationally complex (matrix inversion, transposition and multiplication) what used to be a problem but nowadays it is possible to easily implement them in modern digital signal controllers.

More information on Kalman filter can be found in [2].

We would like the students to work also on this method.

VI. Example

To show the properties of the signal from the sensors and its processing, the following experiment with complementary filter was done.

A step input of angle from horizontal to vertical position (90 degrees) was provided. Fig. 4 shows the signal obtained by the accelerometer and the signal measured directly by potentiometer of the servo. It can be seen that the signal from accelerometer has an oscillating response with a noticeable overshoot. Its steady state value corresponds with the real angle. Hence, the properties for low frequencies are much better than the properties for higher frequencies.

Integrated signal from gyroscope is shown in Fig. 5. The transient response corresponds with the real value but the steady state value is rising in time. Signal with such drift cannot be used directly for angle measurement. In this case, the advantage of gyroscope is its function on higher frequencies and its drawback is at lower frequency range.

Fig. 6 shows filtered signals from accelerometer and gyroscope. A low pass filter is used for accelerometer and a high pass filter for gyroscopes. Fig. 6 shows also their sum which is compared to real angle in Fig. 7.

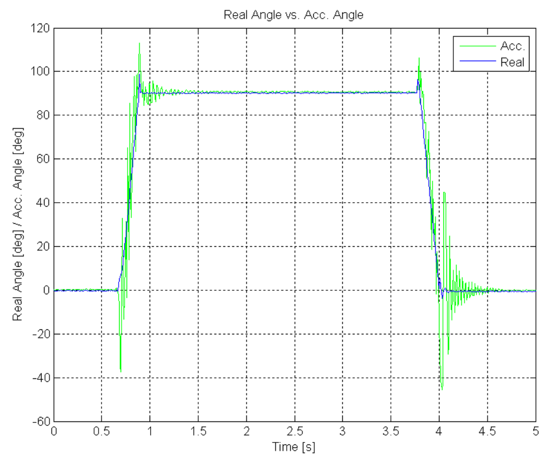


Fig. 4 Comparison of angle directly obtained from accelerometer and real angle

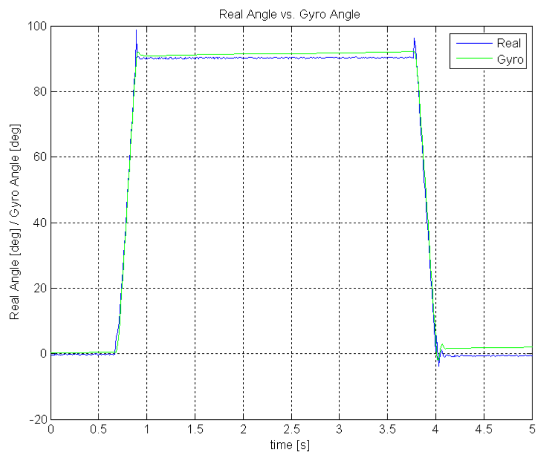


Fig. 5 Comparison of Angle obtained by integrating gyroscope signal and real angle

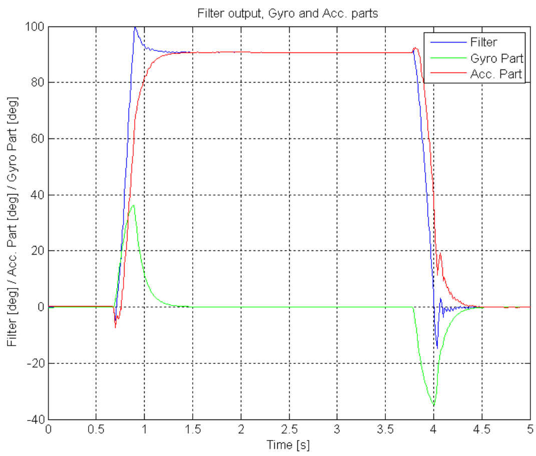


Fig. 6 Comparison of filtered signal from gyroscope and accelerometer and their sum

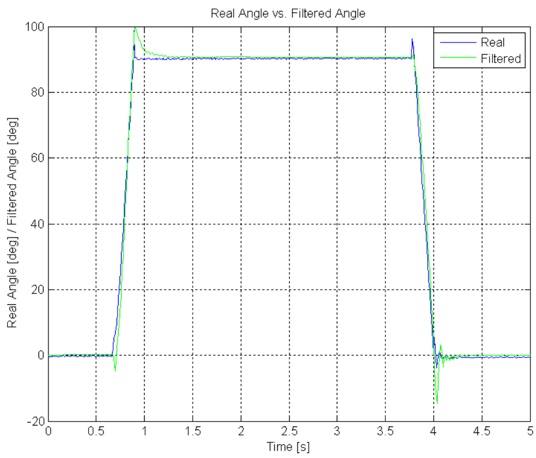


Fig. 7 Comparison of filtered angle and the real angle

VII. Workshops

There are several issues which the students can deal with. First they should get familiar with the platform by modifying the output of the digital signal controller. This requires some knowledge of embedded system programming in C language.

They can control the PWM output of the controller, so they are able to control the motor position. More commands or more sophisticated interface can be implemented for motor control than mentioned in chapter [IV]. Either desired trajectory can be provided from an external source via serial port or some trajectory can be preprogrammed in the flash memory.

The signal processing is the most important part. Students learn how to obtain the information from the sensors and try to compute the state (angle and angular rate) from it. They can compare it with the real trajectory obtained from the potentiometer. Students can implement aforementioned algorithms. Complementary filter will be obligatory. Depending on time reserved for this platform Kalman filter or quaternions can be also applied.

It is preferable to implement the algorithms in the digital signal controller; however, it is possible to do it in more complex software in personal computer.

VIII. Conclusions

A new platform equipped by modern MEMS sensors and digital signal controller was introduced. The platform contains also a plane rotating on the motor shaft. The angle of this plane is measured by processing the signal from the sensors.

The purpose of the platform is to train students on implementation of various algorithms for the estimation of the angle and its derivation.

A description of the mechanical and electrical part was provided together with a simple algorithm based on complementary filter. This algorithm was verified and its results are shown in this paper.

A picture of the platform is shown in Fig. 8.

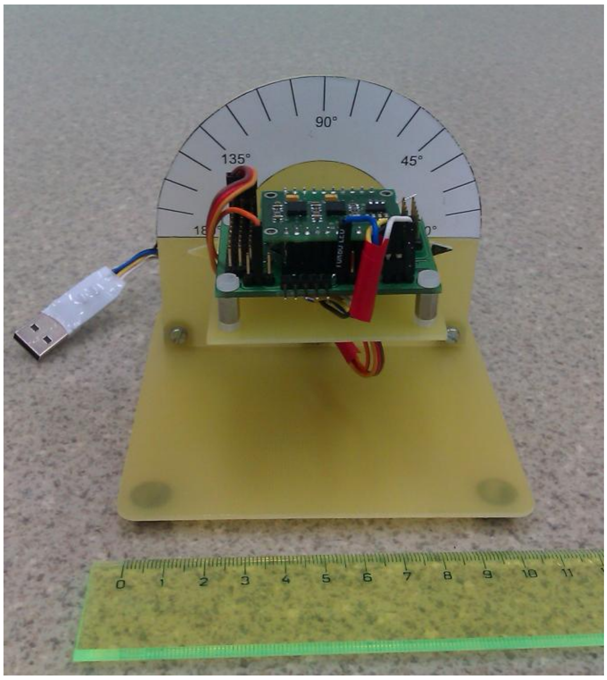


Fig. 8 Picture of the platform

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 acknowledge this support.

REFERENCES

[1] S. Fux, “Inertial Measurement Unit for UAVs and MAVs”, M. Eng. thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, 2008

[2] G.C. Goodwin, S. F. Graebe and M. E. Salgado, *Control System Design*; Upper Saddle River, USA: Prentice Hall, 2001

[3] G. Baldwin, R. Mahony, J Trumpf, T. Hamel and T. Cheviron, “Complementary Filter Design on the Special Euclidean Group SE,” in European Control Conference, Kos, Greece, 2007

[4] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee and M. J. Zyda, “An Extended Kalman Filter for Quaternion-Based Orientation Estimation Using MARG Sensors,” in Proc. International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, 2011, p. 2003 - 2011

[5] “dsPIC33FJ64GP306A data sheet, “ Microchip, USA, http://ww1.microchip.com/downloads/en/DeviceDoc/70593C.pdf

[6] “LPY530AL data sheet,“ ST Microelectronic, http://www.sparkfun.com/datasheets/Sensors/IMU/lpy530al.pdf

[7] “LPR530A data sheet,“ ST Microelectronic http://www.sparkfun.com/datasheets/Sensors/IMU/lpr530al.pdf

[8] “MMA7361 data sheet,” Freescale, USA http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf

Using the Android Platform to control Robots

Stephan Göbel, Ruben Jubeh, Simon-Lennert Raesch and Albert Zündorf
Software Engineering Research Group
Kassel University
Wilhelmshöher Allee 73
34121 Kassel, Germany
Email: [sgoe | jubeh | lrae | zuendorf]@cs.uni-kassel.de

***Abstract*—The Android Mobile Phone Platform by Google becomes more and more popular among software developers, because of its powerful capabilities and open architecture. As its based on the java programming language, its ideal lecture content of specialized computer science courses or applicable to student projects. We think it is a great platform for a robotic system control, as it provides plenty of resources and already integrates a lot of sensors. The java language makes the system very attractive to apply state-of-the-art software engineering techniques, which is our main research topic. The unsolved issue is to make the android device interoperate with the remaining parts of the robot: actuators, specialized sensors and maybe co-processors. In this paper we discuss various connection methods and present a first approach to connect Android with the LEGO Mindstorms NXT robotics system, which we successfully used in our robotics/software engineering courses so far.**

I. Introduction

Android devices are powerful mobile computers with permanent internet connectivity and a rich variety of built-in sensors. More properties make the Android system very applicable for university use: Android uses the Java programming language, which our students are familiar with. Getting started with the Android API is easy; the API is open, i.e. developers can access almost every low-level function and are not sandboxed. In addition, the Android API allows easy access to the hardware components. Interesting for robotics use are the numerous communication interfaces like WiFi, Bluetooth and GSM/UMTS, USB, and the integrated sensors, that is: accelerometer, gyroscope, compass and GPS. Because its a mass product, devices are available for already around 100$, which is much cheaper than any other ARM-based processing unit (e.g. Beagle Board). But the Android platform currently lacks the ability to physically extend it to control more sensors and actuators. This is actually a precondition if we want to use an android device as robotic processing unit, and section VI-A will discuss various options to overcome this restriction.

As we are software engineers, the main focus of our robotic related courses lies in software aspects like model driven software development, code generation, test based development, and strict object orientation. To make algorithms, data structures and software behavior more concrete, we started to create a bridge to real world objects by the use of robotics. Because it's easy to build robots with, we initially used the LEGO Mindstorms RCX and later NXT for our projects and courses. Mindstorms NXT allows to control up to three servo motors and provides a set of useful sensors, which is sufficient for building simple robots like path finders, forklifts etc. From our point of view, another advantage of the NXT system is the availability of a Java Virtual Machine, called *leJOS*. However the leJOS Java (no reflection), the CPU power and the RAM and ROM space (64kb each) provided by the NXT are quite restricted. Due to our experiences, the capabilities of the NXT do not suffice to run complex Java programs with complex runtime data models that want to use for smart system behavior. The LEGO Mindstorms NXT and leJOS will be further discussed in section III.

To overcome the restrictions of LEGO Mindstorms NXT while still using their sensor and actuator control capabilities, we use a two layer approach. The lower layer uses NXT controlled sensors and actuators and the upper layer provides the more complex behavior exploiting the capabilities of an Android device. The two layers are e.g. connected using Bluetooth. The NXT provides connectivity via Bluetooth or USB. As Android provides USB and Bluetooth as well, we use these communication methods to combine Android and the LEGO Mindstorms NXT. This is the key idea of this work, which discusses several connection methods and presents one technical solution to interface the two systems. Benefits are obvious: Android brings in much more processing power, plenty of RAM, integrated sensors, various wireless connectivity and can be easily extended with gigabytes of flash memory. It simplifies programming a lot, for example, processing the live camera image is much easier to implement than within the NXT system. Another argument to use an Android device in combination with the NXT is that sensor integration is much easier. Android sensors are build in, they are already power-optimized which might be beneficial for certain robot types (e.g. in combination with a quadcopter), the whole device is autonomous because of the integrated battery.

In future versions of our robots, we plan to replace the NXT completely for the lower layer by a microcontroller platform, e.g. the popular Arduino board [1]. For the tasks left to the NXT, it is quite expensive, which makes it less attractive for low-cost robot solutions. Thus, we plan to get rid of the Lego Mindstorm NXT brick and just use the NXT motors and LEGO technic construction features for robots, which forms an attractive, low-cost but powerful and extensible basis for

[1] http://www.arduino.cc/

robots. Arduino boards provide open source software solutions to control LEGO sensors and motors. In addition, Arduino boards provide connectivity for many other cheap sensors and actuators. This would lower the cost of a robot even more: a sufficient android device plus the Arduino board costs less than a LEGO Mindstorms NXT brick (but already has sensors integrated and provides much more resources).

Currently, we have a number of LEGO Mindstorm NXT bricks still in our lab. Thus we still use them for basic sensor and actuator control. In this paper we present a technical solution to make Android and Lego Mindstorms NXT interoperable, in form of a software library. This library is called *LPCCA* (LeJOS-PcComms-Android) and is responsible for the connection between the two systems. It provides a powerful, object-oriented API on the Android side, based on the LeJOS API resp. the FujabaNxtLib, see section V. This library is designed to be usable for an Arduino based basic layer, too.

II. Related Work

While we were working on this library, similar approaches showed up. Most of them don't focus on the specific connection between Android and NXT, but rather on Android and microcontrollers in general, or to bring physical, wire bound microcontroller-enabled interfaces to Android. As a microcontroller, most projects refer the popular Arduino platform as defacto standard for a microcontroller-based low-cost platform. It consists of an Atmel AVR CPU with a USB programming interface and a certain layout that allows easy extension of the base circuit board by adding standardized so-called shields (stackable circuit boards) on it. Arduino is programmed in a simplified C/C++ dialect and comes with its own IDE (called Arduino as well). There are different kinds of board designs available, e.g. the Arduino Mega Board comes with a bigger CPU and more I/O ports than the standard Arduino.

A. MicroBridge

The first related approach is a software project called MicroBridge [2] which builds upon the following hardware: an Arduino microcontroller board and a USB Host Shield. The USB Host is required because almost every Android device is a USB slave. The MicroBridge software emulates the host side of the ADB (Android Debug Bridge) protocol. This protocol can be used to transfer arbitrary data between the android device and the host, in this case, the AVR CPU. This project deals only with the low-level connection via USB and ADB and doesn't add any higher-level communication upon it. The developer now has to implement some meaningful communication between the AVR and the Android device to read sensors, control actuators etc by using a virtual TCP connection between host and device. The major advantage of this approach is that it works with almost every android device, even version 1.x. The device itself remains unchanged for successful operation. Just the "ADB debugging" feature has to be activated.

[2]http://code.google.com/p/microbridge/

B. IOIO

The IOIO Board [3] is a direct extension of the Android device and comes as hardware circuit board. The project also provides powerful android software and API. Like the MicroBridge approach, it connects to the device as USB host, but it does not feature a user programmable microcontroller CPU. The onboard PIC CPU has a fixed firmware. The idea is to control the boards IO ports directly via the Android host. Therefore, a powerful Java API is provided. With that API, its easy to directly access the boards general I/O pins or any dedicated communication pins for e.g. SPI, I2C and Serial communication. Internally, it also stacks upon the ADB protocol and a virtual TCP connection. To connect this to NXT components, you would still have to implement for example direct NXT sensor readings as Android host software. But we think the IOIO approach is the most general one to connect any electrically interfaced hardware, actuator or sensor, to Android.

C. Google ADK

Google recently introduced a very similar approach which looks like a combination of MicroBridge and IOIO: The Android Open Accessory Development Kit (ADK) [4] consists on the hardware side of an Arduino Mega with the USB Host Shield integrated. Google also provides an extension shield which adds buttons, a joystick, relays etc. to the base board. ADK also contains a device API, but that one comes only on the newest devices (Android version 2.3.4+).

D. Cellbots

The cellbots.com [5] group, which arose from the 20%-Google-employee-free-time, was the first one to show some interoperability between Android and robot hardware. First experiments used the debug serial port of developer phones to connect to a microcontroller board via serial connection. Meanwhile, there's a wide range of software and hardware development projects available from cellbots. Because the first debug-serial-connection approach was limited to a few phone types, an intermediate workaround idea was to use the headset (and microphone) connector to talk to an external microcontroller. Using a modulated signal, the TRSSTAN kit [6] can control up to four servo motors. Meanwhile, cellbots supports many robot platforms, including Lego Mindstorms NXT, iRobot Create und Roomba, TRRSTAN etc. There is a user-friendly Android Market Application, which uses Bluetooth connections to connect to various robots. It mimics a remote control and can control a NXT without any programming directly by user interactions. A more complex usage scenario requires two android devices, one as user interface, the other one as robot controller, preferably directly attached to the robot. For software developers, Java and Python libraries are available. There's also an App Inventor plugin available which enables cellbots robotic features within the web-based App

[3]http://ytai-mer.blogspot.com/2011/04/meet-ioio-io-for-android.html
[4]http://accessories.android.com/
[5]http://www.cellbots.com/
[6]http://www.cellbots.com/robot-platforms/trrstan/

Inventor Tool, that is used to graphically program or design Android apps with little knowledge of the internal structures.

E. Amarino

The Amarino Toolkit [7] is a library to connect Android devices to Arduino boards. It is limited to bluetooth connections and requires either the Bluetooth variant of Arduino, called ArduinoBT, or a bluetooth extension shield. With the library, Android applications can send data to the Arduino, e.g. their sensor values. As the library doesn't provide any return channel, it doesnt seem applicable for our purposes. Furthermore, being bound to just low-bandwidth, high-latency bluetooth connections might not fit all application examples.

F. leJOS 0.9/Android

Since version 0.9 the leJOS library supports Android directly. This is done by adapting to the Android Bluetooth API, similar to what we did it. But upon that, there's just a single class encapsulating LCP command creation for remote controlling a NXT, so this approach lacks a powerful API, which we think is necessary for more complex robot control applications.

The three USB-related developments, MicroBridge, IOIO and specially Googles own ADK approach, are another reason why we concentrated on the bluetooth interoperability between Android and NXT first. All three are currently under heavy development or have just been announced, so we're waiting until the situation stabilizes.

III. LEGO Mindstorms NXT

The LEGO Mindstorms NXT system has become very popular in schools and universities. It is cheap (compared to more professional robot platforms), it's very flexible because it comes with a wide range of mechanical elements of which you can build a variety of robots, and it comes with an easy-to-understand graphical programming software. LEGO provides a basic set of sensors (ultrasonic range sensor, light sensor, buttons). In addition, there are different kinds of compatible sensors available from 3rd-party manufacturers. The LEGO actuators are continuous servo motors, which are flexible and powerful enough to power a vehicle but also accurate enough to move a robotic arm within the accuracy of a degree. The system features a central component, the LEGO Mindstorms NXT brick, as processing unit. It contains a 32-Bit-ARM7-CPU, a LC-Display, some buttons for user interaction and a rechargeable battery. Via RJ11 wires, up to three motors and four sensors may be directly connected. The brick can communicate via USB or Bluetooth e.g. with a host PC or with other NXTs. Programs can be uploaded via both connection methods. After being programmed, the brick is autonomous. LEGO provides a programming environment called NXT-G. It features a graphical programming language which is sufficient for very simple robotic programs. Because the brick specification was opened by LEGO, there's a wide range of alternative firmwares, development environments and libraries for various languages (C, Assembler, Java, Matlab...). Most of them are open-source. We use the Java firmware implementation, called leJOS [8].

Fig. 1. LEGO Mindstorms XNT

[7]http://www.amarino-toolkit.net/

IV. LeJOS for NXT (NXJ)

With the development of leJOS (Java for LEGO Mindstorms, available as RCX VM for older systems as well for the NXT), it became not only possible to remote control NXT from computers running java but also to deploy native Java programs to the NXT brick and to run them locally. In order for this to work, leJOS consists of several parts. The most important part is the alternative firmware for the NXT. This leJOS firmware is loaded into the bricks flash memory and permanently replaces the original LEGO firmware. The leJOS firmware has been completely rewritten in C and ARM assembler and consists of drivers for the hardware and the Java Virtual Machine (JVM) to run native Java bytecode. The startup-menu is already a native Java program which is capable of loading other Java programs from the systems flash storage into the JVM. The JVM is running one program at a time, which means it is replacing the old one when a new one is loaded. The local programs are stored in the internal storage, with read-only parts of the code being left in the flash storage and read-write parts of the program copied to a heap in the RAM which is watched by a garbage collector. The leJOS API provides the means to write Java programs running on the brick that interact with the sensors and motors directly. The leJOS PC API, while organized very similar, is used for running Java programs on a computer and remotely control the NXT via bluetooth or USB connection, utilizing the LEGO Communications Protocol (LCP) over Java stream connections. The connection to the brick is handled by classes in the `lejos.pc.comm` package, which is only part of the leJOS PC API and not included in the plain leJOS API.

V. FujabaNXTLib

Fujaba [9] is a graphical UML CASE tool based on Story Driven Modeling (SDM) [1] methodology. This is a software development approach where software functionality is specified in so called *Story Diagrams*, among standard UML diagram types like class diagrams. An adaptable code generator generates Java source code out of the graphical diagrams. The methodology supported by Fujaba focuses on strictly object oriented, example- and test driven development. In [2] was shown, that this software development approach gets more intuitive when brought into the real world. A forklift robot was solving the Towers of Hanoi game. This example was mainly used in high schools for first programming education. In [3] we

[8]http://lejos.sourceforge.net/
[9]http://www.fujaba.de/

created an object oriented library called *FujabaNxtLib* upon the leJOS framework, which was modeled within Fujaba Story Diagrams, to control LEGO Mindstorms NXT robots. It adds an abstraction layer upon the leJOS PC API. The generated application code runs on a standard desktop PC and connects to the NXT via bluetooth, so we're remote controlling the NXT. This way, we can create more complex programs as we are not bound to the NXT hardware limits (64kb RAM/ROM). Debugging is possible, we can see and graphically represent the software internals (using a graphical heap visualizer, called eDOBS [4]) and use interactive, stepwise development of the control algorithms, see [5]. This plays benefits especially for educational purposes, because graphical representations are easier to understand and have a more direct reference to real world objects, in this case, our robots and their components. Fujaba in combination with NXT robots for educational purposes has been also used by [6] resp. [7]. The FujabaNxtLib added the aditional abstraction layer for another reason: different adapters, either to leJOS or to virtual sensors and motors, can be plugged in without changing the robot control code at all. This way, we can plug in a simulation layer and run our code independent of an actual robot for testing and simulation purposes.

Figure 2 shows an excerpt of the FujabaNxtLib API. `FNXT` represents the NXT brick. Connected to it by one association per port are the motors and sensors. Every sensors derives from the `FSensor` superclass and implements the observer pattern (not shown in the diagram). The `FMotor` class controls a single motor, which has an integrated rotation encoder. The `FNavigator` class, developed for driving robots with two motors, abstracts the motor control and adds higher-level functions for rotating, turning and driving the robot by delegating to the two driving motors. The navigator can be configured for a certain wheel size, track width and gear reduction, because of that it can move the robot predictably in absolute coordinates. The navigator is the central class for driving control, so we will concentrate on that class in the following sections and don't show the sensor event handling etc.

VI. LPCCA Bluetooth library

In order to establish useful connections between Android and NXT, a library had to be developed. Its purpose is to manage the connection efficiently and provide a developer-friendly way of using the connection, i.e. have some useful API, which led to wrapping the leJOS PC API so it could be used from the Android device. The library is designed as a tool for realizing new application ideas without having to worry about the connectivity and using an already established API. An example of how this library can be used for new applications is the *WebMoteRobot*, which will be presented in section VII. Other application examples, mainly remote controlled robots, are currently under development by this semesters students course.

A. Connection considerations

1) Direct via USB: On the first sight, USB seems to be the preferred connection method between Android and NXT. But the NXT is a USB slave device, and usually every Android device is a slave too. Therefore, USB connections can only established by enforcing USB host mode (or USB OTG) or by adding a USB Host device in-between. The first solution is not supported by the vast majority of Android devices. The latter requires a micro-controller with USB host interface, which introduces new problems: first of all, some adapting bridge code has to be implemented. Then, it's unclear how the intermediate host device gets powered. Furthermore, the different android kernels might not support drivers for serial-over-USB-connections, so the device firmware needs to be modified. The main benefit of using USB would be that a wire-bound connection is quite fast (300kbit/s), has a low latency and is robust against radio interferences. Mainly because of the compatibility issues, we wait for new developments in this area and look for alternative solutions for now. Some Android devices offer direct serial ports on their connectors. This solution wasn't pursued either, as we would limit our solution to certain device types.

2) Indirect via microcontroller: The second possiblity is to use an indirect connection: a microcontroller board, e.g. Arduino, adapts between the NXT and the Android device. This idea was developed mainly because of ongoing talks about moving to a NXT-less robotics system, directly controlling sensors and motors with microcontrollers. It mainly arose as it is an even cheaper approach than the Lego Mindstorms system. However for easier and faster results it can also be used with the NXT still in place basically as a batterypack for the motors and easy connectibility. On the Android side we either have to use USB as the only wired connection method, or any wireless method, e.g. bluetooth. The connection between the NXT and the microcontroller board is usually established via the I2C protocol, attached to one of the NXT sensor ports. Having a microcontroller board connected to the Android device allows a wide range of flexible configurations, e.g. leave the NXT out and connect sensors and actuators directly to the micro-controller, or using it to multiplex between different NXT at once. This requires a lot of software implementation on all three system components. But this setup is too complex and more expensive than the direct connection between Android and NXT and should only be considered as a move towards NXT-less systems.

3) Bluetooth: The third possibility for interconnectivity is Bluetooth. Bluetooth based wireless connections are basically supported by all shipped Android devices today. All NXT firmwares support remote Bluetooth control via LCP (Lego Communications Protocol). This combination is the one big advantage over any other possible connection. leJOS already supports translating commands from its object oriented library to LCP transmission protocol commands and encapsulates for sending it over a SPP-Bluetooth connection. Just one class of the leJOS library, which is responsible for adapting to the

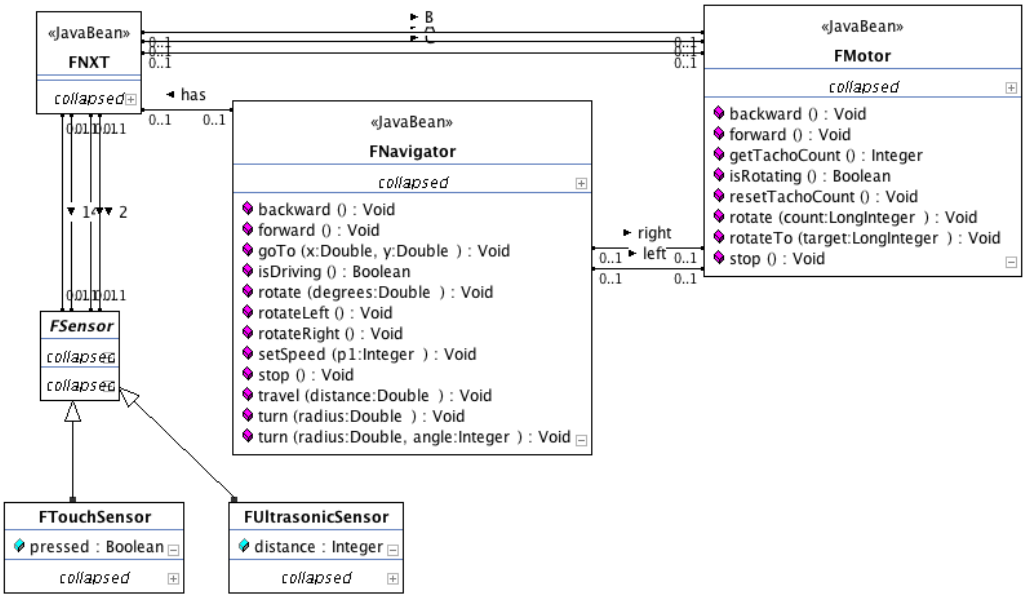


Fig. 2. FujabaNXTLib Main Classes

concrete systems bluetooth stack, had to be modified to get the LCP over Bluetooth-code working on Android. The main disadvantage of using a bluetooth connection is the high and jittering latency, limited throughput and sometimes stability issues. As most of our application scenarios are not really time critical, this is acceptable. On the other hand, using a wireless connection doesn't require the Android device being close or wired to the NXT, so it can be instantly used as remote control. As we did only minimal modifications to the leJOS implementation, upgrading to new versions seems easy. Furthermore, we are not bound to NXT with the leJOS firmware: as LCP is used by the standard LEGO firmware and others, we can control even the standard LEGO NXT firmware with this approach.

B. Implementation

The library in its current implementation consists of a RemoteService called `LPCCARemoteService`. A remote service is an Android system wide software interface. Once started, it encapsulates and controls the connection to the NXT system-wide for all Android applications. This makes it possible to establish a connection in one Android application and actually use it in another one. The *WebMoteRobot* for example could be set up by someone using the Android device and an on-device application to establish the connection, which is then used by a web application running on the device also, but that is controlled remotely by another users browser. Such a setup is presented in section VII. An Activity (corresponds to an Android User Interface Screen) that can be started by the RemoteService provides a simple means of setting up the connection to the NXT, basically providing a list of all bluetooth-enabled NXT in discovery range. Once a decision for bluetooth pairing and connection was made, the Activity returns to the application that asked the RemoteService to start the Activity, enabling it to now control the NXT via the leJOS PC API. The use of this Connect-Activity is completely optional, each application can choose to establish the connection itself, asking the RemoteService for a list of available NXT, and telling it which one to connect to. To determine which NXT shall be used, it can either have its own implementation of a visual selection for the user, or just have some hard coded naming scheme, always connecting to the same NXT.

`LPCCARemoteService` extends the Android API class Service. It implements `ILPCCARemoteService` which is defined via an AIDL file. AIDL stands for *Android Interface Definition Language* and allows you to define the programming interface between separated Android processes, i.e. our service, and the client, newly developed applications, agree upon.

In order to provide a list of NXT only devices, all discovered devices are checked upon their mac address, filtering those that specify LEGO as the manufacturer, as the NXT is the only device with bluetooth build by LEGO. In order to notify applications using the library of newly discovered NXT the LPCCARemoteService sends its own broadcasts so interested applications can subscribe via `BroadcastReceiver` objects. This is useful if an application is started before the NXT is turned on and discoverable, because the NXT won't be in the initially transferred list of available devices, but rather show up as a broadcast.

```
package org.lpcca.service;
import org.lpcca.service.*;
interface ILPCCARemoteService {
```

```
    List<String> getAvailableDevices();
    void requestConnectionToNXT();
    void establishBTConnection
        (String deviceKey);
    void requestDiscovery();
    boolean isConnected();
    Navigator getNavigator();
    NXT getNXT();
    Motor getMotor(char port);
}
```

The first four methods are used to initiate the bluetooth connection to the NXT. When being connected, the getter-methods can be used to retrieve a proxy object that represents the NXT, a motor or the navigator. In detail, the `getNavigator()::Navigator` method returns a navigator instance, which is defined as AIDL interface as well, which follows in the next listing:

```
package org.lpcca.service;
interface Navigator {
    void forward();
    void rotateLeft();
    void rotateRight();
    void stop();
    void backward();
    void travel(double distance);
    void turn(double radius, int angle);
    void rotate(int degrees);
}
```

By declaring the complete *FujabaNxtLib* API layer via AIDL, it is accessible to Android applications, which can use the exposed proxy objects and don't have to deal with the low level LCP protocol etc. Furthermore, this object oriented approach makes it easy to introduce non NXT components externally connected later on, or to multiplex and support multiple NXT connections at once.

An exemplary sequence for connecting an application using the LPCCA library and connecting to a NXT is shown in figure 3. All communication between the application, the remote service and the bluetooth subsystem is inter-process communication and therefore done via Android intents.

Fig. 4. WebMoteRobot

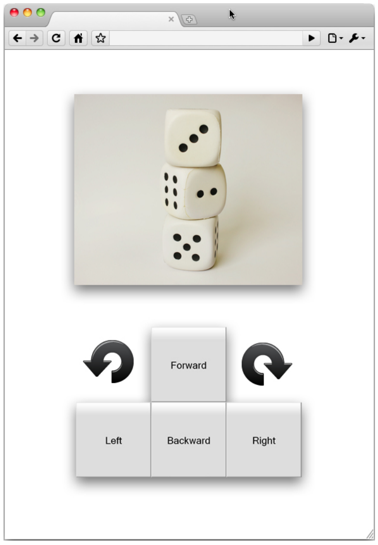


Fig. 5. Web Interface

VII. WebMoteRobot

As a proof of concept we have developed a small application that makes use of the provided functionality by the Android API. The application can be accessed via web and provides a videostream from the Android camera as well as an interface with buttons to control the robot. Using the camera and the wireless connection to access the web makes the Android device act as sensor and actuator. There are no NXT extensions that provide this functionality. The robot is to be build with the Android device mounted in place, preferably rotated around the cameras axis by a third motor. The first two motors are used for steering and movement. If setup correctly, and with the Android device connected to the internet it is now possible to visually explore the surroundings of the robot from any browser that has access to the internet. We called our application WebMoteRobot. It is implemented using the Google Web Toolkit (GWT) [10]. The web application is deployed on a Java web server called iJetty that is running on the Android device. The user can access the webapp via the IP address of the Android device and control the NXT with the provided buttons that are linked to the corresponding navigator methods of the LPCCA library.

Once a client connects to WebMoteRobot the server tries to establish a connection to the LPCCARemoteService, i.e. binding to the service. If the service was not started yet, it will be started now. Upon a successful connection it has access to

[10]http://code.google.com/webtoolkit/

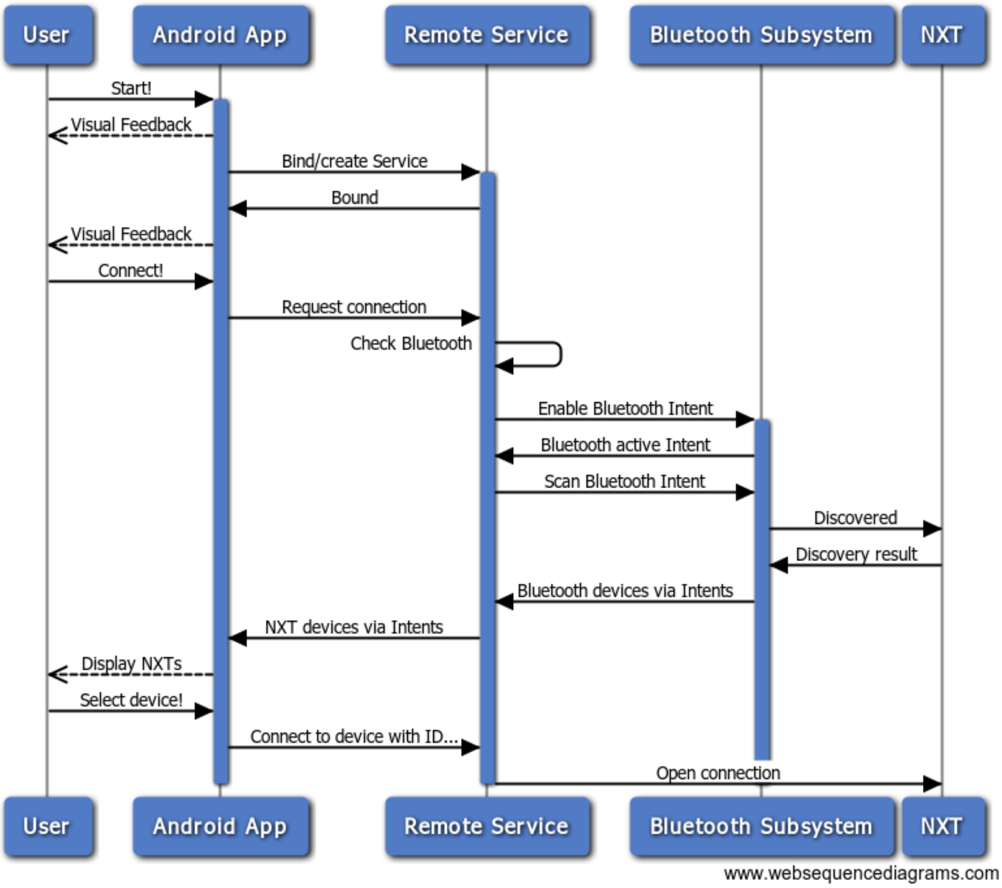


Fig. 3. App startup

the NXT via the navigator and maps the buttons actions to the respective methods. To achieve this the functionality has once more to be wrapped as the clicks on the buttons are handled at client-side (the users browser), and connect to server-side (the Android device) via Remote Procedure Calls (RPC) that are defined as a GWT service. It not only wraps functionality but implements some further logic that makes using the web interface more simple, e.g. checking whether a connection has already been set up and in case it has not, calling the setup from the LPCCARemoteService. This makes using the web interface consistent, because it doesn't affect the user whether a connection is still established by a previous session or if it is the first session. The whole interaction of the browser/client side, the web application on the device and the library service is shown in figure 6. With the Android device connected to the robot, changes in position and angle are visually fed back to the user via the web interfaces videostream. For easier navigation and to prevent unwanted movement, the buttons in the web interface move the robot by a predefined distance each click. This helps reducing latency induced errors that occur due to the lag between movement of the robot to actual reaction of the videostream in the web interface. It is not only the Bluetooth connection but mainly the wireless streaming of video that increase this lag. Another approach might have been reducing the constant moving speed, leading to problems if the internet connection is terminated, which is why we chose the predefined distances. It was possible to navigate through all of the rooms of the Software Engineering Research Group in Kassel University.

VIII. Conclusion

The LPCCA library introduced in this paper is fully functional, enabling developers to write Android applications that remote control NXT robots. Using the library, the Android may programmatically determine complex control sequences for the NXT. The LPCCA library has been successfully introduced in our course Distributed Robotic Systems Modeling at Kassel University this term. It proved to provide simple access to standard LEGO robot sensors and actuators. Based on the LPCCA library it was easy to build complex Android applications that use object oriented data models in Java. The LPCCA library may however still be expanded beyond the actual state, making it more flexible and easier to configure.

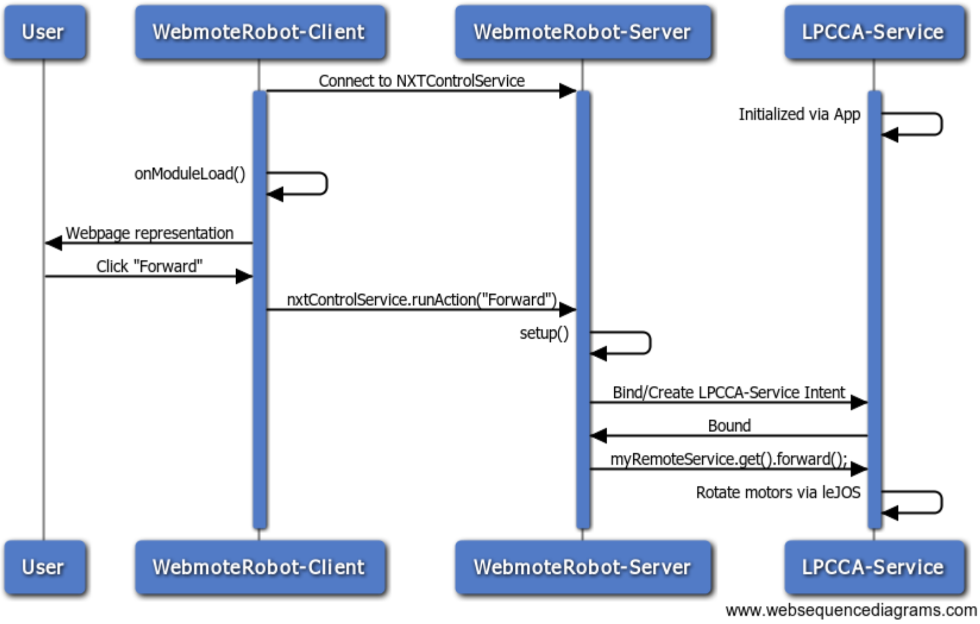


Fig. 6. WebMoteRobot dispatching

Keep in mind that the library as of the writing of this paper is still a proof of concept and needs project specific adjustments to do more than basic controlling of the NXT. The following areas have been identified as being the most interesting ones to develop further:

- An Android controlled Segway (inverted pendulum principle). For real world results concerning the bluetooth induced latency, an Android controlled Segway would be very interesting to develop. The Android devices come with just the right sensors for a Segway, as it needs exact positioning data of the device, i.e. angle and orientation data provided by the accelerometer, gyroscope and compass sensors in the Android device. If the latency induced by Bluetooth prevents the Segway from staying upright, this would again motivate to go for USB based connections between Android and NXT or for an Arduino based solution.
- A configurator providing visual means of configuring the setup of the NXT, i.e. the usage of ports. This would increase the user experience as any changes to the wiring of the NXT can be adapted without changing the source code of the program, as long as the right sensors and motors are connected. The need for hard-coded port usage would be eliminated. This functionality should be provided by an Activity class in the library itself so it can be reused for all projects utilizing the library.
- USB support has yet to be implemented. Even though there is only a small subset of Android devices that are capable of this means of connection, it is an enhancement to the library, as some latency-related problems can be circumvented.
- Several NXT controlled by one device. The leJOS API is making heavy use of singletons in its current implementation, leading to a situation where only one NXT can be controlled at a time. However it is in general possible to use the bluetooth connection to control multiple NXT at the same time.

Altogether, the new LPCCA library provides a simple means for using Android devices to remote control LEGO Mindstorms NXT bricks and thus to bring complex computation capabilities to LEGO based robots. In addition, the Android devices add a lot of new sensors to LEGO robots like GPS, acceleration, and last but not least a camera and WIFI connectivity. And students love it.

REFERENCES

[1] A. Zündorf, "Story driven modeling: a practical guide to model driven software development." in *ICSE*, G.-C. Roman, W. G. Griswold, and B. Nuseibeh, Eds. ACM, 2005, pp. 714–715.
[2] I. Diethelm, L. Geiger, A. Zündorf, "Fujaba goes Mindstorms," *Objektorientierte Modellierung zum Anfassen; in Informatik und Schule (INFOS) 2003, Mnchen, Germany*, Sep. 2003.
[3] R. Jubeh, "Simple robotics with Fujaba," in *Fujaba Days*. Technische Universität Dresden, Sep. 2008.
[4] L. Geiger and A. Zndorf, "eDOBS - Graphical Debugging for Eclipse," Natal, Brasil, September 2006.
[5] Leif Geiger, "Design Level Debugging mit Fujaba," in *Informatiktage*. Bad Schussenried, Germany: der Gesellschaft für Informatik, 2002.
[6] P. Van Gorp, R. Mans, and J. Samalikova, "Help Wall•E door de vuilnisberg heen! handleiding voor publieksdag project," Sep 2009.
[7] B. Grusie, P. V. Gorp, R. Jubeh, and A. Keller, "Fujaba hits the wall(-e)," in *Proc. Fujaba Days 2009*, P. V. Gorp, Ed., 2009, pp. 34–39.

The Kinect Sensor in Robotics Education

Michal Tölgyessy[#1], Peter Hubinský[*2]

[#]*Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology,*
Slovak University of Technology in Bratislava
Ilkovičova 3, Bratislava, Slovakia
[1]michal.tolgyessy@stuba.sk
[2]peter.hubinsky@stuba.sk

***Abstract*— This paper deals with education in the field of robot sensing abilities. It briefly introduces the commonly known and used concepts and sensors, but focuses mainly on the recent Kinect sensor. Technical information and background on the Kinect are provided. The last part of the article deals with possible applications of the sensor in various robotic fields with emphasis on the educational process.**

Keywords— sensor, Kinect, depth map, camera, 3D modelling

I. Introduction

One of the basic attributes of a robot is its ability to interact with the world around it. Students have to learn that activities such as sensing and interpreting the surroundings, which are absolutely intuitive for a human, are very complicated tasks in the case of robotic systems. In order to understand how robots accomplish these tasks students have to get acquainted with sensors. The most commonly used sensors in mobile robotics are the active ones, such as infrared, ultrasonic and laser sensors [1]. They help the robot to understand the environment providing it with depth information which can be used to avoid collisions, bypass obstacles or to create more or less sophisticated maps. Students usually have to learn to understand the basic functionality of these sensors, they need to be able to analyse the analogue and digital outputs and interpret them in order to accomplish given tasks. For example a line following robot is usually equipped with three or more infrared sensors that are used to detect a black or white line placed on the floor underneath a simple wheeled robot. After a correct algorithm implementation the robot should be able to properly process the sensor information to follow the line. By implementing algorithms and accomplishing tasks such as the one mentioned students gradually develop their understanding of the interaction between robot and its local environment. Over time they fully grasp the fact that it is a large set of discrete operations that have to be implemented on real-time digital systems. What we consider a higher level in the educational process is using a digital RGB camera as a sensing unit. The use of camera introduces the vast technical field of visual systems, where an M x N matrix of pixels represents the reality around the robot as a 2D image. In order to use a camera as a sensor effectively students have to learn to extract features from images. This includes image pre-processing, segmentation, edge detection, blob detection, object recognition and other operations [2].

The purpose of this article is to present an even more sophisticated sensor that combines the advantages of standard distance sensors and RGB cameras. The Kinect sensor originally developed by PrimeSense and Microsoft for the Xbox 360 gaming console has been hacked and is being used in many hobby and robotic applications. There already is a commercially available robotic platform called Bilibot, which is based on the iRobot Create platform and uses Kinect as its main sensor [3].

Fig. 1 The Bilibot platform [3]

Authors of this article believe that complex depth sensing combined with RGB sensing will become a trend in robotics and Human-computer interaction (HCI) in general. A 3D visual sensor simplifies many common perception tasks and can be a powerful tool in education because working with directly mapped depth data on particular image pixels is much more intuitive than working with either pure RGB data or pure depth data.

II. The Kinect Sensor

The Kinect entered the global market in the beginning of November 2010 and was immediately a huge success selling more than one million products in ten days. To access the Kinect data with non-proprietary software the USB communication had to be reverse-engineered. Using data grabbed by a USB analyser the Spanish hacker, Héctor Martín, was the first one to access and display the Kinect RGB and depth data on a PC using the Linux operating system with OpenGL and OpenCV libraries [4].

The basic parts of the Kinect are (Fig. 2):

- RGB camera
- 3D depth sensing system
- Multi-array microphone
- Motorized tilt

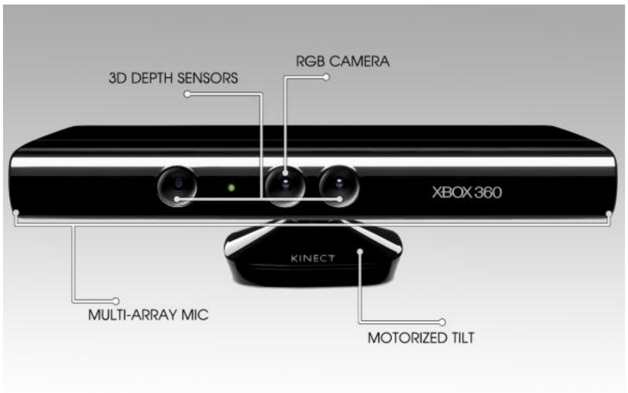


Fig. 2 The Kinect sensor

Kinect is able to capture the surrounding world in 3D by combining the information from depth sensors and a standard RGB camera. The result of this combination is an RGBD image with 640x480 resolution, where each pixel is assigned a color information and a depth information (however some depth map pixels do not contain data, so the depth map is never complete). In ideal conditions the resolution of the depth information can be as high as 3 mm [3], using 11 bit resolution. Kinect works with the frequency 30 Hz for both RGB and depth cameras. On the left side of the Kinect is a laser infrared light source that generates electromagnetic waves with the frequency of 830 nm. Information is encoded in light patterns that are being deformed as the light reflects from objects in front of the Kinect. Based on these deformations captured by the sensor on the right side of RGB camera a depth map is created. According to PrimeSense this is not the time-of-flight method used in other 3D cameras [5].

III. Basic Advantages

The first and major advantage of Kinect in robotics and the educational process is its impact on image segmentation tasks. The simplification comes from the depth data. With a single camera it is impossible to distinguish objects of similar colors. For instance, if a white box stands 1 meter in front of a white wall a robot with a camera is not able to find differences between these two objects from the RGB data. However, with the Kinect providing a 3D map the segmentation is very simple using just a single distance threshold. Without its application, there is a lot of noise and unwanted objects in the image (Fig. 3). After its application the desired objects can be easily segmented (Fig. 4).

Fig. 3 Depth data without a threshold

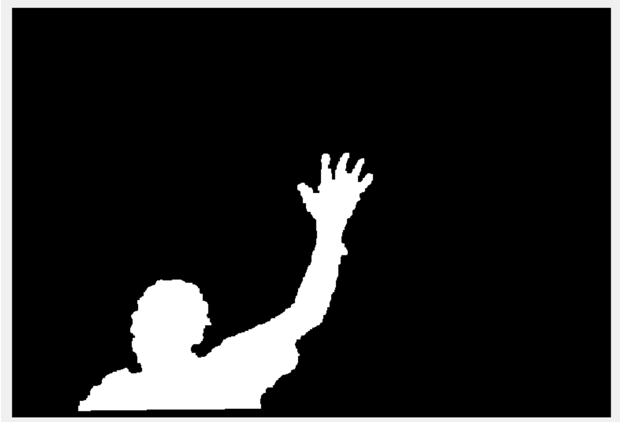


Fig. 4 Depth data with the threshold applied

Using this thresholding method is great for the educational process, because students can immediately see the results of the operations applied, whereas with IR or ultrasonic sensors they see only numerical output provided by the microcontroller via serial interface and have to interpret the data themselves. With 3D data combined with color RGB data a lot of this interpretation is intuitive and automatic.

IV. Applications

The Kinect sensor can be used in a variety of robotics applications being an addition to older methods or a complete substitution. Students can compare the pros and cons of several approaches and find out first-hand what suites their particular project more.

A. Data Fusion

Students can learn to fuse different data. The RGB information can be converted to any commonly used color space, such as Normalized RGB, HIS, HSV, HSL, TSL, YCbCr, CIELAB or CIELUV. All of these color spaces are commonly used for different tasks in visual systems applications. One of their primary utilizations is object detection and segmentation, which is also used in computer and robotic vision. Hence, by using Kinect as a sensor for a robotic system students can learn to fuse depth information with different color space data.

B. Obstacle Avoidance And Collision Detection

Providing a depth map with good resolution the Kinect can be used for collision detection and obstacle avoidance. By applying safety thresholds the robot can be informed any time an unknown object crosses a defined distance. Based on this information it can interrupt its current activity and stop motion or change the direction of its movement. For this the depth map alone is good enough, however for more sophisticated tasks the combination with RGB image is beneficial. Robots can be trained to behave differently when encountering different type of obstacles. When there are several obstacles with same color characteristics, such as tree trunks (brown), concrete objects (gray) or water (blue), the robot will be able to gather more information and classify each object with more detail. Thus, more sophisticated decision trees can be implemented resulting in greater and more reliable autonomy.

C. Object Recognition

When accomplishing autonomous tasks robots might need to recognize certain known objects, such as inscriptions, signs, faces, holes or cracks. All these objects can be primarily recognized by visual systems methods, but the depth information comes in as a very helpful addition. It can help in determining the vertical position of objects, informing the robot wether the recognized object is above, beneath or on the ground level. It can be hard to find cracks and holes from RGB image, but is much easier to do so with depth information. Combining visual algorithms with depth algorithms is a beneficial fusion in object recognition tasks.

D. Gesture Control

A part of the Human-robot interaction (HRI) is controlling a robotic system with hand gestures and body poses. It has been the subject of many research works. Interacting with digital systems without the need of a mouse, keyboard or joystick is the future of modern households. Many algorithms using only a single RGB camera have been proposed. However, the 3D information the Kinect provides is a major help in accomplishing both static and dynamic gesture recognition algorithms. Segmentation and skin detection is an important step in finding hands and is greatly simplified with depth information.

E. Localization And Navigation

One of the basic goals of autonomous robots is their ability to localize themselves and successfully navigate to a defined destination. The depth information provided by the Kinect can be of great help in map creation and localization. If the global map used by the robot contains color information, the interactive online color object recognition can be used to enhance the localization precision.

Kinect can also be used to implement visual odometry. With bare 2D color information visual odometry is practically impossible. Combining the color image with depth map opens new possibilities to odometry applications. Students can compare for instance incremental sensor outputs with the visual-depth odometry outputs. The incremental sensor data corruption caused by wheel sliding can be corrected by reliably designed odometry algorithms based on the Kinect outputs.

F. 3D Modelling

The depth data acquired from the Kinect can be used to create a 3D map of the environment, however adding the color data allows the creation of a complex color 3D model. If a reliable algorithm is implemented the robot can add static local 3D images together as it moves. The result of this is a 3D color model of a corridor (Fig. 5, Fig. 6). If such corridors are correctly attached to each other a complex 3D map of a local environment can be gradually created.

Fig. 5 3D model of a corridor (A)

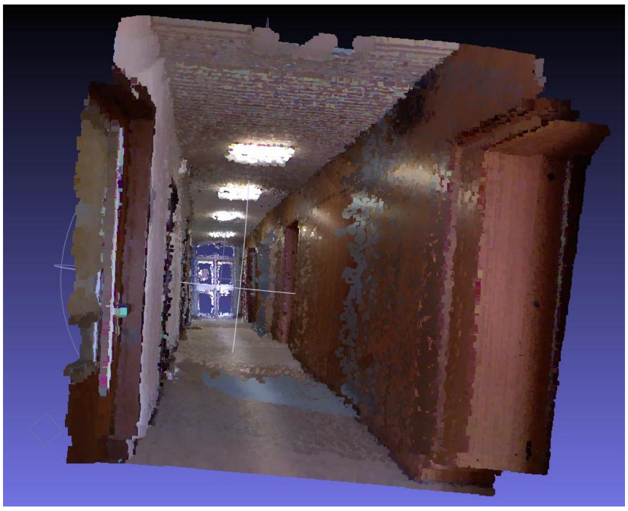


Fig. 6 3D model of a corridor (B)

V. Practical proposals

The technical parameters and capabilities of the Kinect are one thing, but its practical application in classes for students is another. Our ideas are forced to be merely theoretical because the Kinect is a very recent sensor. The official Microsoft SDK was released only in June 2011, therefore its inclusion to the educational process is limited by lack of time, despite the existence of many hobby applications. A hobby application requires a functional or semi-functional result with no documentation. On the other hand a University course requires a fully functional platform with an environment and software interface suitable for a particular group of students with certain experience level.

We decided to divide the Kinect's educational application into two areas. The first one focuses on standard practical classes lasting several weeks that usually support lectures, the second one consists of large projects, such as bachelor's and diploma projects.

Of course, the role of the Kinect sensor as a part of a class depends on the subject taught. Generally, we think that it's best to start teaching the least sophisticated algorithms and hardware and only then finish the course with a complex sensor, such as the Kinect. This way the students will learn to use the standard ways, but will also realize what benefits can the Kinect bring comparing to what they already know. We think that the "comparison moment" is very important. Students won't just read and learn what 3D sensing brings to the subject they study, but they will experience it. Through this experience they will creatively come up with their own ideas and solutions.

The second area mentioned includes large projects where Kinect is the central sensor. We advise mounting it on mobile platforms, such as hexapods, wheeled mobile robots or quadrocopters. Student can then pick a task, design and implement a method or algorithm under the supervision of his or her teacher.

VI. Conclusions

The presented sensor is a great and complex tool that can be used to teach students many common robotic tasks, ranging from the easiest (collision detection) to the most complex (3D mapping). As 3D sensing is a current research trend, studying robotics with Kinect as a part of the sensor equipment is both challenging and motivating. Students can compare the results obtained from depth and color algorithms with the more common methods that use infrared, ultrasonic or laser sensors. We also believe that working with a hacked proprietary sensor sold by a major company can increase their motivation.

Acknowledgment

This work was supported by MS SR under contract VMSP-P-0004-09 and VEGA 1/0177/11.

References

[1] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Massachusetts Institute of Technology, The MIT Press, 2004.

[2] M. Šonka, V. Hlaváč and |R. Boyle, *Image Processing, Analysis and Machine Vision*, Pacific Grove : Brooks/Cole Publishing company, 1999.

[3] The Bilibot website. [Online]. Available: http://ww.bilibot.com

[4] We Have a Winner. [Online]. Available: http://www.adafruit.com/blog/2010/11/10/we-have-a-winner-open-kinect-drivers-released-winner-will-use-3k-for-more-hacking-plus-an-additional-2k-goes-to-the-eff/

[5] Kinect: The company behind the tech explains how it works. [Online]. Available: http://www.joystiq.com/2010/06/19/kinect-how-it-works-from-the-company-behind-the-tech/

Wheeled Mobile Autonomous Robot for Eurobot 2011

Raimund Edlinger#1, Michael Zauner*2, Walter Rokitansky*3

#*FH OOE Forschungs & Entwicklungs GmbH*
Stelzhamerstr. 23, 4600 Wels, Austria
[1]raimund.edlinger@fh-wels.at

**FH OOE Studienbetriebs GmbH*
Stelzhamerstr. 23, 4600 Wels, Austria
[2]michael.zauner@fh-wels.at
[3]walter.rokitansky@fh-wels.at

Abstract— **Vision and sensor systems and an efficient motion system are fundamental components in the process of designing and building autonomous robots.**
This paper describes the technologies behind a "chess robot", built to collect pawns, queens and kings and to separate them into on specified squares.
The robot, designed to participate to the International competition of robotics "Eurobot 2011", is able to distinguish the objects according to their shape and color.

Keywords— **wheeled mobile robot, WMR, Eurobot, differential drive, computer vision**

I. Introduction

A self-navigating fully autonomous robot with the capabilities of element searching and collision avoidance would be an ideal platform for robotic researchers and students to develop robots for the competition EUROBOT. One of the important processes involved in the design of a robot is the evaluation of the concept of the robot because the Eurobot association defines a new theme each year. This paper describes the development of an autonomous robot for this contest. It describes the experience with the design, the evaluation of sensors and grasping systems and the implementation of a vision system which communicates via RS232 with the main board.

II. Eurobot Competition

A. Eurobot in general

Eurobot is an international robotics contest which involves students, researchers and amateurs from all over the world.
Created in 1998 as the "French Cup of Robotics", in 2006 350 teams from 26 countries took part in the competition. Organized in two phases (national qualifications and the international final), the competition consists of a real tournament in which the robots duel in "one-on-one challenges". At Eurobot finale, the first 16 teams from the qualifying phase are selected for the final round.
Every year, a different robotic challenge really with a newly defined set of playing rules is established. Robots must be absolutely autonomous and any kind of communication with the robots (either wired or wireless) during the matches is forbidden. Robots are limited, in size to an area of 120cm and a height of 43cm and they must implement an obstacle avoidance system.

B. Eurobot 2011

This year the Eurobot association decided to play a special kind of chess. The robots have to collect the pawns, queens and kings which are detected by a system of bar-codes. In the game it is allowed to stack a maximum of two pawns and a king or queen to have more points. The goal is to have more points on squares of our playing colour as the opponent after 90 seconds, see Fig. 1. A color (red or blue) and therefore a side of the playing area is allocated to the team before each match. When both teams and the referees indicate they are ready, the referee will determine the random positions for the playing elements to be placed on the table. This is done by drawing from a set of cards.

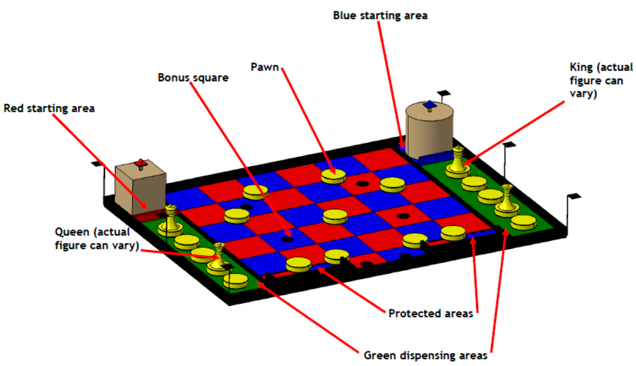


Fig. 1 Playing Area

III. Robot Development and Experiments

A. Drive mechanism

The drive mechanism is in the base of the chassis and is equipped with a differential drive. Two brushed DC-motors with a planetary gear and a resistance R_A, machine constant k_m and gear ratio n are used to move the robot. The drive motors also contain a gearbox with a gear transmission ratio of 14:1 and a magnetic encoder for the speed and positioning control (shown in Fig. 2).

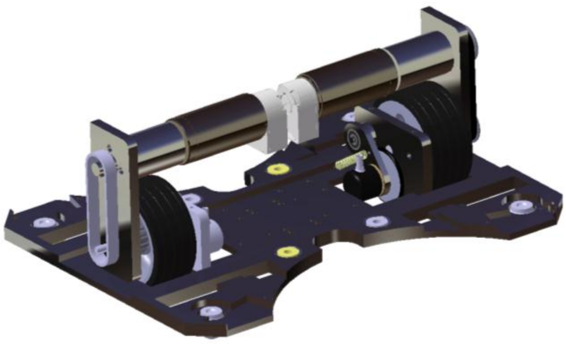


Fig. 2 Motion Control Unit

The posture $[x, y, \theta]$ of the wheeled mobile robot (WMR) is given with the centre of wheels axis *(CoA)* as reference point, see Fig. 3. The kinematic model of the unicycle type system is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

with control input $u^T = [v, \omega]$. We assume ideal velocity control of the inner loop and therefore the velocities may be considered as inputs [1].

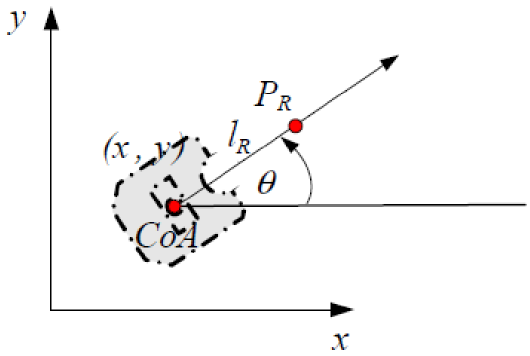


Fig. 3 The coordinate system

An Atmel ATxmega256 controller [7] on the main board is implemented to guarantee the exact execution of the movement and to calculate the trajectory.

B. Electronic and power supply

The power supply of the robot consists of two NiMH accumulators. These accumulators are connected to the supply-board which consists of 3.3V, 5V (logic), two 5.5V (power), 12V and 24V TRACO voltage converters. On the main board there are implemented three Xmega 256 microcontrollers from Atmel [2]. The main board, based on a modular electronic system [5], includes all of the intelligence needed for reading sensor information, controlling the movement of the robot and accessing the robots actuators (see Fig. 4). To reduce wiring complexity each input/output port has its own power supply for the sensor/actuator that is connected to it. The main processor distributes the tasks to the slave processors. It is the central unit which communicates with the slaves via serial interface (RS422). The main processor executes the main program which includes all the strategies, the timing, the route planning, etc.

The slave processors control three DC motors and four servos which control the speed, the position and the odometric navigation system.

A Human-Machine Interface (HMI board) is also connected to the main board. The HMI board allows the parameters to be easily changed and show error messages from the main board, moreover it provides easy access to the program and JTAG ports that are looped through to the main board.

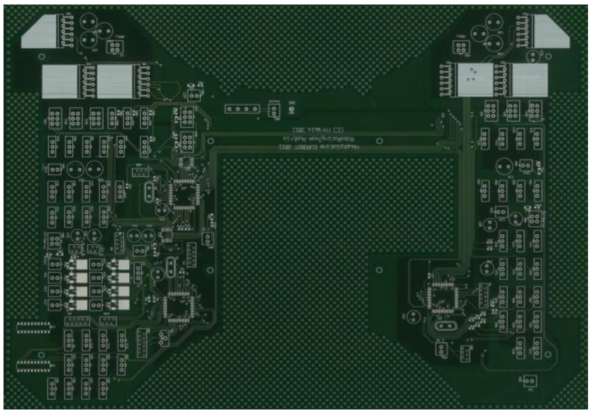


Fig. 4 Main board control

C. Element Management System

At the front and at the back of the robot are two grasping systems which are used to collect the royals and pawns (see Fig. 5). On each side of the robot there are two arms to collect pawns which have two infrared sensors for detecting the elements. Additionally on each arm there is a vacuum grasping system with an individual vacuum pump so that each arm can operate independently from the other. The elements are stored in special receptacles by the grasping system, enabling the transport of a maximum of six pawns and two royals (Fig. 5).

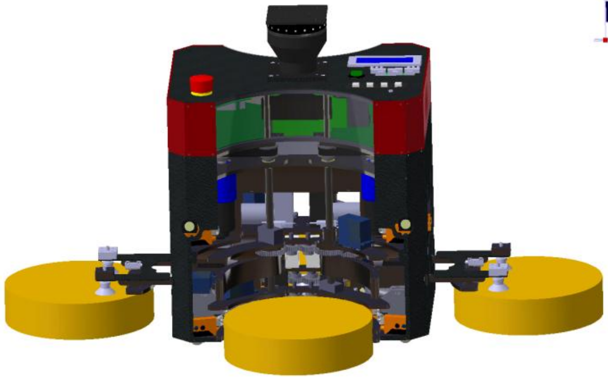


Fig. 5 Autonomous Robot for Eurobot 2011

D. Avoidance System

Eight ultrasonic sensors [3] are located around the robot and are required for the obstacle avoidance system. The sensors have a maximum operating distance of 100 cm. That way the robot can detect obstacles within a perimeter of over 2 m. Located on top of the robot is the opposing robot detection unit. This unit determines the location of the opponent robot. Due to this fact the robot can detect the position of the opposing robot at any time and thereby avoid collision. The robot tries to create a profile of the opposing robots' movement so that it is able to calculate the best way to go around it

E. Vision System

To plan an optimized path for the robot, there are two cameras which detect the playing elements. The cameras act as colour sensors and detect the mean colour of a predefined position. Fig. 6 shows the close range of the front camera. The classification of this section allows the determination of all royals and pawns. The information is transferred via serial interface to the main control board.

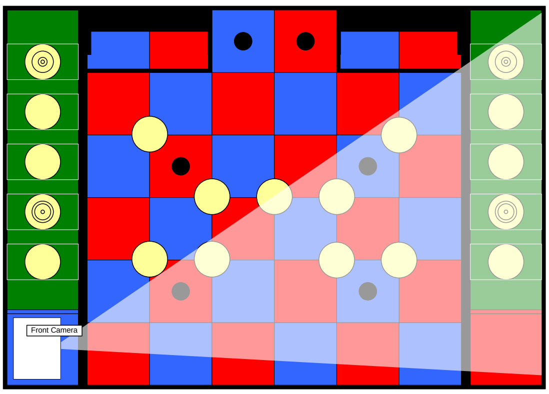


Fig. 6 Close range of the cameras

The way in which the vision system is set up depends on the imaging environment and the type of analysis and processing required. Based on the limitation of size of the robot the vision system consists of two webcams and a pico ITX to produce enough quality to extract information from the acquired images.

To ensure the full detection of the elements before starting the game the resolution of the camera have to change to 1024x576 pixels. At the beginning of the match, the robot must be placed fully with in the starting zones and it is important that the robot always has the same starting position *[x,y]* set to the reference position zero.

The positions of the front and back camera are fixed and acquire images of the objects from an angle with perspective errors. It is necessary to calibrate the system to assign real-world coordinates to pixel coordinates and compensate for perspective and nonlinear errors inherent in the imaging system.

To detect the yellow elements, region of interests (ROI) are used for pulling out the useful colour information in an image, see Fig. 7. The colour of a surface depends on the direction of illumination and the direction from which the surface of the objects is observed.

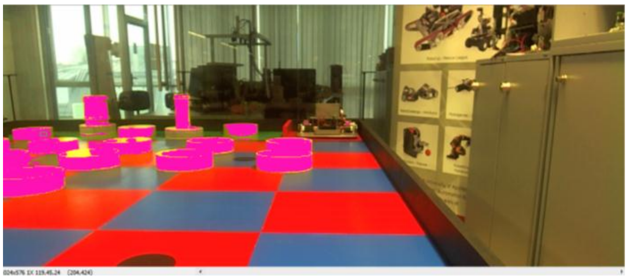


Fig. 7 Detection of the elements

The created ROIs are useful to focus the images analysis and to guarantee the full recognition of the constellation of pawns, queens and kings. After the start-up procedure the particle measurements of calibrated real-world units determine the location of particles and their shape features.

The measurements are based on characteristic features of the object represented in the image, the real-time detection of which is a processor-intensive task. Using 1GHz processor the performance of the machine vision system is limited to four frames/seconds. Nevertheless this limitation allows for scanning of the elements during the game.

Before and during the game the vision system sends and receives the following strings:

COMMUNICATION VIA SERIAL INTERFACE

Nr	**Messages between main board and vision system**		
	Read String	**Write string**	**Action**
0	#C*	#CB* #CR*	Playing Colour – Blue Playing Colour – Red
1	#A*	#APQPPKXPX PXXPPXX*	Constellation of the elements (see Fig. 6)
2	#M*	#Mxy*	Mapping the elements and send the x and y-coordinates
3	#Pxy*		Current position [x,y] of the robot on the table
3	#S*		Standby
4	#Z*		End

IV. CONCLUSIONS AND OUTLOOK

Our experiments demonstrate successfully that the management between vision and main board is very robust against unexpected faults in execution and the sensing of elements in the playing area. The generation of mapping the elements is dependent on the performance of the pico ITX and particle analysis in finding statistical information – such as the area, location and presence of particles. With this information we have performed many machine vision inspection tasks. The robustness of the measurement relies on the stability of the image acquisition conditions, sensor resolution, lighting, and vibration.

Future research will entail optimization of the machine vision and the implementation of the search and rescue system, which the robots will use to explore and navigate the generated map.

ACKNOWLEDGMENT

The authors are indebted to Gernot Grabmair for the support of the model-based nonlinear control for two-wheeled mobile robots (WMR). The authors grateful acknowledge the support of the Upper Austria University of Applied Sciences, National Instruments, IFM, IGUS and SMC for setting up the Robo Racing Team [4].

REFERENCES

[1] G. Grabmair, “Model-based Nonlinear Control of 2-WMR”, 4th International Conference on Research and Education in Robotics,: Springer-Verlag CCIS-series, 2011.

[2] AVR-XMega (2011) homepage ATMEL. [Online]. Available: http://www.atmel.com/dyn/products/devices.asp?category_id=163&family_id=607&subfamily_id=1965&source=left_nav

[3] UNDK 20 – Sonic Sensor (2011) homepage BAUMER. [Online]. Available: http://sensor.baumer.com/sensor/products/scripts/level2.php?cat=CONFUltraschall_Sensoren&psg=ulProduktgruppe|Distanz_messende_Sensoren,ulErfassungsdistanz_max|-188456959&pid=Ultraschall_Analogausgang_UNDK_20_Sd_200_mm&language=de&country=CH&header=Schweiz+%2F+Baumer+Electric+AG&ptk=

[4] RoboRacingTeam (2011) homepage [Online]. Available: http://rrt.fh-wels.at

[5] R. Edlinger and M. Zauner, “A Modular and Scalable Electronic System for Mobile and Autonomous Robots”, 4th International Conference on Research and Education in Robotics,: Springer-Verlag CCIS-Vol. 156, 2010.

Technikum Wien's entry in the Robotour'11 competition

Thomas Koletschka
University of Applied Sciences Technikum Wien
A-1200, Vienna, Austria
Email: thomas.koletschka@technikum-wien.at

Alexander Hofmann
University of Applied Sciences Technikum Wien
A-1200, Vienna, Austria
Email: alexander.hofmann@technikum-wien.at

***Abstract*—This paper describes the system for an autonomous ground vehicle developed by a team at the University of Applied Sciences Technikum Wien. The goal is to deploy this robot in a city park in vienna and have it navigate autonomously from one point to another without colliding with any objects or driving off the permitted paths. We describe the hardware used by the robot and the software system developed for this competition which is based on the Robot Operating System.**

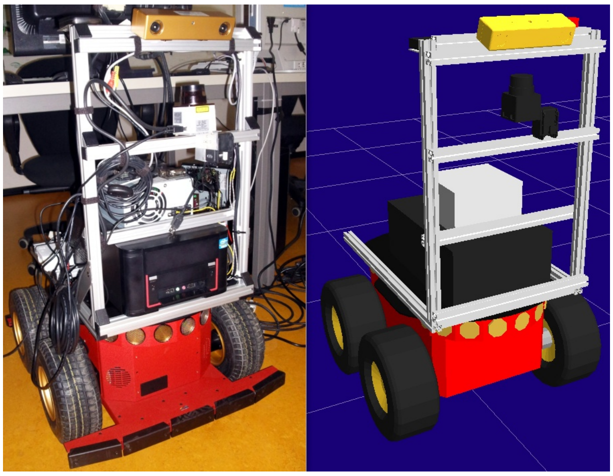


Fig. 1. Our entry in the Robotour 2011 competition (left) and its 3D model (right). The Robot is based on a Pioneer 3AT and additionally equiped with a Bumblebee2 stereo camera and a tilting Hokuyo.

I. Introduction

Robotour [1] is an outdoor delivery challenge for small sized fully autonomous vehicles. The goal of this competition is to navigate within a city park with the only available prior information being an OpenStreetMap map. In four different runs the robots need to carry a 5l beer barrel from a random location within the park to a random target destination. Points are awarded for traveled air distance towards the goal.

This paper describes our approach in conquering this goal with a robot based on a Pioneer 3-At (P3AT), displayed in figure 1 along with its simulation model. The robot will mainly be used on paved park roads, but we want it to be able to navigate on all different kinds of drivable areas such as pedestrian areas or sidewalks. Using this approach we hope to develop a more reliable system which is able to deal with different situations and is therefore able to achieve a good result.

This robot has been developed as part of an undergraduate project at the University of Applied Sciences Technikum Wien.

II. Hardware

A. Base Platform

The robot, as seen in figure 1, is based on a Pioneer 3-AT (P3AT), a robust four-wheel skid-steering robot suitable for outdoor use.
The P3AT comes with an array of each 8 sonar sensors and 5 bumpers on the front and back of the vehicle. Furthermore the platform provides access to its batteries to connect additional sensors and a serial port to control the robot with an external computer.

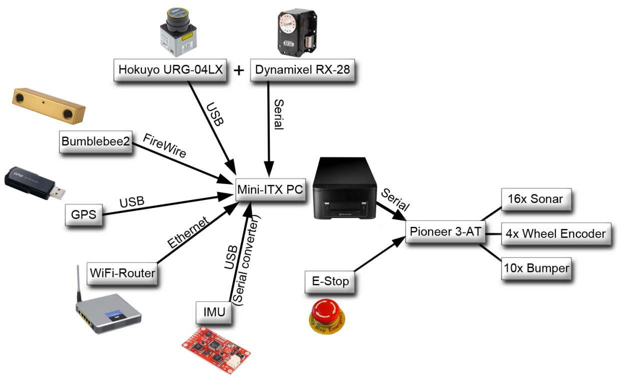


Fig. 2. Hardware Architecture

The robot's computer system consists of a 2.5GHz Intel i5 quad core CPU, 4GB of RAM, a 120GB SSD drive and an additional PCI-E FireWire card to connect the Bumblebee2 camera. The components are placed inside a Mini-ITX tower which is mounted on top of the P3AT above the front axle. That way we have enough room to place the required 5 liter

barrel above the rear axis which should provide an equal distribution of weight and a low center of mass.
Currently the computer system is connected to P3AT's battery system. This is sufficient for short testruns but an additional external battery system that powers the computer only is required to increase the duration to the required minimum of four 30 minute runs.

B. Sensor System

As shown in figure 2 the robot is equipped with 7 different kinds of sensors. The advantage of using multiple kinds of sensors is that one can build a more reliable sensor system by fusing the measurements of different sensors. Combining multiple sensors also helps in covering a wider area which allows the robot to navigate more reliably. Figure 3 illustrates the robot's (red rectangle) sensor coverage. While most of the sensors are oriented in driving direction the sonars on the back of the vehicle also allow for a rough estimation of what's happening behind the robot. This might be useful when passing other robots or when the need arises to back off a couple of centimeters.

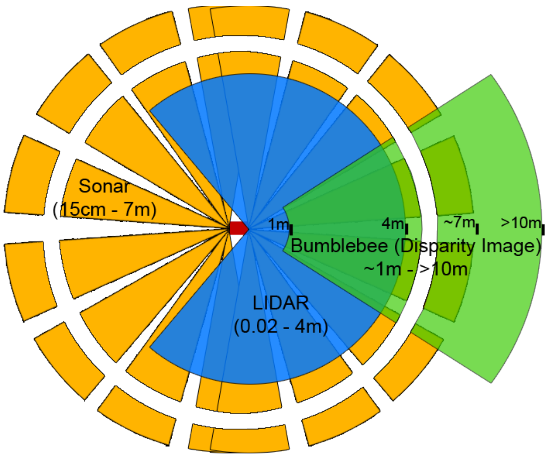


Fig. 3. Sensorcoverage

For inertial navigation, a CH Robotics UM6 Orientation Sensor, consisting of 3-axis gyros, accelerometers and magnetic sensors, is used in combination with a simple GPS receiver, P3AT's wheel odometry and visual odometry [2] computed from the stereo camera's images.
Based on the nature of the robot's environment we assume that GPS signals will be blocked off or heavily distorted most of the time. Furthermore the P3AT uses skid-steering so the odometry calculated from its wheel encoders is not reliable enough either. Therefore it is important to combine those measurements with more reliable sensors such as an inertial measurement unit (IMU) or visual odometry. As shown in [3] the combination of those sensors significantly improves the performance of an inertial navigation system.

The robot uses two main sensors to perceive its environment, a Bumblebee2 stereo camera and a tilting Hokuyo URG-04LX laser scanner.
The Bumblebee has a resolution of 640x480 pixels, a focal length of 3.8mm and a horizontal field of view of 66°. This stereo camera is the main sensor used in the robot's perception system. Its colour images are used for image based path detection and by using the images of both cameras a disparity map can be calculated which is then used for further caluclations such as obstacle detection. Additionally the camera images are used to calculate a visual odometry to improve the overall localization.
We decided against using the Microsoft Kinect which gained increasing popularity in indoor robotics applications since its release in late 2010 as this sensor is based on infrared structured light and is therefore not suited for environments with strong sunlight.
The Hokuyo laser scanner is mounted on a Robotis RX-28 servo motor. The laser has a range of 4 meters and a field of view of 240°. Mounting the Hokuyo on a servo motor enables us to tilt the laser which adds a third dimension to its readings. Tilting the laser allows the robot to cover a larger field of view and, especially due to the slow driving speed of our Pioneer robot, we are able to create a more accurate 3D model of the environment immediately in front of the robot.

The sonars and bumpers that come with the base platform are only used as supplementary sensors and emergency stop. Sonars are too imprecise to use them for accurate obstacle detection, but they still provide some rough estimate which e.g. can be used when passing other robots. The bumpers are used as emergency stop to protect the robot and the colliding object or person when all other sensors have failed.

III. Software Architecture

Our software architecture uses the Robot Operating System (ROS) [4] framework and is therefore mainly written in C++. ROS gained increasing popularity in the robotics community throughout the past years and today provides a large amount of robotics libraries and tools. Using ROS we are able to focus on the key problems of the competition such as path detection and reliable navigation as ROS already provides drivers for using all our sensors, a fairly mature communication interface and a lot of tools such as logging which are already integrated in the system.

Our system, illustrated in figure 4, is divided into three main layers:

- **Sensor Layer**: The sensor layer is the first layer in our processing pipeline. The processes in this layer simply read the measurements of all our sensors and publish them using the ROS communication framework. All the sensors used on our robot are supported by ROS and therefore we did not have to write a lot of code to integrate them in our system.

- **Perception Layer**: This layer receives the stereo camera, LIDAR (Light Detection And Ranging) and sonar data and fuses them into an evidence grid [5] using the sensor locations as described in the global robot model. Furthermore it tries to detect a drivable path and calculates the visual odometry.
- **Planning & Control Layer**: The last layer in our system is responsible for calculating actions based on the information provided by the sensor and perception layers. It calculates the current global position of the robot and tries to find the optimal path to the designated goal using an OpenStreetMap (OSM) map. After calculating the required trajectory to follow the path it sends the corresponding steering commands to the robot.

ROS' flexible communication system enables us to easily log, print and visualize information sent between all the processes which allows us to debug the system more easily. Another big advantage is that we are able to use the same system for both the real robot and a simulated model in the Gazebo simulator.

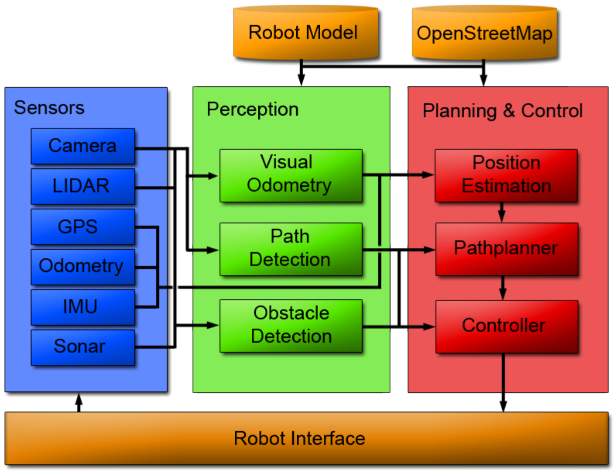


Fig. 4. Software Architecture

A. Global Information

There are two kinds of (mostly) static information which are of use to the whole system: the robot model and the OSM map used for navigation.

1) Robot Model: ROS uses the Unified Robot Description Format (URDF) to describe the structure of robots. A URDF file is an XML file describing a tree structure of the robot and different attributes such as size and weight of its parts. It can also be expanded with information about simulated sensors to use the robot model in a simulator such as Gazebo.
The robot model is modified and published using ROS' TF library which provides an easy way of dealing with the manifold of coordinate frames in a robotics system.
Besides for simulation purposes we use the URDF model to locate the origin of each sensor to fuse their data in a global reference frame. In the planning & control layer the model is used to avoid collisions with objects and plan a feasible path.

2) OpenStreetMap: The OSM map is mainly used for global path planning and the only map permitted for use in the Robotour'11 competition. We've developed a custom OSM parser and service layer to provide useful information such as the distance to the street crossings connecting to the current street or the width and type of the current street.
At the beginning of each run the globally optimal path to the goal is calculated using Dijkstra's algorithm with a Fibonacci heap [6]. If a path should turn out to be blocked or we should somehow miss a path we are still able to calculate a new path and proceed towards our goal.

B. Perception Layer

The perception layer is responsible to feed the planning & control layer with sufficient information to safely navigate the robot through its environment. The perception layer consists of three main modules: visual odometry (VO), sensor fusion and obstacle detection, and path detection.

1) Visual Odometry: Visual Odometry (VO) [2] is the process of determining the position and orientation of a robot by calculating the motion between each pair of images. VO is usually much more precise than wheel odometry and when combining the measurements of all position estimation sensors the best inertial localization results can be achieved, as demonstrated in [3]. This also helps to accomodate for blocked and distorted GPS signals as it is possible to reliably travel several hundreds of meters using only VO, IMU data and wheel odometry.
ROS provides a package called vslam to calculate the visual odometry based on stereo image pairs, therefore we did not have to implement this from scratch and save a lot of valuable development time. If the vslam package should not prove to be reliable enough for our purposes we will write a ROS wrapper for the VO library recently released[1] by [7].

2) Sensor Fusion and Obstacle Detection: Probabilistic fusion of multiple sensor measurements and obstacle detection is achieved using an evidence grid [8]. The advantage of using evidence grids is that different sensor models can be used to accommodate for differently accurate sensors. For example the uncertainty of an obstacle being present at a given location is much higher when using sonar measurements compared to the measurements obtained from a Hokuyo laser scanner.
After fusing all the sensor data into one common evidence grid, we simply need to threshold the values to detect obstacles and drivable terrain. Due to the probabilistic nature

[1] http://www.rainsoft.de/software/libviso2.html

of the grid and the short amount of time between each update, moving obstacles are updated accordingly and do not require any additional processing.

ROS already provides 2D and 3D occupancy grids, but we were not satisfied with the current implementation as it does not use a probabilistic model. Therefore we use the ROS wrapper of OctoMap [9], which implements an octree storing probabilistic sensor readings, for fusing our 3D sensor readings and afterwards convert those values into a 2D occupancy grid used by ROS' navigation system.

3) Path Detection: Using solely the fused evidence grid it would already be possible to follow a save path and avoid obstacles. The Robotour'11 rules specify that the robot must not leave the labeled pathways. This would not be possible using solely 3D information as the paths are not always bound by high objects such as fences or bushes. Therefore we also use the 2D color images published by our Bumblebee camera in combination with the previously calculated 3D environment to detect the drivable path.

Our approach is to find drivable terrain in a region immediately in front of the robot using the information stored in our evidence grid. After locating the drivable surface within that area we project the region into the image plane of one Bumblebee camera and use the selected pixel information to learn a visual model of the current street. Using that model we are then able to detect the path in the image and further extract an approximate shape of the path to avoid driving off the permitted path.
This approach has already been described in [10] and [11] and has successfully been applied to real-world navigation problems.

C. Planning & Control Layer

The Planning & Control layer is the final step in our processing pipeline. It is responsible for calculating the current position, calculating and following a global path, and finally sending the appropriate steering commands to the robot. To fullfill those tasks the layer's processes incorporate all the previously calculated information such as obstacle maps and drivable surfaces and global information such as the OSM map.

1) Position Estimation: The robot needs to have good knowledge about its current global position on a given map to reach the goal expressed in gps coordinates. In the robot's main domain, a city park, it is expected that the gps signal is blocked off or distorted by trees and buildings most of the time. Therefore it is not safe to rely solely on gps navigation. We use an approach as described by [3] to improve our global position estimate by fusing GPS, visual odometry, wheel odometry and IMU measurements. This allows us to travel reliably even with long GPS dropouts and large errors in received signals.
To do this we use a modified version of ROS' extended kalman filter (EKF) pose estimator as the standard filter only uses the estimates of three sensors while we are able to use the estimates of four different sensors.

2) Pathplanner: Robotour'11 rules demand the use of an OpenStreetMap (OSM) map as the only source for global navigation. Teams are not allowed to use private data such as custom built maps or pre-recorded trajectories. To fullfill this requirement the pathplanner uses the information provided by our global OSM service to calculate a global path to the desired goal position. In the best-case scenario this calculation usually only needs to occurr once at the beginning of each run. If a path should turn out to be blocked or we miss a crossing we are able to calculate a new route and will still be able to proceed towards the goal.
While maps usually provide a fairly good representation of street roads, unstructured paths such as paths in a park are usually only provided with simplified representations. To accommodate for those approximations we do not entirely rely on the path shape defined in the map but rather use the path detected by the vision layer to plan our local path. It is the responsibility of the controller module to follow this local path.

3) Controller: After the pathplanner has finished calculating a local path, it is the controller's responsibility to calculate the required steering commands to follow this path while avoiding all obstacles. Many robots, such as [12], use a simple PID controller to keep the vehicle on track. This method has the disadvantage of ignoring vehicle kinematics and dynamics. We chose to use a technique known as *trajectory rollout*, e.g. used in [13] and [14]. This method calculates multiple trajectories by simulating different feasible steering commands over a short amount of time. It then iterates over all calculated trajectories and selects the trajectory that does not collide with any obstacles and is closest in distance and orientation to the desired path.
We do not take speed control into account as the maximum speed of our robot is limited to 2.8km/h so we simply differ between stop and full-speed which simplifies our calculations when iterating through different possible steering commands. ROS already provides an interface for different controllers which we were able to use. In theory it would also provide rajectory rollout but, like many ROS libraries, this planner is tailored to the PR2's base and therefore was not suited for our robot.

IV. Conclusion

In this paper we have described our approach in conquering the goal of the Robotour competition which we will compete at in September 2011. We explained the hardware used in this system and the advantages of each sensor and how we accommodate for their disadvantages. Further we described the software system and the use of the Robot Operating

System in this system.

Currently our system uses just dead reckoning for navigation, which is the calculation of one's current position based on the mostion estimate since the last calculated position and hence prone to increasing errors with traveled distance. This can lead to wrong decisions when trying to follow a route on a street network and the robot might end up far from the intended destination.
On roads with lane markings localization can be improved up to a few centimeters in precision by aligning detected lane markings with previously recorded maps or aerial images of the current street, as shown in [15] and [16]. This approach is not possible with roads in our vehicle's domain, such as park roads and other unstructured paths, as those do not have any defined lane markings. Therefore a new approach needs to be designed to improve localization in such road networks e.g. by reliably detecting road crossing or road shape features to compare them against a human map and thereby increase knowledge about the current position.
Another interesting application would be the use of an unmanned aerial vehicle equipped with a downward facing camera to support the autonomous ground vehicle in determining its current position by providing aerial imagery of the current road segment and the robot's position on it.

We hope this paper provides good insight for both current and future teams and helps them in developing their systems. Our software system will be released after the competition and teams are welcome to take a look at it.

Acknowledgment

The authors gratefully acknowledge the contributions of all persons involved in the project. Special thanks to Mohamed Auf, Patrick Schwab and Marvin Vesely for their work in developing the system and Ewald Schmudermayer for his mechanical support. The project is funded by the University of Applied Sciences Technikum Wien.

References

[1] J. Ia and M. Dlouh, "Robotour - robotika.cz outdoor delivery challenge," in *Proceedings of the 1st international conference on Robotics in Education, RiE2010*. FEI STU, Slovakia, 2010, pp. 89–93.

[2] D. Nister, O. Naroditsky, and J. Bergen, "Visual Odometry," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, pp. 652–659, 2004.

[3] M. Agrawal and K. Konolige, "Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS," in *ICPR*, August 2006.

[4] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[5] H. P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots." *AI Magazine*, pp. 61–74, 1988.

[6] M. Fredman and R. Tarjan, "Fibonacci Heaps And Their Uses In Improved Network Optimization Algorithms," *Foundations of Computer Science, Annual IEEE Symposium on*, vol. 0, pp. 338–346, 1984.

[7] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d Reconstruction in Real-time," in *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[8] M. Martin and H. Moravec, "Robot Evidence Grids," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-96-06, March 1996.

[9] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems," in *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010. [Online]. Available: http://octomap.sf.net/

[10] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised Monocular Road Detection in Desert Terrain," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[11] I. Katramados, S. Crumpler, and T. P. Breckon, "Real-Time Traversable Surface Detection by Colour Space Fusion and Temporal Analysis," in *Computer Vision Systems*, ser. Lecture Notes in Computer Science, vol. 5815. Springer Berlin / Heidelberg, 2009, pp. 265–274.

[12] 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.-E. 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*, 2006, accepted for publication.

[13] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. D. Stewart, P. Vernaza, J. Derenick, J. Spletzer, and B. Satterfield, "Little Ben: The Ben Franklin Racing Team's entry in the 2007 DARPA Urban Challenge," *J. Field Robot.*, vol. 25, no. 9, p. 598–614, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1405647.1405653&coll=GUIDE&dl=GUIDE&CFID=30320363&CFTOKEN=48228491

[14] B. P. Gerkey and K. Konolige, "Planning and Control in Unstructured Terrain," in *ICRA 2008 Workshop on Path Planning on Costmaps*, May 2008.

[15] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *ICRA*, 2010, pp. 4372–4378.

[16] O. Pink, F. Moosmann, and A. Bachmann, "Visual features for vehicle localization and ego-motion estimation," in *Intelligent Vehicles Symposium, 2009 IEEE*, june 2009, pp. 254 –260.

Learning Robot-Environment Interaction with Neuroevolution

Martin Kysel
University of Vienna
Email: martin.kysel@vegi.sk

Pavel Petrovic
Comenius University Bratislava
Email: ppetrovic@acm.org

***Abstract*—At the Department of Applied Informatics at Comenius University, we follow a tradition of Artificial Intelligence (AI) study programs since the early 90s. Recently, the focus has widened towards Cognitive Science, where our department participates in an interdisciplinary international master program. Building upon the ideas of embodied intelligence, we are making efforts to include robotics experiments, studies, and projects into both programs. In this paper, we discuss our activities and present a sample student semester project that researches the important problem of learning robot environment using a predictive recurrent neural network (RNN) that is learned using state of the art evolutionary algorithm, namely NEAT - the NeuroEvolution through Augmenting Topologies, which stands out in its ability to evolve both the weights and the topology of RNN. The project brings new results, and provides a scenario for getting graduate students invloved in the academic research.**

I. Introduction

Robotics as an interdisciplinary field attracts the interest of various groups. On the one hand, electrical engineers who focus on automation and control engineering study robotics systems from the point of view of how to build and interconnect sensors, actuators, and power cells. They are concerned with processing the signals and build controllers that provide suitable responses, and steer the systems to achieve the intended behavior with maximum accuracy. On another hand, machine engineers study the different morphologies of robotics systems, how they can perform motion and manipulation, exhibit stability, durability, safety, and be power efficient. Yet another point of view is that of computer scientists, or AI researchers who are interested primarily in the logic of the control, what and under which circumstances the robot should do, and should not do, how could it be trained to do so, what algorithms should it use to perform its tasks. Thus AI researchers usually reason about implementing some intelligent behavior of robots that have already been built earlier outside of their lab. At the Department of Applied Informatics, we follow this third perspective, although we feel the era has finally arrived when the different perspectives can effectively be combined in interdisciplinary teams. We attempt to achive that through cooperation.

Our students are introduced to robotics in the third year of the undergraduate student program in the course Algorithms for AI Robotics. The course combines theretical lectures with practical exercises and project work. The lectures include basic introduction to concepts of sensing, locomotion, navigation, and control, as well as selection of AI Algorithms that have the application in robotics. These include bayesian and probabilistic approaches, reinforcement learning, several evolutionary methods (learning classifier systems, CMA ES, NEAT), some vision algorithms (SIFT and SURF, Fly Algorithm). In the exercises, we provide the students with several introductory hands-on activities with various platforms - LEGO Mindstorms NXT, SBOT robots (a simple educational modular differential-drive robot built within our group with bootloader-enabled AVR microcontroller, bluetooth radio, and basic set of sensors), and Robotnacka platform with its remotely-controlled robotics laboratory. The exercises are particulary appreciated by the students as this is one of the few courses where they get some practical experiences. In the second part of the course, students work on various projects of their own choice. These include robots that participate in robotic contests, various improvements of the platforms in our laboratory, but mainly various exercises of learning algorithms for robots. In one project, for an example, our students designed an algorithm for the E-puck robot to solve the maze navigation task for the Micromouse contest that is annually held as part of the Istrobot contest [1], see Figure 1.

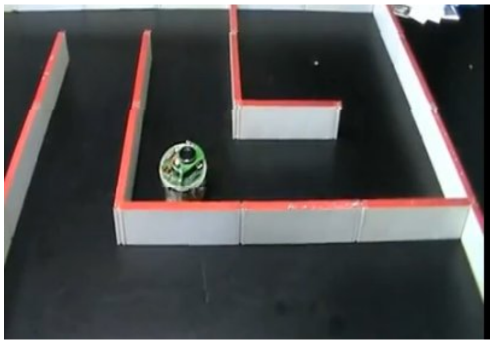


Fig. 1. E-puck robot navigating the practice Micromouse maze.

Our graduate students have the opportunity to get involved with robotics in a practical seminar, and the visiting students of the cognitive science study programme also in their semester project. In addition, every year, several students choose bachelor and master theses in the area of robotics. These include the studies on localization, mapping, simple 3D vision, or using robots in primary or secondary schools. All these works have the advantage of being applied and motivating, A care has to be taken to provide the students with the hardware and software setup that is consistent and proven to work to avoid the possible technical traps that could make the project unfeasible.

In the remaining sections of this article, we present an example of a semester student project in cognitive science that combines the evolutionary methods with a recurrent neural network controller that learns the map of the environment in such a way that it is able to predict the next sensory inputs based on the current sensory inputs and the control commands sent to the actuators. The following sections explain the background and motivation, the task, the evolutionary algorithm, the implementation, the experiments performed, and the coginitve connections.

II. Motivation

Our work is inspired by the approach of Tani [16], and its follow-up of Nolfi and Tani [12]. The basic idea is to develop a software mechanism that can effectively predict what is going to happen in the environment next, if a mobile robot that is navigating in that environment takes a particular action. The sofware mechanism thus receives the current sensory percepts and the intended action and should estimate the next sensory perceptions. Nolfi and Tani in their previous work utilized a complex architecture of two feed-forward, backpropagation-trained neural networks arranged in a predefined topology with one extra winner-takes-all self-organized network in the middle of the two. By this choice, they had to make a guess on the topology, which was not guaranteed to be the correct choice. For instance they bounded the interface between the second and third networks to be only one of three values (index of the winning neuron). Such topology would have to change each time the structure and complexity of the task and environment changes. We are also not so convinced by their layering approach. On the contrary, we suggest to solve a similar task using a recurrent neural network with both topology and weights evolved. This allows the method to select the appropriate topology, and tune the weights to perform the task required. A state-of-the-art method NEAT [14] has been shown to be able to find the RNNs for complex tasks [15]. In addition, we will assume a robot navigating in the environment randomly as contrasted to a deterministic behavior of Nolfi.

Prediction in general is one of the elementary and well studied tasks [3], despite that we are not aware of others applying this method on the selected task, altough a similar task was studied in [6]. We follow the setup of Nolfi and use an embodied learner - The Khepera Bot[11], In a series of succeeding experiments, we performed a learning task with a population of simulated Khepera-like robots. Both versions, the real and the simulated ones are able to determine the distances to obstacles with built-in ultrasonic or infrared sensors. Each robot is free to move in the environment avoiding obstacles and room boundaries. Our work started with specifying concepts, that had to be fulfilled during the whole set of experiments. How they were solved and what problems we encountered can be found in the following sections.

- We know that running a neuro-evolution with real robots is very time consuming, so we will simulate the process. As a logical consequence of this step, we have to choose an appropriate Khepera-like simulator and a neuro-evolutionary framework.
- We want the complexity of the environment to be increasing during the set. We want to start with a simple circular environment and move on to more complex shapes.
- The starting pose of the Khepera-like agent will be random in all runs. We will add a random movement pattern into the behaviour of the agent to ensure generalization.
- Selecting the starting genome is a non-trivial task for the neuro-evolution. Starting too complex can slow down the solution convergence or totally miss a global optimum. It is a principle in NEAT to start with a zero genome and let the framework build upon the combinations that prove useful.
- We wish to see how can both sensory and motor values be combined and how do both contribute to the sensory prediction. We therefore performed multiple runs with different number of sensors.
- Among the variables each neuro-evolution depends on are adequate evolutionary parameters (mutation coeffcients, mating probabilities, cross-species mating, recurrence probabilities, number of hidden nodes)
- In the beginning, we do not know how many individuals and generations will be required to solve the problem. We will find these values empirically. Once found the values will be fixed across all experiments.

III. Task

As introduced above, the task deals with a simulated Khepera-like robot. The simulator should satisfy the following requirements:

- create simple obstacle patterns,
- measure distance to the next obstacle in a given direction with a set of built-in sensors,
- use eight different distance sensors - four in the front, two at the sides and two in the back. We expect the sensors to be noisy, forcing the network to generalize,
- simulate movement of the agent and provide a reading of the motor action,
- real embodied agents do not move with an exact distance during each run, this must be accounted for,
- update the sensory reading based on the current pose in the surrounding environment,
- generate environments with obstacles of increasing complexity, we did not solve this requirement completely.

A. Khepera Simulator v2

The neuroevolution builds on top of a simulated khepera-like robot. We used the *Khepera Simulator v2* by Olivier Michel from the University of Nice Sophia. It uses an older

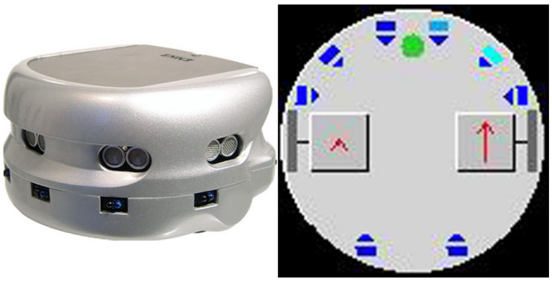


Fig. 2. Real and Simulated Khepera [9]

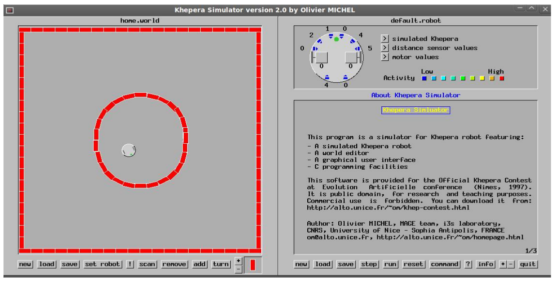


Fig. 3. A simulated circular room with the Khepera robot inside

version of the C standard, but could be adapted to our current needs.

Khepera was first developed in 1992 as a research and teaching tool by the Swiss Research Priority Program. Because it is a simple, robust and miniature mobile robot it is still widely used for research and educational projects. It communicates over a serial port with any application written for this purpose. Because our goal was a successful neuroevolution we would need a long period of time to run all generations and genomes on one single hardware and therefore decided to use a simulator that can be accelerated in timely manners.

The simulator creates an environment of $1m \times 1m$, in which obstacles (bricks) and light sources (lamps) can be placed. Before the environment can be used for further spatial tasks, the graphical interface has to be scanned. This has shown to establish difficulties with randomly placed obstacles, as every placement would have to be scanned into the environmental data file and the whole application would be slowed down considerably.

The *Khepera*-like robot includes eight infrared sensors allowing it to detect obstacles in many possible directions. The distance to the closest obstacle is represented by a 10bit value. The range of the infrared sensor is approximately 7cm and all obstacles further than this threshold return only random noise. The distance is an inverse function with $0 - 10$ meaning no immanent object and 1023 being right next to an obstacle. Because of the built-in artificial 10% noise function these values are only approximations of the real distance.

The motors of the Khepera can take values between -10 and 10 and also have a 10% noise built-in. Both motors can run backwards allowing the Khepera to turn in place. We used a random movement function with different motor values each five steps. If there was an obstacle found on the side of the Khepera, it turned to the opposite direction.

B. Evolutionary Algorithm

We used NEAT because of its interesting attributes and abilities. It was developed by Stanley [14] and we used a Java implementation.

Neuroevolution

Neuroevolution is a class of search algorithms based on evolution and neuronal computation, both being inspired from the nature. Every individual found by a neuroevolutionary algorithm represents a point in the search space. By point we understand either the connection weights, as common in some approaches, or both the connection weights and the topology of the network. The goal of a neuroevolution is to find one or more individuals that approximate global optimum of the search space. The search is parallel, working on a population, i.e. a set of individuals that are modified and combined using evolutionary operators. A set of the optimized parameters (in this case, nodes and connection genes) form a *Genotype*(genome) of the network and is further discussed in later sections. The actual network that is further evaluated by the objective function is the *Phenotype*.

Stochastically selected individuals with higher fitness values are allowed to mate with other fit individuals in order to create offspring that form the population in the new generation. A combination of two fit genotypes using the *cross-over* evolutionary oepartor generates with some probability even a better set of parameters. To ensure that the whole search space is potentially covered by individuals a *mutation* probability is introduced. Mutation changes a selected individual depending on the mutation strength and creates a new point in the search space. It is a fact that many of these individuals are unfit and will be discarded in the course of the evolution. Still, this process is necessary in order to traverse the whole search space and therefore find the global optima.

NEAT diverges from many NE algorithms in the way it sees the search space. NE usually tries to find the optimal set of parameters for the given network topology, not adapting the topology itself. NEAT modifies the search space by adding or subtracting dimensions to/from the genotype. In praxis this means modifications in the number of connections between the nodes, or in adding new nodes to the network. Therefore we are not required to perform previous research of the search space and an optimization of the number of nodes. We do not even require knowledge whether particular input nodes contribute to the overall change in the output. All this is searched and tested along with the other parameters by NEAT itself.

Nodes and Connections

Nodes represent an abstraction of the neurons present in the living brain. They can be connected together to form an Artificial Neural Network. Most ANNs are constructed of

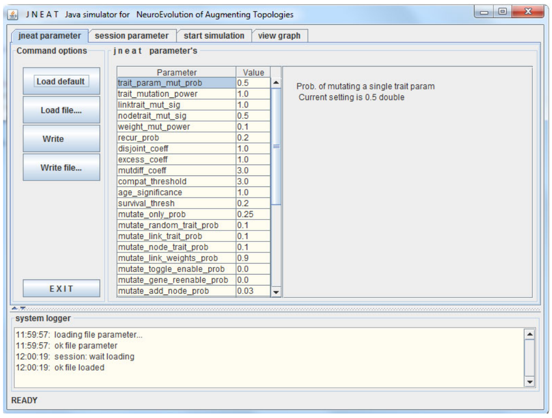


Fig. 4. The jNEAT GUI

layers. In principle there is an *input layer* of sensors that receive information from the environment, an *output layer* that simplifies the reaction of the individual to the stimuli that is returned to the environment and finally a variable *hidden layer* that performs the transformation of input to output. The hidden layers of our evolved networks consisted of multiple layers with variable number of neurons each.

Connections are an abstraction of the synaptic connections between neurons. Every neuron has an activation function F_a that is dependent of the weighted sum of incoming connections. $y_j = F_a(\sum_j w_{ij} x_{ij})$. Connections can be either positive (excitatory) or negative (inhibitory) and always have a direction (from input via hidden to output).

Some of our networks have intra layer connections (output-to-output or recurrence). This is important so that the network can have an internal state and solve task that are not purely reactive.

Genome

Every neuroevolution framework uses a different Genome for specifying the network topology (Phenotype). NEAT uses a representation that specifies node types (input, output, hidden) and connections/links (with from node, to node, recurrence flag and weight). Genes not only identify links but also get a unique identification number that helps the evolution to cross only equivalent connections.

Many NE genotypes have the problem of *Competing Conventions*. The same configuration of network weights can be seen as a permutation of its connections. A network with 3 hidden nodes can have $3! = 6$ equivalent encodings. All of them will have the same fitness and if mated together they will create an offspring with a drastic loss in functionality. If each connection (a gene) has its own unique identification number as it is in NEAT, this problem cannot occur. Connections between input node 1 and hidden node 3 will only be mated with connections between input node 1 and hidden node 3.

Starting Genome

NEAT uses a starting genome, with all input and output nodes and optionally with some number of hidden nodes. There can be either no connections, or all nodes can be interconnected. One of the principles of NEAT is to start as small as possible. It is the task of NEAT itself to create the topology of the network. One does not need to specify connection weights for the starting genome as this is varied across the individuals of the first generation. In fact the starting genome only specifies the topology of the network. We tried multiple starting genomes ranging from no connections to every-input-with-every-output thus recurrent connections on the output side. We noticed that it is mostly the best practice not to start with a simple genome, but to provide a certain starting complexity. With no-connections at the start of the evolution we created a network with interconnected thus recurrent nodes at the 30th-50th generation. Therefore starting with a more complex network saves computational time and shows no difference in the final network topology.

Species

We are solving a multimodal problem. Each of the multiple optima can be located in a different part of the search space. Speciation included in the NEAT protects the population from premature convergence towards a sub-optimal solution. It also groups genotypes into similar problem solvers. Because individuals share the fitness of other individuals in their species, (see below), they are given time to optimize their structure for a given problem before competing with the other species. Whether individuals belong to the same species is determined by the *Compatibility Threshold*.

The weakest individuals of each species are eliminated (a fraction of 1-*Surv Threshold* = 80%) and a random pairs of the remaining ones are allowed to reproduce. The best individual in the species always survives as long as the whole species is not eliminated.

Fitness Sharing

Introducted in [4] NEAT uses a technique to limit the size of a species by forcing all individuals to share their fitness with all other individuals. Even if many organisms perform well, the species cannot grow too big taking over a large portion of the population. It is a crucial limitation to support multiple search subspaces. The outcome of the fitness sharing is that the number of individuals reserved for a species is proportional to the average fitness in that species.

Generations

In our experiments 200 generations were enough to find a stable genome for the given task. We separated the experiment into smaller 20-40 generation sessions to protect ourselves from unexpected interruptions. After a time (drop-off age + 5) jNEAT starts a delta-coding process and creates a completely new set of species. This causes a drastic decrease in the number of species and mostly no advantage in the means of average fitness.

Evolution Parameters

The Table I shows some of the evolution parameters as they were set in our experiments. *Value* represents the optimal parameter and *Range* the meaningful value interval across different types of experiments.

TABLE I
PARAMETER VALUES

Parameter	Value	Range	Description
Weight Mut	0.1	0.1-2.0	Mutation power of the connection strength
Recurr Prob	0.2	0.2 - 0.3	Chance that a link will be recurrent
Compat Thres	3.0	3.0-5.0	Similarity measure of individuals in one species
Age Signif	1.0	1.0	Inverse boost for new species
Surv Thres	0.2	0.2-0.4	Percentage of species allowed to survive
Mut Only Prob	0.25	0.25	Probability that reproduction will have no crossover
Add Node Prob	0.03	0.-0.03	New gene will be added to the genome
Add Link Prob	0.1	0.05-0.2	New connection will be added to the genome
Pop Size	100	50-400	Number of genotypes in the population
Drop-off Age	25	1-400	Maximal age of a species before being penalized. In original NEAT this value is called maximum Stagnation

Recurrent Connections
The task of the EA was to find a function that was dependent of preceding steps of the robot. A network that solves such a problem must therefore include the previous state into its prediction model. This is enabled by recurrent connections. We used a recurrence parameter of 0.2 and in average one recurrent connection was adopted in 30 generations. In the resulting model all output nodes had recurrent connections to themselves.

C. Known Issues

- The delta coding process recodes all genomes into new species if there is a stagnation in the evolution. This process is implemented in jNEAT to start after $DO + 5$ generations without a fitness winner. A value of 0 for the Drop-off-age turns off this criterion but anyways starts the delta coding process every fifth generation. If there should be no Drop-off in the population the value has to be set to MaxGeneration or higher.
- jNEAT only reads parameters on startup. So if the evolution is stopped and the parameters are changed the whole application must be restarted.

IV. TERMINOLOGY

A. Error

In this work the term Error refers to the sum of differences between the actual and the expected sensory reading. We had a sum of 5 sets of runs with 2000 steps each. The average/worst run was used for the evaluation of the network. The *maximal error* is equal the number of sensors times the number of steps: $2000 * 2/4/6/8 = 4000/8000/12000/16000$. As explained in Section III-A, the Khepera simulator provides sensory readings with a $\pm 5\%$ margin of noise. Values below 0.1 that are physically out of the sensors range are randomized between 0 and 0.1 providing virtually no meaningful values. The noise for all other distances can range from -0.05 to +0.05.

In two consecutive steps the worst case will be the double noise margin. in the worst case of prediction error a total difference of 0.1 has to be tolerated as faultless. The worst error of a perfect predictor (*minimal error*) is $2000*0.1*2/4 = 400/800$.

For each error larger than 0.3 there was an extra penalization in the form of an additional error value of 1. An error of 0.35 therefore becomes 1.35. In some experiments we used an extra penalization measure of 5 if the error was larger than 0.5 (0.51 becomes 6.51).

$$maxError = Sensors * Runs$$
$$minimalError = Sensors * Runs * 0.1$$

B. Fitness

We used different fitness functions for different experiments. All have in common that they are dependent of the total penalized error and that the maximal available fitness is 1000, with 0 error. The basic fitness function normalized to the 0-1000 scale was:

$$f_i = (maxError - Error_i)^2 - Pen^2$$
$$Pen = \begin{cases} 0, & \text{if } Error_i < 0.3 \\ 1, & \text{if } 0.3 < Error_i < 0.5 \\ 6, & \text{if } 0.5 < Error_i < 1.0 \end{cases}$$

The limited visibility of the Khepera robot made most of the values fed to the network be in the range 0 to 0.1 causing a very high average fitness.

For additional evolutionary pressure, all fitness functions were sigma-scaled before used in the evolutionary selection.

$$SigmaScaleFactor = 1 + \frac{(f_i - \overline{f})}{2\sigma}$$

The mean $\overline{f}$ and stdDev σ was not computed from the current but from the closest passed generation. As the fitness functions from all the networks in the generation ranged from 0 to 900 having a high standard deviation, this scaling was of very little use. Even with a low mutation coefficient a mean value shifted between 400 and 600 with a standard deviation of 40.000-80.000.

V. IMPLEMENTATION

Our application makes use of several different libraries and programming languages. The Khepera Simulator runs an older C code and is written for Linux. It can be started under Windows with Cygwin or other Linux simulating environments. Currently the application only works under Linux or Mac OS X, because Cygwin cannot be started from Java. The main application and the jNEAT are all Java based and provide the main GUI. The entry point is jNeatMain and its main function that can be called without any parameters.
The first step is to load parameter values for the evolution. It

TABLE II
FITNESS OF NAIVE PREDICTORS, 2 SENSORS, CIRCULAR ROOM, DATA FROM 10 RUNS

Predictor	Min	Max	Mean	Median	σ	mean error
Rnd (pen)	0	0	0	0	0	14476.5
Rnd (0 pen)	262.1	275.6	268.9	270.3	24.5	1949
0.5(0 pen)	163.3	206.2	182	182.5	108	2293
Zero	456.2	816	667.1	677.7	8144	796
Naive	855.5	958.2	925.9	932.4	508	158
Nv(0 pen)	972.5	976	974,5	977.2	15,2	52

can by default be found in the resources folder and it should be enough to press *load default*. Parameters and their values are discussed in a section III-B.
After the parameters have been set the user can move on to the second tab called *session parameter*. Here we define data input and output pairs, the fitness function and the starting genome. The epoch parameter equals to the number of generations. A predefined file can also be loaded.
We can start the simulation by pressing *start* under the *start simulation* tab. After the defined number of generations the application stops and writes the last generation into a separate file - *primitive*. All generations and their species can be found in the output folder.
The Khepera simulator is started from within the main application and it must not be started by the user. Because the simulator is a separate application that is not meant to be part of the main applications core, it is required to *make* the simulators source. Any changes to the simulator - number of runs, behavior functions, number of steps... have to be recompiled and often changed in the main application as well. All changes to rooms (new ones and not predefined experiments) have to be changed in the C code as well.

VI. EXPERIMENTS

In this section we provide the reader with an overview and summary of the experiments we have done: there was a total of six naive predictors and eight predictor networks in three types of rooms. Networks varied in their number of sensors, the penalization and their fitness function.

A. Preliminaries

A part of the preparations was the evaluation of all evolutionary parameters. In addition to operator probabilities, this involves the population size and the number of iterations. All this, has already been presented in this paper. Before experimenting with ANNs, we wanted to see how various naive predictors solve the task. Their summary is shown in Table II.

Random Predictor
Is a predictor that answers with a random number between 0 and 1. Fitness in case of penalization is set to 0, because the penalized $error_i$ is above $maxError$.
ZeroPointFive Predictor
Always predicts the expected average value of 0.5. We can see that this expectation is not accurate, because the Khepera is mostly out of its average sight range. A fitness of 182 is very inaccurate.
Zero Predictor
The zero predictor always predicts the next input to be zero. This is due to the fact, that most of the time the Khepera is out of range of its sensors and only noise is recorded. A zero predictor is a good measure to see if it would be best for the network not to solve the main problem, but just try to approximate the noise. As we can see in Table III the average fitness of a zero predictor for the circular room with two sensors is 667.1 while being penalized. It is not the best possible outcome but still very above average for a random guess (fitness = 268.9).
Naive Predictor
The Naive Predictor is the first predictor that actually uses the sensory information. Without looking at previous runs, the naive predictor easily predicts the same value as it reads from its sensors at the time t. This would be equal to a genome with only one connection from sensor i to output i with a constant maximal weight of 1.0. As we can see from Table III the fitness for a Naive Predictor is very high. We expect the fitness of a trained network not to fall below 800, which then would still be worse than a naive predictor.

TABLE III
ALL CIRCULAR PREDICTOR NETWORKS

Network	Min	Max	Mean	stdDev	mean std err
Circular 2	920.2	993.1	945	193	55.3
Circular 4	817.2	908	875.4	571.4	128.8
Circular 6	741.1	854.1	798	896.6	201.4
Circular 8	728.4	838.7	792.8	921.7	219.2
Naive 2	855.5	958.2	925.9	508	79

B. Circular Room

For the comparison sake we introduced a new measure the *standardized error per sensor*. The error reported in Tables III,IV is divided by the number of sensors used for prediction. Because fitness is independent of the number of sensors it is a standardized value per se. All values in Tables III, IV are computed from the best individual in 25 randomized trials.

We started the experiment with a circular room without any obstacles in the room with two front sensors active. This is also the scenario in which all naive/random predictors were tested. The only network that performed better than a naive predictor was a simple two sensoric representation. As we can see in Table III the normalized error of 55.3 lies below expected minimum of 200 ($2000 * 0.1$). This points out this mean value is very satisfactory and even worse values can be tolerated. An average of 10 steps out of 2000 were not predicted correctly by this network. This network of the 153th generation is in the appendix.

The second experiment used slightly more sensors: 2 (left), 3(front-left), 4(front-right) and 5(right). The best prediction was achieved by a network of the 120th generation and species 49. The normalized error of 128.8 still lies below expected

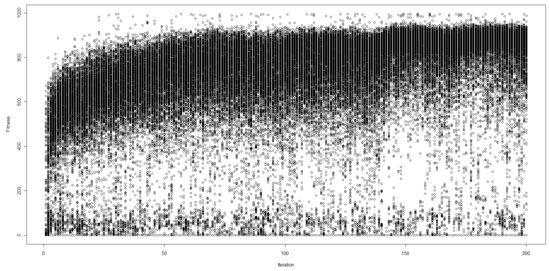


Fig. 5. Fitness by Iterations in a 2 sensors run

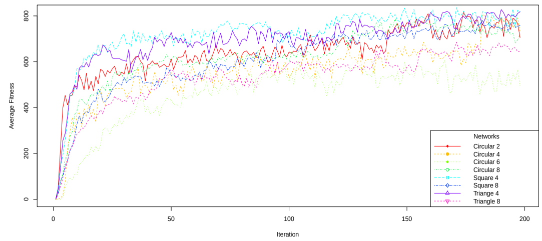


Fig. 7. Average Fitness for all rooms per Iteration

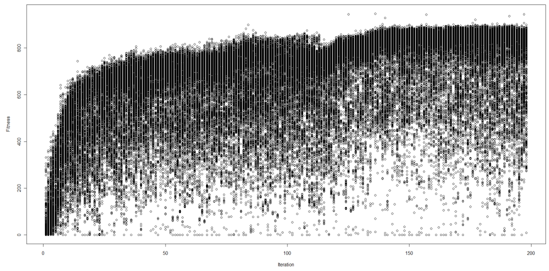


Fig. 6. Fitness by Iterations in a 8 sensors run

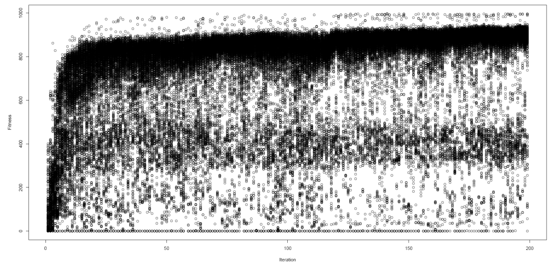


Fig. 8. Error by iterations for a Quadratic Room with 4 Sensors

minimum. There is no need for further evolution of this type of network. We noticed a dependency of given prediction from the previous near-by sensory output. This link was extablished by NEAT itself and was not included in the starting genome.

The third and fourth experiment using six and eight sensors respectively, could use more evaluations of the networks genome. Both stanardardized error values lie above the expected minimum. The standard deviation also shows a very unstable prediction performance. We think that using multilayered evolution scenario might improve the prediction outcome. After two houndred generations, both networks became very big slowing down the solution convergence. After computing the network topology with NEAT with a maximum of one houndred generations we would suggest to perform a standard evolution on top of the existing topology, without adding or removing genes.

C. Quadratic Room and Triangle Room

The similarity in outcomes in Tables III and IV across different types of rooms shows us, that NEAT is able to compute the best genome for all tested types of rooms. See Table 8.

TABLE IV
OTHER ROOM PREDICTORS

Network	Min	Max	Mean	stdDev	mean std err
Square 4	841.6	935.1	894.1	617.2	108.8
Square 8	386.6	902.9	788.9	7415.9	223.6
Triangle 4	852.3	961.7	892.3	726.7	110.8
Triangle 8	654.1	876.8	806.4	2634.8	204.0
Naive 2	855.5	958.2	925.9	508	79

D. Learning Outcomes

The neuroevolution we applied to all our rooms started with a minimal genome. There were no hidden nodes, no recurrent nodes and a minimal set of connections. After a few generations a basic template emerged, that used previous output activation as reference. The biggest change in the output was caused by the activation of the same sensor, other sensors contributed with a lower ratio. The specific structure of the room and the fine tuned genome was computed after another 100-150 generations. The structure of all available final networks is very similar. There is a dependence on:

- current input of given sensor i
- previous input of sensor i
- previous prediction for sensor i
- current near-by sensory readings
- previous near-by sensory readings
- motor readings

This list of dependencies was developed by NEAT itself along the topology of the evolved network. They were identified from weighted genome graphs such as in Figure 9.

VII. A Cognitive Agent

Our model of sensory prediction is in principle very similar to an abstract visual pathway located in the human brain. The biological original does not work in a purely feedforward manner. We have to keep in mind that 80% of the input to visual processing areas (such as the LGN) actually come from the same area. Only the remaining 20% come from the actual visual sensors. Our framework partly approximated this in including recurrent nodes that combine previous activations

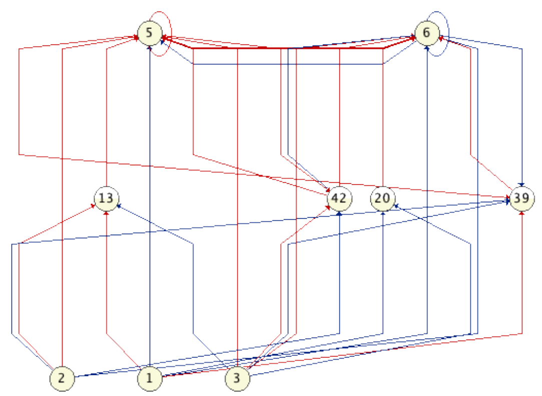


Fig. 9. Example network phenotype for a 2 sensory network after 40 generations. Numbers represent unique identification numbers for nodes. 1 and 2 are sensor readings and 3 the motor reading.

with the current input. There is of course no claim that this abstraction is biologically valid in any sense, but it is not a simple input-output forwarding function. If we would pre-process the data and create more complex rooms, a robot should be able to introduce means for information filtering and attention. This was not implemented, but in surely emerged relatively early in the biological evolution. We also know that higher level cognitive feedback modifies the processed input that reaches the cognition itself. Our work uses a simple model for both sensory representations and the agents feedback. To make the model more biologically valid, we would need devices that provide much more informational divergence that the IR distance sensors. In principle it should be the task of the agent, to find and extract the relevant information from any available reading. Along all the cognitively implausible outcomes, this one was held true. We provided the robot with sensory readings that were partly irrelevant for many of the sensors that were to be predicted. Yet each sensor 'selected' only the useful information. Attentional mechanisms are by no coincidence an integral part of the cognition. Our agent had no explicit representation of either obstacles or free space. The motivation of the agent used in our experiments was purely extrinsic and the actions (obstacle avoidance, random movement) were alien to the agent. What our experiments showed is the relevance of all sensory information. In the topology of the network developed by NEAT we see, that not only current readings, but previous prediction, inputs and outputs of other sensors and motor actions were of relevance alike the real embodied agents.

VIII. Conclusions

In the article, we have showed how a student semester project in cognitive science uses robotics task to learn about various methods of AI. The resulting work represents a tangible research result and advances the academic research. Selecting the open-ended research problems for the student work is not only useful to motivate them to further studies, but it is a nice opportunity to identify issues that could otherwise easily remain overlooked.

In particular, the student work showed how neuro-evolution frameworks such as the 'NeuroEvolution through Augmenting Topologies' extract relevant information from the environmental readings and create a meaningful network topology. We provided an overview of the terminology used in neuro-evolution and an array of parameters that have to be specified. In the future work, we could add variable obstacles. The performance of the learning algorithm could be improved by clustering the input signals into different groups and then balancing the number of samples from each group. For instance, the robot may spend most of the time far from obstacles, when the prediction task is easy, and only few time steps in front of obstacles, when the data are critical. Filtering out this duplicit information should result in a faster convergence. Another interesting direction could include the prediction into the behavior control and action selection of the agent. The project source code is available at [5].

References

[1] R. Balogh. Popularization of the robotics. *DidInfo 2008. 14th International Conference*, 2008.
[2] B. V. Bonnlander and A. S. Weigend. Selecting input variables using mutual information and nonparametric density estimation. In *International Symposium on Artificial Neural Networks*, pages 42–50, 1996.
[3] J. T. Connor and R. D. Martin. Recurrent neural networks and robust time series prediction. 2006.
[4] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application.*
[5] M. Kysel. Neuro Evolution of Simulated Khepera Behaviour Prediction. *Google project hosting svn code repository*, 2011. online: `http://srwddds.googlecode.com/`
[6] N. Kohl, K. Stanley, R. Miikkulainen, M. Samples, and R. Sherony. Evolving a real-world vehicle warning system. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006.
[7] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall symposium on relevance*, pages 140–144. AAAI Press, 1994.
[8] R. Legenstein and W. Maass. Input prediction and autonomous movement analysis in recurrent circuits of spiking neurons. *Rev Neurosci.*
[9] O. Michel. Khepera simulator version 2.0, February 2011.
[10] M. Mitchell. *An Introduction to genetic algorithms*. MIT Press, 1998.
[11] F. Mondada, E. Franzi, and P. Ienne. Mobile Robot miniaturisation: A tool for investigation in control algorithms. In *Proceedings of the Third International Symposium on Experimental Robotics*, 1993.
[12] S. Nolfi and J. Tani. Extracting regularities in space and time through a cascade of prediction networks: The case of a mobile robot navigating in a structured environment. *Connection Science, vol. 11, no. 2, pp.125-148*, 1999.
[13] D. Parisi, F. Cecconi, and S. Nolfi. Econets: Neural networks that learn in an environment. *Network, Volume: 1, Issue: 2, Informa Healthcare,pp 149-168.*
[14] K. O. Stanley. Efficient evolution of neural networks through complexification. *THE UNIVERSITY OF TEXAS AT AUSTIN, 2004, 227 pages; 3143474.*
[15] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.
[16] J. Tani. Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:421–436, 1996.

Simulation and Control of a Biped Walking Robot using Kinematic and Dynamic Modelling

M. Kojouri Manesh[1a], H. Kojouri Manesh[2b] and A. Torabi[3c]

[1] Dept. of Mechanical Engineering, Shiraz University, Shiraz, Iran
mohamadkojouri@yahoo.com

[2] Dept. of Electrical Engineering, Islamic Azad University, Marvdasht, Iran
kojourihamid@yahoo.com

[3] Dept. of Industrial Engineering, Islamic Azad University, Shiraz, Iran
torabi_amir251@yahoo.com

***Abstract*— In this article, we intend to consider the behavior and control of a biped walking robot using kinematic and dynamic relations. At first, by using simple model of humanoid robot and essentional equations the angles, angular velocities, accelerations of motors and required torques for moving on a straight line are find out. In the second step considering numerical values of the robot parameters and constructing the dynamic model the abilities of robot are examined and simulated.**

***Keywords*—Humanoid robot; simulation; control**

I. INTRODUCTION

The need for robots has recently been changed from industrial automation to human friendly robot system [1]. One of them is WABIAN constructed by Waseda University and WABOT which is the world's first life-sized humanoid robot with the ability of walking and dancing [2]. H6 and H7 are humanoid robots constructed by University of Tokyo [3]. JOHNNIE is an anthropomorphic autonomous biped robot constructed by Technical University of Munich [4]. MK.5 is a compact size humanoid robot with 24 D.O.F. constructed by Aoyma Gakuin University [5]. The most impressive humanoid robot should be HONDA humanoid robots. P2 is the world's first cable-less humanoid robot, which can walk and can go up/down stairs [6]. P3(height 1600 mm, width 600 mm, weight including batteries 130 kg, 6 D.O.F./Leg, 7 D.O.F./Arm, 1 D.O.F./Hand) appeared in 1997 with the same mobility as P2 [7]. In 2000, further downsizing P3, ASIMO that stands for Advanced Step in Innovative Mobility appeared with children-size (height 1200 mm, width 450 mm, weight including batteries 43 kg, 6 D.O.F./Leg, 5 D.O.F./Arm, 1 D.O.F./Hand, 2 D.O.F./Head) [8,9].

This work includes the simple model of ASIMO robot and simulates its motion using series of motors to establish automatically robot stability during its motion. This robot has 23 D.O.F. Six motors that move the robot on the straight direction, eight motors control robot's stability and the other motors are used for extra body movements and 3D motion.

II. EQUATIONS OF MOTION

A. Two dimensions dynamic equation

There are many and complex equations to control a biped walking robot accurately; therefore, it is difficult to achieve a standard control algorithm. So, the simple models are used. A simple 2D model which has 5 D.O.F is shown in Fig. 1 [10].

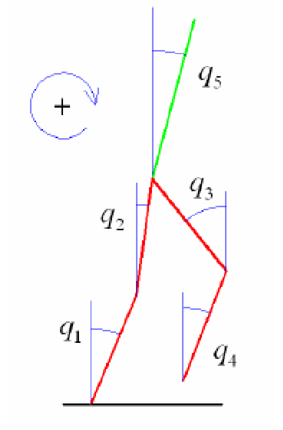


Fig. 1 absolute angels of 2D model with 5 D.O.F.

The dynamic model is given by:

$$B(q)\ddot{q} + c(q,\dot{q}) + g(q) = A\tau \quad (1)$$

Where $B(q)$ is the inertia matrix, $c(q,\dot{q})$ groups the coriolis and centrifugal terms, and $g(q)$ represents the gravitational term. The vector τ represents only the torques on the actuated motors, and matrix A is the mapping from the relative torques to the absolute torques. When only one foot is in contact with the ground, the system is said to be single support phase and when both feet are in contact with the ground is double support phase. Note that in this situation there is a closed kinematic chain formed by the two legs and the ground. The total number of degrees of freedom in this phase is three.

B. Impact model

The transition from the single support phase to the double support phase is assumed to occur with an anelastic collision of swing leg. This event results in a discontinuity of the joint velocities described by this equation [11].

$$\dot{q}^{+} = \Delta(q^{-}, \dot{q}^{-}) \tag{2}$$

Where the superscript ($^{+}$) indicates a value immediately after and ($^{-}$) immediately before the impact. The starting point is the extended dynamic model of the system that also includes the position and velocity of the stand foot. The new model has then 7 degree of freedom and can be represented as:

$$B(x)\ddot{x} + \hat{c}(x,\dot{x}) + g(x) = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} D_1(x) \\ D_2(q) \end{bmatrix} F \tag{3}$$

Where $x = (q_1, ..., q_n, x_p, y_p)^T$ is a new state vector that includes the Cartesian coordinate's x_p and y_p of the stand foot. On the right hand side there are the joint torques T and the constant forces $F = (F_0^t, F_0^n, F_c^t, F_c^n)$ are the forces exerted by the ground on the robot. The key idea is now to integrate the motion. With this integration, all forces that are not impulsive can be eliminated. Suppose that the stand foot gets fix on the ground and to be lifted after finishing impulse, by using collision theory, the velocity will be obtained after contacting the foot.

$$x^{+} - x^{-} = \hat{B}(x)^{-1} \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \beta \tag{4}$$

That β is the integrated impulsive force.

C. Robotic linear control equation

By using four suitable outputs which are shown in Fig. 2 and *z5* which is moving of body on the *x* axis, we have [11]:

$$\begin{aligned} z_1 &= y_1 \\ z_2 &= y_2 \\ z_3 &= y_3 \\ z_4 &= y_4 \\ z_5 &= \eta(q, \dot{q}, \upsilon) \end{aligned} \tag{5}$$

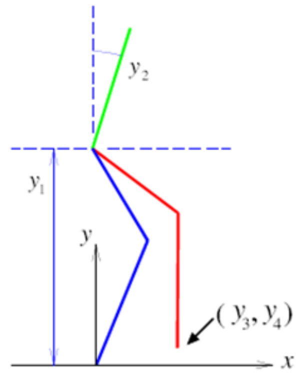


Fig. 2 Taken outputs from robot.

These values should be obtained by using absolute angles as follow:

$$\begin{aligned} y_1 &= l_1 c_1 + l_2 c_2 \\ y_2 &= q_5 \\ y_3 &= l_1 s_1 + l_2 s_2 - l_3 s_3 - l_4 s_4 \\ y_4 &= l_1 c_1 + l_2 c_2 - l_3 c_3 - l_4 c_4 \\ y_5 &= X_{hip} = l_1 s_1 + l_2 s_2 \end{aligned} \tag{6}$$

In this Equations l_i is length of each parts and $c_i = \cos(q_i)$, $s_i = \sin(q_i)$. If we derivate from above Equations to *t* will have:

$$\ddot{y} = J(q)\ddot{q} + n(q, \dot{q}) \tag{7}$$

The $\ddot{q}$ state should be obtained from dynamic model:

$$\ddot{q} = B^{-1}(q)[A\tau - c(q,\dot{q}) - g(q)] \tag{8}$$

The simpler form of Eq. 7 with substituting the above Equations is:

$$\ddot{y} = \tilde{J}(q)\tau + \tilde{n}(q, \dot{q}) \tag{9}$$

By applying this linear algorithm we express a new dynamic system:

$$\begin{aligned} \ddot{z}_1 &= \ddot{y}_1 = \upsilon_1 \\ \ddot{z}_2 &= \ddot{y}_2 = \upsilon_2 \\ \ddot{z}_3 &= \ddot{y}_3 = \upsilon_3 \\ \ddot{z}_4 &= \ddot{y}_4 = \upsilon_4 \\ \ddot{z}_5 &= \ddot{y}_5 = \varphi(z, \dot{z}, \upsilon) \end{aligned} \tag{10}$$

To obtain a linear system with static feedback position, we specify the torques value like this:

$$\tau = \tilde{J}(q)^{-1}(\upsilon - \tilde{n}(q, \dot{q})) \tag{11}$$

By choosing υ, this dynamic system will be stable and y_i will reach to the designed value to move on straight direction.

D. Motional constraint for controlling

The main duty of robot controller is ability to adjust robot's motion and speed. In this article, the time of swing foot trajectory is used for controlling walking and velocity. The velocity of walking is controlled by using of end in time of the swing foot trajectory or by ending phase of joint leg on the *x* axis direction. Therefore, an approximate simple dynamic system is assumed which is including inverted pendulum with variant length and concentrated mass in one point, as shown in Fig. 3 [12]. The pendulum mass (*m*) shows the total mass of the robot.

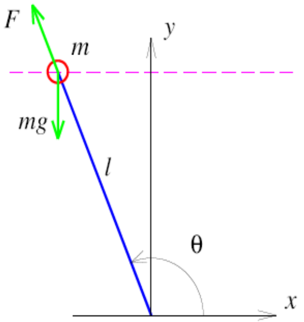


Fig. 3: Scheme of robot with assuming of concentrating mass and variant length.

If the most robot mass to be concentrated in the middle of the body, we can assume that the approximated mass of the inverted pendulum has always constant height. To keep y_1 constantly, the concentrated mass is desired to move on a parallel line with *x* axis. So we can do this by a linear motor and a control system. The motion of mass *m*, in *x* axis, is given as follows:

$$m\ddot{x} = T \tag{12}$$

Where *T* is the horizontal component of *F* that is shown in Fig. 3 we conclude:

$$\ddot{x} - \gamma^2 x = 0\ ,\ \ \gamma = \sqrt{\frac{g}{y_0}} \tag{13}$$

Where:

$$\begin{aligned} x(t) &= C_1 e^{-\gamma t} + C_2 e^{\gamma t} \\ C_1 &= \frac{\gamma x_0 - \dot{x}_0}{2\gamma} \\ C_2 &= \frac{\gamma x_0 + \dot{x}_0}{2\gamma} \end{aligned} \tag{14}$$

$\dot{x}_0 = -\gamma x_0$ is the lowest initial speed that the mass should possess to reach to the point $x = 0$ so for continuing the motion on the *x* axis, the initial speed should be more than this value.

E. Swing foot rejection of robot

We approximate the swing foot trajectory by a cubic function including 2 functions on the *y* axis and one function for motion on the *x* axis that is shown in Fig. 4 and Eq. 15 [10].

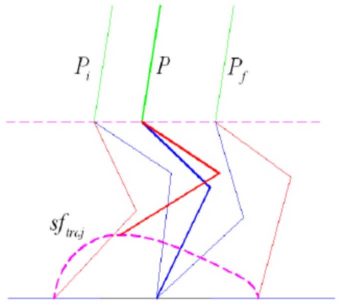


Fig. 4: Swing foot trajectory.

$$\begin{aligned} x_{sf}(s) &= a_x s^3 + b_x s^2 + c_x s + d_x \\ y_{sf1}(s) &= a_{y1} s^3 + b_{y1} s^2 + c_{y1} s + d_{y1} \qquad s \in [0, c) \\ y_{sf2}(s) &= a_{y2} s^3 + b_{y2} s^2 + c_{y2} s + d_{y2} \qquad s \in (c, 1] \end{aligned} \tag{15}$$

By using the boundary condition in single support phase these 3 functions are derived. Using robot dimensions, we can obtain the torque values in each motor. Thereafter with Eq. 11 and by applying obtained torques to the motors, robot moves on the defined direction.

III. NUMERICAL EXAMPLE

With considerate 2D robot model as shown in Fig. 5 and given dimensions and size in Eq. 16 we want to calculate the rotating angle values, angular accelerations, angular velocities and torques of each motor.

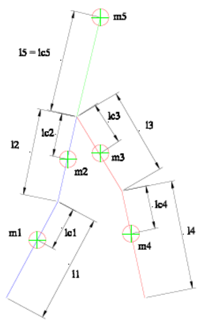


Fig. 5: Length and mass of each part in 2D model.

For doing this it needs to calculate the swing foot trajectory and body trajectory in single support phase which is used plastic contact assumption. A prepared program in Matlab-Simulink gives the trajectory of swing foot. The result is depicted in Fig.6. The numerical value of parameters are as follows:.

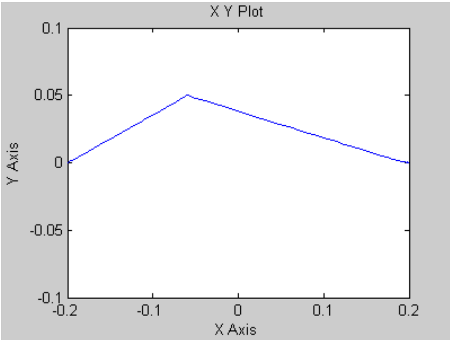


Fig. 6: The *x-y* diagram of swing foot trajectory in single support phase.

$$\begin{aligned} &m1 = 4; m2 = 4.6; m3 = 4.6; m4 = 4; m5 = 9 \\ &l1 = .41; l2 = .34; l3 = .34; l4 = .41; l5 = .67 \\ &I1 = .06; I2 = .04; I3 = .04; I4 = .06; I5 = .32 \\ &lc1 = .26; lc2 = .16; lc3 = .16; lc4 = .26; lc5 = .67 \end{aligned} \quad (16)$$

IV. OBTAINED SIMULATION

To evaluate results of motion on the straight direction the robot of Fig. 7 is considered and its motion is simulated [13,14].

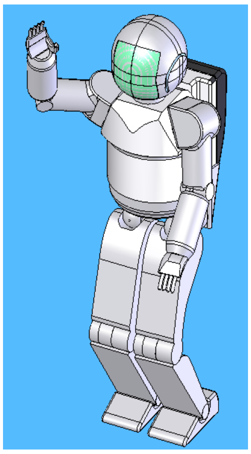


Fig. 7: Simulating 3D model of robot in the SOLID WORKS software.

The illustrated 3D model has 23 DOF which is as follows: 6 D.O.F in each leg, 4 D.O.F in each hand, 2 D.O.F in head, 1 D.O.F in waist.

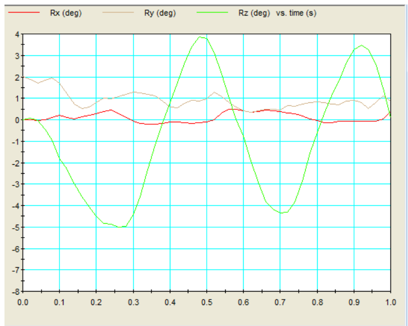


Fig. 8: The rate of rotating of robot's upper body in 3 directions of coordinate axis's.

To control the robot, joint angles of the upper body in 3 directions is obtained from motion simulation by using control systems in MATLAB software and also using feedback from the angles. The results are shown in Fig. 8.

Simulation of the robot motion on the straight direction is shown in Fig. 9 by MSC. Visual Nastran software and rotating angles, angular velocities and angular accelerations in simulating of motion of this robot are shown in Fig. 10:

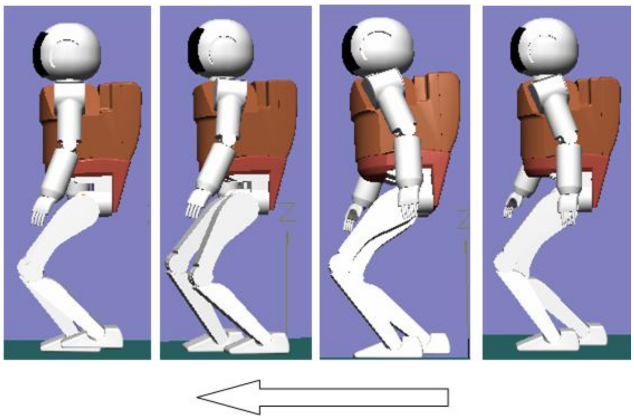


Fig. 9: Simulation of robot motion on the straight direction.

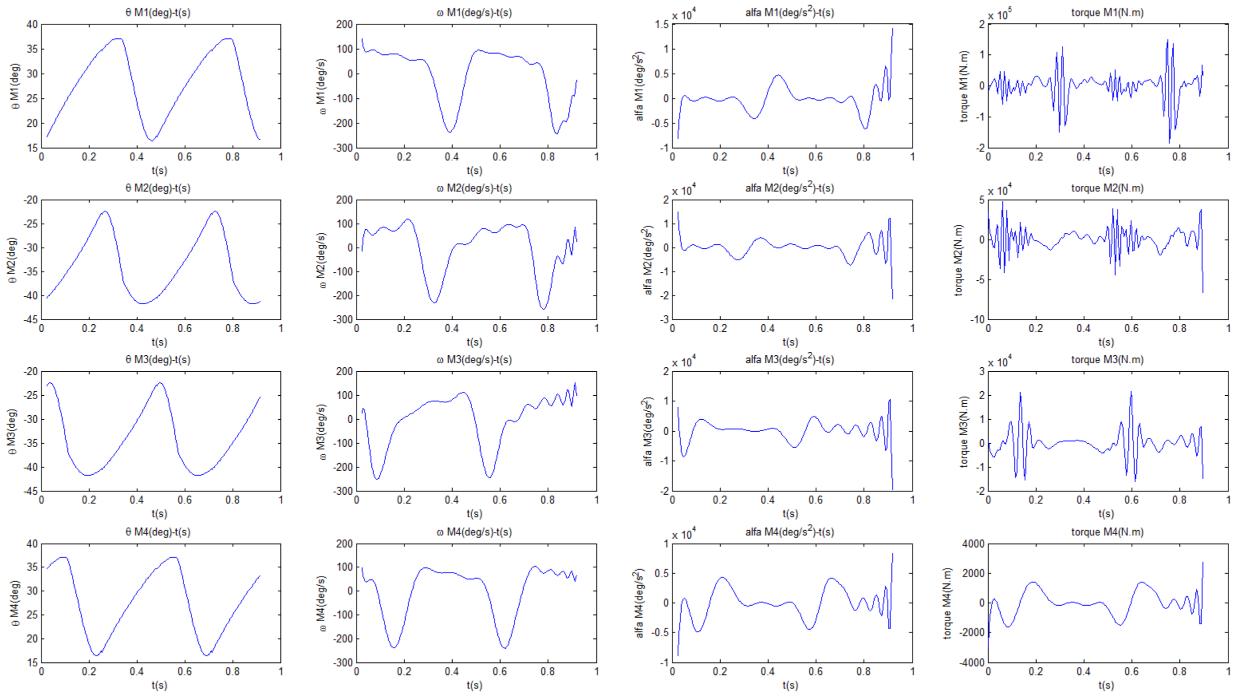


Fig. 10: Calculated values of the angles, angular velocity, angular acceleration and angular torque of the motors in 2D situation.

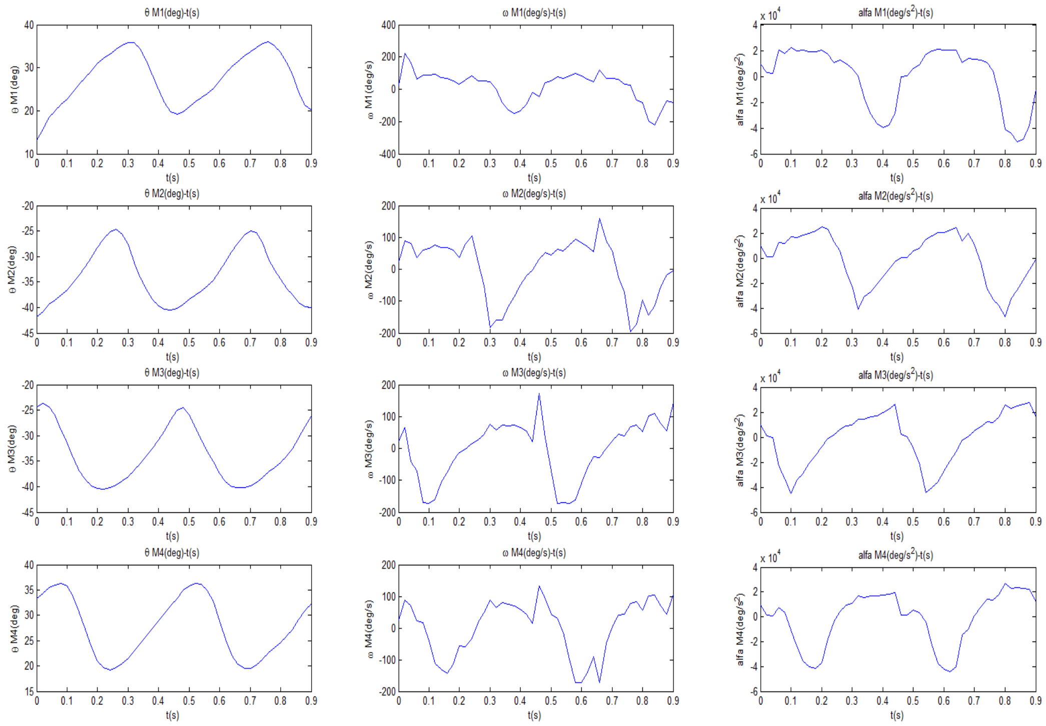


Fig. 11: Measured values of angles, angular velocities and angular accelerations in simulating of the motors in the 3D simulating situation.

V. CONCLUSION

Considering figures 10 and 11 the following results can be obtained:

The angular motion motors in 2D and 3D simulation are the same. During the contact time in simulation process there is jump discontinuity and jerk in angular acceleration of revolute joints which is due to the impulse between the ground and robot foot. Therefore the angular acceleration also takes variants to the 2D situation. The calculated torqueses are not comparable to each other because of different natures in 2D and 3D situations. Controlling the robot is done very well; during it moves on the straight direction in *x*-*y* plane and right on *z* direction. This matter is occurred because of keeping stability of the robot in single support leg of course as it is shown in Fig. 8 effect of the robot control as well as the movements of hands and feedback of upper part of the robot, the value of this revolute joint angles with respect to time can be investigated.

REFERENCES

[1] K. Yokoyama, J. Maeda, T. Isozumi, and K. Kaneko, "Application of Humanoid Robots for Cooperative Tasks in the Outdoors," *Proc. Int. Conference on Intelligent Robots and Systems*, 2001.

[2] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, "Development of a Bipedal Humanoid Robot – Control Method of Whole Body Cooperative Dynamic Biped Walking," *Proc. IEEE Int. Conference on Robotics and Automation*, pp. 368-374, 1999.

[3] K. Nishiwaki, T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, and H. Inoue, "Design and Development of Research Platform for Perception-Action Integration in Humanoid Robot: H6," *Proc. Int. Conference on Intelligent Robots and System*, pp.1559-1564, 2000.

[4] M. Gienger, K. Löffler, and F. Pfeiffer, "Towards the Design of Biped Jogging Robot," *Proc. IEEE Int. Conference on Robotics and Automation*, pp. 4140-4145, 2001.

[5] T. Furuta, Y. Okomura, and K. Tomiyama, "Design and Construction of a Series of Compact Humanoid Robots and Development of Biped Walk Control Strategies," *Proc. IEEE-RAS Int. Conference on Humanoid Robots*, 2000.

[6] K. Hirai, "Current and Future Perspective of Honda Humanoid Robot," *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 500-508, 1997.

[7] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The Development of Honda Humanoid Robot," *Proc. IEEE Int. Conference on Robotics and Automation*, pp. 1321-1326, 1998.

[8] M. Hirose, Y. Haikawa, T. Takenaka, and K. Hirai, "Development of Humanoid Robot ASIMO," *Proc. Int. Conference on Intelligent Robots and Systems*, 2001.

[9] K. Yokoi, M. Kobayashi, H. Hasunuma, H. Moriyama, T. Itoko, Y. Yanagihara, T. Ueno, and K. Ohya, "Application of Humanoid Robots for Teleoperations of Construction Machines," *Intelligent Systems Institute, AIST, Tsukuba, Ibarak* , 2002.

[10] F. Zonfrilli, G. Oriolo, D. Nardi, "A Biped Locomotion Strategy for the Quadruped Robot Sony ERS-210," 2001.

[11] F.Zonfrilli, "Theoretical and Experimental Issues in BipedWalking Control Based on Passive Dynamics," *Ph.D. Thesis in Systems Engineering*, 2004.

[12] H. Miura and I. Shimoyama, "Dynamic walk of a biped," *The Int. Journal of Robotics Research* 3, 1984.

[13] K. Kaneko, F. Kanehiro, Sh. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, Sh. Ota and T. Isozumi, "Design of Prototype Humanoid Robotics Platform for HRP," *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems,* 2002.

[14] K. Kaneko, F. Kanehiro, Sh. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi and T. Isozumi, "Humanoid Robot HRP-2," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation,* New Orleans, 2004.

Teaching Robotics at the Postgraduate Level: Assessment and Feedback for On Site and Distance Learning

Jenny Carter, Stephen Matthews and Simon Coupland
Centre for Computational Intelligence
De Montfort University
Leicester, United Kingdom, LE1 9BH
Email: jennyc@dmu.ac.uk, sgm@dmu.ac.uk, simonc@dmu.ac.uk

***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 7 years and are now in the fourth 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 assessment of students' work and the subsequent feedback given to students within the course as a whole and more specifically, the Mobile Robots module. The approaches maximise the use of electronic methods and as such there is a specific focus on those students that are studying in distance learning mode. We believe it serves as a model for others attempting to assess students studying robotics courses at a distance.**

I. Introduction

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 are delivered mainly by the members of the Centre for Computational Intelligence (CCI) at De Montfort 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.

The MScs each consist of 8 taught modules and an independent project which is equivalent to 4 modules. The 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. 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 a module titled 'Applied Computational Intelligence' enables students to pursue an appropriate area of their own interest in greater depth. In this paper we discuss recent enhancements to our approaches for assessing work and providing students with timely feedback. The full structure of the course is illustrated in Figure 1.

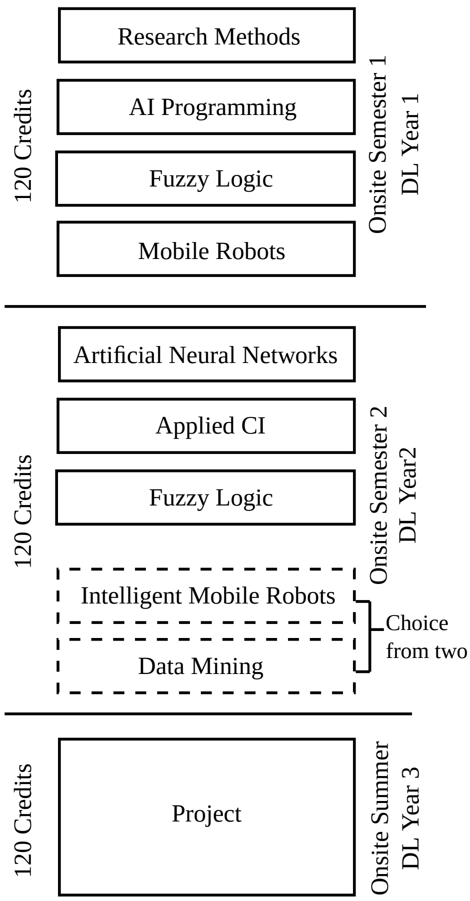


Fig. 1. Course Structure for MSc IS and ISR.

The remainder of the paper is structured as follows: Section 2 discusses the literature associated with assessment and feedback in e-learning; 3 describes the approaches to assessment and feedback that we have adopted for the course; Section 4 gives an account of student opinion regarding the recently adopted electronic approaches to feedback; Section 5 gives a detailed account of assessment and feedback within the Mobile Robots module and finally Section 6 draws conclusions from this work.

II. Assessment in Postgraduate e-Learning

This section reviews approaches to assessment and more specifically, feedback to students, in e-learning. The Quality Assurance Agency (QAA) for Higher education in the UK provides codes of practice for all types of learning. There is a section of the documentation that is aimed specifically at flexible and distributed learning and within this they include e-learning [1]. These codes of practice are observed by all higher education institutions in the United Kingdom and there are government led procedures in place to monitor their appropriate application. Precepts are stated in the QAA documentation that define what the students should be able to expect from their institution, their learning materials, their tutors and so on when engaged in flexible, distance or electronic learning. Also of particular interest for this study are those precepts that relate to assessment and feedback of student work [2].

One area of attention is that of formative feedback, where students are given feedback on their work but that feedback does not relate to any marks or grades for the course or module. One of the ways that we address this is using regular discussion board activity; this is described more fully in [3]. Another area of attention to highlight is that of plagiarism detection and prevention. We adopt various strategies for this including the use of TurnitinUK for checking authenticity, the use of vivas or presentations/demonstrations and the discussion board is also a substantial aid in both prevention and detection. In addition to this we set assignments that can be approached in a variety of ways, which reduces the opportunity for students to work too closely together. Prevention and detection of plagiarism is beyond the scope of the work presented here so will not be addressed further.

The QAA suggest that excessive amounts of summative assessment should be avoided. They state that "it is good practice to provide students with sufficient, constructive and timely feedback on their work" [2, p. 20] and this is the area that we have been addressing recently. Timing has been an issue on our course as there has been a significant delay before the students receive their marked work. Our new approach addresses this and is described fully in section 3.

As well as the need for feedback to be timely it also needs to be of a high quality in order for learners to be able to use it to determine further actions. This is identified in case study 4 of the Joint Information Systems Committee (JISC) which states that "feedback must:

- Be helpful, detailed and appropriate to learners' current understanding
- Provide more detail with each failed attempt
- Identify a means of rectifying errors
- Invite an active response. " [4, p. 1]

The report emphasises the particular importance of this with respect to distance learning students. Adding quality to feedback is also highlighted by [5] where studies are de-scribed that show that explanatory feedback resulted in improved learning compared with the effects of corrective feedback, explanatory feedback being where some explanation is given in the feedback when something is incorrect. The authors in [5] also go on to state that such explanations ideally should be succinct and positioned so that they are close physically to where the error in the students work took place. Other studies, notably [6] and [7], also promote explanatory feedback by referring to it as descriptive and emphasising how it provides useful information to enable the gap to be filled between the current student performance and the desired performance.

In order to offer good quality courses we aim to provide appropriate feedback that adheres to the codes of practise identified by the QAA and promotes students learning as described in the previous paragraph. Student numbers have grown on the courses and government spending cuts in the UK put a greater strain on the available resources which means that the course team need to increase efficiency but without losing (and whilst still improving) the quality of the provision. With this in mind, approaches to assessing students' work and providing feedback have been adapted and the new methods that are now in place on most of our modules are described in the next section. Section 5 considers the mobile robots module specifically.

III. Assessment and feedback strategies on the MSc IS/ISR

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 University already uses the Blackboard learning environment 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.

Assignments are made available to students on Blackboard and they are asked to submit them for assessment to Blackboard for electronic marking. The students submit their work twice - once to Turnitin, which checks for plagiarism and once to an assignment submission link. This work is then marked using electronic methods, and the annotated scripts with provisional marks are posted in a feedback space on Blackboard that is generated when the students submit their work. Multiple files can be uploaded to this space both by students and the marking tutors. This means that the students get feedback as soon as the work is marked.

In previous years, the second form of electronic assignment submission was not used, instead students either posted or

Marking Scheme for essay/report - worth 50% of overall mark

	0–44% Fail	**45–49% Marginal Fail**	**50–54% Pass**	**55–59% Pass**	**60–69% Merit**	**$\geq$ 70% Distinction**
Coverage of area including literature review	Not acceptable	Some attempt to cover the area but with serious limitations	Brief with significant limitations	Good coverage, but with some noticeable limitations	Very good coverage of area and associated issues with good review of literature	Excellent coverage, showing a sound understanding of topic. Excellent critical review of literature
Practical (e.g. implementation or experimental work)	Very little of value	Weak, with substantial limitations. Some effort evident.	Satisfactory amount of work. Significant limitations in design & documentation	Good work, with some limitations	Very good work, very good documentation and design. Only minor limitations	Challenging work, well documented, well designed
Conclusions, recommendations, critical evaluation, new ideas, etc.	Missing, poor or not meaningful	A minimal attempt with serious limitations. Not acceptable	Satisfactory but with significant limitations.	Godd, but with some notable limitations. Lacks depth	very good, comprehensive, with good ideas	Excellent, follows logically from body of report and contains excellent and original ideas
Structure and presentation, references and bibliography	No clear structure and presentation very weak. Poor or no bibliography, reference list, citations in report	Weak structure, poor presentation. Poor bibliography, reference list, citations in report	Satisfactory approach to structure and presentation. List of references present with significant limitations	Well structures and presentation good. Most references in correct format with both web and traditional resources	Very well structures and prepared with only minor limitations. References cited in correct notation from both web and traditional sources	Highly professional approach; excellent structure. Thorough reference citation from a variety of sources

Fig. 2. Example marking grid

physically brought in their work and handed it in to the student office. This meant that the work was marked by hand and although students were given provisional marks, they did not receive their annotated scripts until after the departmental assessment board which could be some weeks later.

The methods adopted for marking the electronic submission vary. Most tutors make use of a marking grid, an example of which is shown in Figure 2 and some staff write summary feedback to go with the annotated grid. In such cases this forms the entire feedback and can be made available quickly even when marking paper based copies of the assignment. Most tutors prefer to write comments on the students' work in addition to the use of a marking grid and it is this that has posed problems in the past for returning the feedback in a timely manner. Staff now all have Adobe Acrobat Professional installed on their computers and in addition to that they have a pen tablet (See Figure 3.). With Acrobat Professional annotations on the students work can be carried out by using the typewriter tool, or by hand-writing comments using the pen-tablet, by inserting electronic sticky notes or even by adding voice recordings. The number of different ways of adding feedback electronically enabled by providing this software and hardware has meant that all staff have adopted one of the electronic methods for semester 2 of the 2010/2011 academic year. Two modules used electronic methods for assessment and feedback in semester 1, these were Fuzzy Logic and Mobile Robots.

The following section describes a short survey undertaken to find out the opinions of the students on the use of the electronic methods in the Fuzzy Logic module of semester 1 and the Applied CI module of semester 2. Section 4 considers the assessment and feedback mechanisms adopted in the Mobile Robots module in greater depth.

Fig. 3. Pen Tablet

IV. Results of Student Survey

Thirty-five students were emailed that studied either the Fuzzy Logic module or the Applied CI module (or both) to give their opinions of the new method of feedback compared to the previous. The questions asked are contained in Figure 4. Students responded and a discussion of the results is given below.

Approximately one third of the students responded and all of them had the same responses (a.) for questions 1, 2, 3. This leaves us in no doubt that the electronic marking is an improvement. We have yet to investigate if there are preferences between the methods used though allowing staff to choose from a selection of methods has encouraged staff to move to a new form of marking and has resulted in a much greater take-up of the new approaches than might have been achieved if only one approach had been allowed.

Interestingly there was a mixture of on-site and distance learners amongst the responders. This demonstrates that even for on-site students, such methods are an improvement even though they would previously be able to collect work physically from the marking tutors. The comments collected in response to question 5 were generally supporting the answers in questions 1–3, though one person commented that hand writing had been an issue in some place on one assignment where the pen tablet had been used. The next section examines the Mobile Robots module and the specific approaches taken in that module to assessing and feeding back on the students' work.

V. 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

The questions below relate to the electronically marked assignments for the Fuzzy Logic & Applied CI modules.

1) Compared to the usual methods of receiving marks and feedback, was the speed of receiving the electronically marked work:
   a) noticeably quicker?
   b) the same?

2) Was the quality of the feedback:
   a) better?
   b) worse?
   c) the same?

3) We plan to extend the use of this so that eventually it will eliminate the need to physically post your assignments to us (you will be able to upload multiple files to Blackboard including program files etc.). Do you see this as:
   a) good thing?
   b) a bad thing?
   c) neither good nor bad?

4) Are you studying as:
   a) a distance learning student?
   b) an on-site student?

5) Do you have any additional thoughts/comments you would like to add regarding electronic marking?

Fig. 4. Survey used for Fuzzy Logic and Applied CI modules, and also the Mobile Robots module.

robotics module. Our strategy is depicted in Figure 5 and fully explored in [3].

Nineteen students on the Mobile Robots module were emailed the same survey content shown in Figure 4. Fewer students responded, approximately one sixth, and the survey results are positive and have provided promising feedback from both on-site and distance learning students.

From the responses, the majority thought the speed of receiving feedback was noticeable quicker in question 1. This included both on-site and distance learning students, which supports the responses to the Fuzzy Logic and Computational Intelligence modules. The importance was noted on one survey that this greatly helped to prevent making the same mistake for the next assignment. This is crucial since some of the assignments were issued on a weekly basis.

The quality of feedback was deemed to be the same according to the responses for question 2, although, in one case it was considered better. This module uses physical robots that distance leaning students have at home. On-site students often discuss the electronic feedback with tutors in laboratories where it is possible to provide a physical demonstration to further support and clarify feedback. When a distance learning student wishes to clarify feedback in more detail, the current approach is for them to request a phone conversation that is typically conducted with VoIP and without a physical demonstration. For this reason it is important to carefully consider the content of electronic feedback, particularly for the distance learning students who can not share a physical demonstration in the same way as the on-site students do.

To overcome the difficulties with physical demonstrations of the robot for the purposes of feedback there are several options that are under consideration for the next cohort. The VoIP call could be enriched by also using video for communication with a webcam. This would allow both staff and students to perform robotic demonstrations during a discussion to enhance the conversation and support the understanding of the feedback. For software demonstrations that do not involve the robot there are a variety of software packages to share the view of a

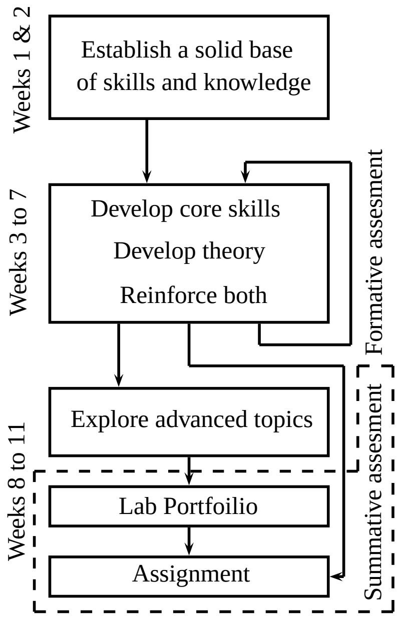


Fig. 5. Teaching and Assessment Strategy for Mobile Robots.

computer desktop with another computer.

For question 3, all respondents agree that this method of electronic feedback is a good thing. One respondent's comment was that this was very much a positive move and a step in the right direction.

An Interactive questions and answers session was held in the last quarter of the module. Students were invited to email questions one week before the session that cover any aspect of the Robotics material studied up to that week. These questions are then collated to form the structure of a lecture which answers the questions. This lecture is videoed and posted on the Universities streaming site for the students to access.

VI. Conclusions

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 approaches taken to assessment and feedback on 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 assessing of students work and providing timely informative feedback to students. We believe that by following this model and the delivery model identified in [3] it is possible to deliver and assess a technical, practical subject by distance learning and 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 64 students currently enrolled

(5 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] The Quality Assurance Agency for Higher Education. (2010) QA codes of Practice, Section 2. [Online]. Available: http://www.qaa.ac.uk/academicinfrastructure/codeofpractice/default.asp

[2] ——. (2006) QA codes of Practice, Section 6. [Online]. Available: http://www.qaa.ac.uk/academicinfrastructure/codeofpractice/default.asp

[3] Carter, J. and Coupland, S., “Teaching Robotics at the Postgraduate Level: Delivering for On Site and Distance Learning,” in *Proceedings of the International Conference on Robotics in Education*, 2010.

[4] Joint Information Systems Committee. (2010) Designing interactive assessments to promote independent learning. [Online]. Available: http://www.jisc.ac.uk/media/documents/programmes/ elearning/digiassess_interactiveassessments.pdf

[5] Clark, R. and Meyer R.E., *e-learning and the science of instruction*. Wiley, 2008.

[6] Nicol, D. J. and Milligan, C., “Rethinking technology-supported assessment in terms of the seven principles of good feedback practice,” in *Innovative Assessment in Higher Education*, C. Bryan and K. Clegg, Eds. London: Taylor and Francis Group Ltd, 2006.

[7] Wiggins, G., *Educative Assessment*. San Francisco: Jossey-Bass, 2001.

Teaching BOTBALL and researching DISBOTICS

Gottfried Koppensteiner
Institute of Automation & Control
Vienna University of Technology
Vienna, Austria
Email: koppensteiner@acin.tuwien.ac.at

Munir Merdan
Institute of Automation & Control
Vienna University of Technology
Vienna, Austria
Email: merdan@acin.tuwien.ac.at

David P. Miller
Schools of AME & Computer Science
University of Oklahoma
Norman, Oklahoma, USA
Email: dpmiller@ou.edu

***Abstract*—The enrollment in STEM fields (science, technology, engineering and math) is not keeping pace with the need, especially in the robotics sector. The university level is often too late for someone to start their engineering education and therefore universities must recruit students well before they are about to start university study. This paper shows how to bridge the gap between research and high school education using BOTBALL combined with an actual research topic: The disassembly of goods with autonomous mobile robots. This paper is based on an extensive cooperative experience between the Vienna University of Technology and the Vienna Institute of Technology (TGM) and a successful first BOTBALL Season with the students of the TGM. It shows the possibilities of BOTBALL, the influence of other courses and a way to start with robotic beginners and end with research experts by the example of the DISBOTICS Project.**

I. Introduction

Throughout much of the world, there is a shortage of skilled engineering talent. Enrollment in the STEM fields (science, technology, engineering and math) is not keeping pace with the need. In addition, women and several other important populations are significantly under represented generally in engineering and especially in computer science. STEM fields require several years of preparation prior to being ready to start university study in one of the STEM disciplines. The earlier in one's academic career they realize they might wish to study engineering, the easier it is for them to prepare (by taking the appropriate math and science courses while in elementary and high school) and therefore the more likely they will be successful in their endeavors. Since the university level is usually too late for someone to start their engineering education, and since there is currently a shortage of students in these fields of study, universities must recruit students and they must recruit them well before they are about to start university study. University outreach programs that start in high school or earlier can have success in encouraging those students to pursue engineering when they get to university. Additionally, personal connections are made between university staff/faculty and future students, which increases those students' likelihood of success and allows the university to work with especially promising students over an extended period of time.

Robots have been playing an important role in education since the advent of the LOGO Turtle [20]. Robotics is a popular, interesting and effective way for teachers as a teaching tool for introducing students to important areas of Science, Technology, Engineering and Maths curricula [8] [21]. It can promote development of systems thinking, problem solving, self-control, and teamwork skills. Involvement of students in a robot contest can offer additional educational benefits [8] [30] [1] [6]. Today, it is of the utmost importance that computer, electrical, control and mechanical engineering university program studies include the teaching of both theoretical and practical courses on robotics. Contrary to traditional technical education strategies that tended to promote individualism and competence between students, nowadays engineering challenges in most areas, and especially robotics, requires working with multidisciplinary teams in order to successfully integrate different areas of knowledge. Practical work on robotics at university level can help engineering students to develop the needed communication and working skills for teamwork [28]. Mobile robots can be used as a motivating and interesting tool to perform laboratory experiments within the context of mechatronics, microelectronics and control at university. Students can study mobile robot design and integration tasks at different levels of complexity. Small mobile robots also allow the students to perform interesting experiments [24]. The current field of robotic educational endeavors is extremely large and diverse; see [5] [17] for an overview. In this paper we present a research project about mobile robots, where both students of a technical high school as well as researchers work together on a common goal, using Botball as tool to bridge the gap and to transform beginners into experts. Therefore this paper is structured as follows: The second chapter gives a short introduction about the research project DISBOTICS, the third chapter points out the educational point of view, explaining Botball. The idea of an Austrian Botball Season is described in the fourth section and finally conclusions are presented.

II. The DISBOTICS project

A. Scientific Aspects

Mobile robotics is an open and challenging field providing practitioners with useful perspectives. There is a large variety of indoor as well as outdoor applications that can help to improve industrial production and some aspects of the workers life quality [26]. Very interesting approaches and applications are being developed that, in the medium term, can be part of our daily life; see for example [22]. The DISassembly roBOTICS project researches the usage of mobile robots for the disassembly process of goods. Disassembly has become a vital industry process due to the increasing necessity of optimizing resource usage [9] [27]. Mobile robots for disassembly should be (a) intelligent in the sense of path planning and able to communicate with other robots, (b) cooperative with other (stationary or mobile) robots, and (c) able to form a disassembly multi agent system, which is one of the future possibilities for reducing disassembly costs.

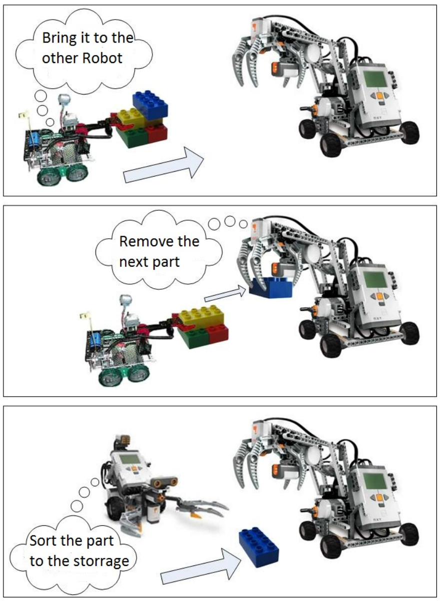


Fig. 1. DISBOTICS Concept

To cope with these requirements, the DISBOTICS project proposes a knowledge-intensive multi-agent robot system. This enables ontology-based communication and cooperation among a set of autonomous and heterogeneous units - agents. Each agent supervises one particular mobile robot and, related to the robot's skills, has its own objectives and knowledge. In this context, ontologies allow the explicit specification of an agent's domain of application, increasing the level of specification of knowledge by incorporating semantics into the data, and promoting knowledge exchange between agents in an explicitly understandable form [16]. An ontology based product model is used [9] to link product designs, disassembly planning and scheduling processes, as well as required disassembly equipment, possessed by a particular mobile robot, in a way that enables automatic reasoning as well as wide data integration. Consequently, on the one side, a vision system can use this model to reason about the content of a captured image. On the other side, an agent controlling a mobile robot can extract required disassembly information from this model to select and perform the necessary actions. The architecture is based on agents that have a rule-based behaviour. Rules are considered as if-then statements applied to the knowledge base. The application of this kind of decision-making mechanism supports a knowledge capture in a more modular and explicit way.

The other aim of the project is to bring more students to the field of robotics, as mentioned in the Introduction. Therefore it is necessary to use robots or even controllers which are well-suited for a collegiate robotics lab and simple enough to be used by young students. The CBCv2 [18], developed for current robotic tournaments and education within the Botball program is being used by thousands of middle and high school students during their education in robotics. It includes an ARM 7 based DAQ/Motor control system, an ARM 9-based CPU/Vision processor running LINUX, an integrated color display as well as a touch screen. The touch screen interface and extensive robot function libraries make the CBCv2 easy for students to use [19]. The embedded Linux and reloadable firmware allow additional software paradigms to be added allowing the CBCv2 to be enhanced with agent and rule based systems. The CBCv2 is a USB host (allowing the use of standard cameras, mass storage and network interfaces) and can also be used as a USB device for software downloads. At the USB port a Wi-Fi Stick can be used; this provides a good possibility for the communication between mobile robots and therefore it is good enough to meet the requirements for an intelligent mobile robot control as described in the previous chapter.

B. Research Aspects

Due to the high degree of complexity of the proposed system we took a step-by-step approach and at the beginning we divided the project into 5 sub-projects, each of which deals with one particular task and consists of one researcher and a diploma-thesis project group from the technical high school, normally up to 5 students:

1) Agent-based robot control - Autonomous mobile robots perform an action based on their goals. It is able to perceive the environment through sensors and act on it with effectors. Based on its responsibilities and observations, an agent has to constantly make decisions that again could influence the environment as well as its state. Each agent in our architecture has an ontology-based world model, the role of which is to maintain the knowledge about the agent's own activities in relation to its environment as well as to its underlying software parts. The ontology specifies the meaning of terms which are used during communication, enabling knowledge inter-operations between agents. Taking into consideration the real world conditions, where particular actions have to be performed under real-time constraints, our architecture divides the control of a robot into two parts. The High Level Control (HLC) is to control the global behavior of the agent responsible for achieving their own goals and to ensure coordination with other agents of the system, so they know about the global state of the system requires. The Low Level Control (LLC) is responsible for the direct control of the physical components (actuators) and those actions that must be performed in real-time. Furthermore, the LLC is responsible for simple diagnostic tasks regarding the hardware.

2) Vision System - The vision system has two basic tasks within the disassembly process: a) it has to determine the product as well as its components; and b) it has to locate and define the next component to be disassembled. For the first process the vision system needs to recognize objects and components. Given uncertain input from used, dirty and partially missing components, the challenge is to detect features, groups, and parts and match them to the product models. It is linked to the knowledge base of the agent by integrating the work piece ontology and related perception algorithms and behaviors in the knowledge base.
3) Navigation - The LLC is responsible for receiving sensory data from the vision system, sensors and their interpretation. The HLC controls complex robot actions by means of the ontological representation of the environment, which includes defined classes such as view, action, topology, etc. *View* is defined as a symbolic abstraction of the sensory input, which is related to the robot location at a given moment and is used to derive position and orientation of the robot as well as resulting actions. An *action* is performed as a reaction of the agent to a specific state of the robot at a given view and can be for example to stop, move forward, move backward turn, and so on. The *topology* includes representations of places, paths and objects, with their associated relationships and constraints. A *place* defines a possible residence location of a robot. A *path* is a one-dimensional subspace that leads from one place to another. An *object* is defined as a three-dimensional device, which may be attached to one place or on the path to a destination, such as tables, chairs, etc. The topological representation can then be used as a map of the area, consisting of groups of objects and their pathways. The navigation should therefore be able to identify objects and the resulting restrictions on the path, and respond accordingly, for example by changing the speed or direction.
4) Grasping and Manipulation - Planning grasping and manipulation activities is often a very difficult process due to unknown or limited movement space, different options how to move and reach an object, and different object types and properties that can occur in normal environments. To handle a specific work piece correctly, a robot needs to consider the information about its position, orientation, dimension, and type of required operational activity e.g. turn or move, as well as the robot's kinematic constraints. This requires path planning strategies that incorporate the ontological and vision-based information.
5) Basics of the Disassembly Process - The conception, planning, and implementation of a disassembly system using mobile robots is a complex task. The success rate of automated disassembly will primarily depend on a sophisticated disassembly plan as well as design of mobile robots and related tools for the disassembly process. The following requirements should be considered and fulfilled: efficient and effective execution of operations, high functionality and accuracy, minimum error rates of the equipment, safety, etc. Since all of these tasks require many resources, a group is especially oriented on the basics of the disassembly process.

C. Technology Aspects

In order to validate our approach, we implemented the knowledge-intensive multi-agent robot system presented above. The overall system has been built on top of the Java Agent Development Environment (JADE) framework [7]. The JADE platform enables each agent to manage its own life cycle, register its services, search for agents providing particular services, discover them and communicate with related agents. The JADE architecture enables agent communication through message exchange based on the agent communication language (ACL) [4]. We have used Protégé [25] as an integrated software tool to develop the knowledge base. The reasoning is implemented using the Jess expert system shell (JESS) [23]. JESS is a tool used for building the rule-based expert systems, which can be seen as a set of rules that can be repeatedly applied to a collection of facts about the world. Rules are simple statements that consist of an if-part and a then-part. When the particular input information, which is coming from the environment, matches the facts in the if-part of the rule, particular actions defined in the then-part are executed. JessTab [3] is used as a plug-in for Protégé that allows us to use Jess and Protégé together.

III. Educational Point of View

From the educational point of view it is necessary to attract the young students with simple and easy to understand tools and to be able to teach them step by step common as well as research technologies, such as those used by diploma projects in high schools and universities. It is necessary to draw a line from the very basics for beginners to the research topics for the diploma thesis sector at (technical) high schools. In cooperation with the Institute of Technology this will happens in four stages through the use of the Botball Education Program. The Institute of technology is a Federal Higher Technical Institute for Educating and Experimenting, a technology and crafts orientated higher college in Austria and specialises in engineering disciplines such as civil engineering, electronics, electrical engineering, information technology, and material grade technologies. It is possible to start this type of course after compulsory education and lasts for five years. Students may complete the school via a Diploma Thesis (with normally less research percentage compared to a diploma thesis at a university), or a final examination project. After that a student has to pass one written test in mathematics and one in a language (German or English) and the final oral examinations to graduate an to be formally enabled to attend university.

A. The BOTBALL Program

Botball is an engineering outreach program for pre-university students (typically aged 12-19 years) organized world-wide by KISS Institute for Practical Robotics - an NGO headquartered in Norman OK USA. Botball teaches general engineering with an emphasis on design, documentation, mechanics and programming skills. The Botball game changes every year keeping it fresh and challenging for both new and established teams.

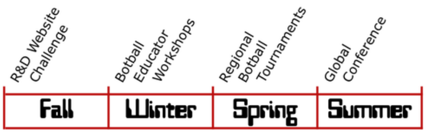


Fig. 2. The Botball Season, [10]

Groups that participate in Botball use a standard kit to design build and program a team of autonomous robots to play in that year's game. The kit includes thousands of parts - so there are almost infinite possibilities, but everyone starts from the same place. Teachers and student leaders participate in a workshop to learn about the tools and technology used in Botball. They then have a build period of approximately two months to create their robots. Documentation assignments turned in throughout the build-period help keep the teams on target and on schedule. All of the teams in the region come together at the end of the build period for a tournament that is both individual team performance and a head to head competition. The tournament activities are used along with the documentation scores to determine event and overall awards. While the Botball task changes each year, it always involves many possible activities requiring different skills and robot capabilities. The 2011 contest used an airport theme (Figure 3); the robots were tasked to tow planes from the hangar to the runway (navigation), sort and transport luggage (color sorting), move biofuel stock to fermentation tanks (object manipulation) and complete airport construction projects (block stacking). The robots are always autonomous with all computation and power onboard.

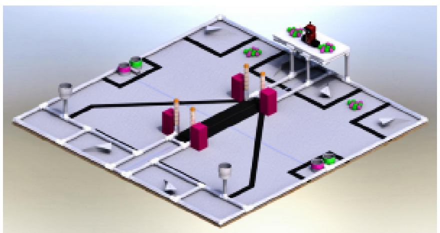


Fig. 3. The Botball Game Table: Botville Airport [11]

While KIPR's software tools support robot programming in C, Botball students have created additional tools for programming Botball robots in C++, Java, Python, Lua and additional languages. These tools and others are often presented and released during the Global Conference on Educational Robotics (GCER). All members of all teams are eligible to participate in GCER and the International Botball Tournament (which is held at GCER). The GCER paper sessions are largely populated by student papers on robotics work related to or beyond their Botball entries. Past papers have discussed the creation of new programming tools, navigation strategies, and original robotics reserach projects on topics such as learning or SLAM. GCER also hosts the KIPR Open, a tournament open to all but specifically targeted for teachers, mentors and Botball alumni, who are not eligible to participate in the standard Botball tournament.

B. Combination of DISBOTICS, Botball and High School Education

The four stages mentioned at the start of this section could be briefly described with Interest-Education-Competition-Research. These four stages should bring the young scientists from TGM smoothly from the very beginning in robotics to the research field in DISBOTICS.

1) Interest: At the very beginning the students (ages 14-15) built up their first simple tiny bot in their electronic lessons to get in touch with an easy level of robotics. This robot consists of a body and a batteries along with a simple electronic circuit with two motors and two light-transistors (Figures 4 and 5). The robot is an 'intelligent' light follower, strongly derived from Braitenberg's simple robot [2] and the BYO-Bot [12]. At this stage they will learn a lot about electrical and mechanical aspects. Typically this activity will be situated at the end of their first year of electrical education.

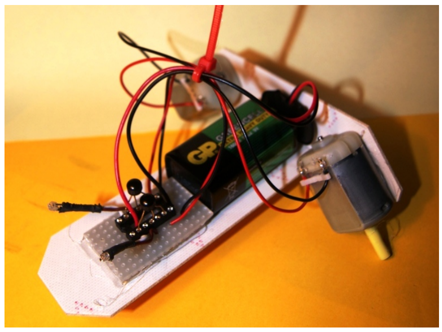


Fig. 4. The Tiny Bot

Fig. 5. A Tiny Bot in Action

2) Education: The next academic semester (September till January) the students are exposed to the technologies used in Botball. The educational repertoire includes programming guides for the C-language (for beginners), teaching them about vision systems as well as sensors, motors, and the CBC controller. An open showcase that utilizes the game-board and previous year's robots (Figure 6) helps the students gain hands-on experience and master these technologies.

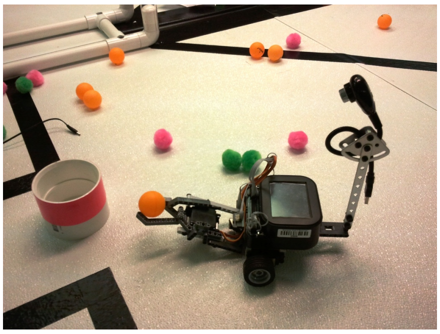


Fig. 6. The Demo-Bot 2011

3) Competition: In the second semester different groups will take part in a competition at the Institute of Technology (Figure 7) in Vienna around April and the winners receive support to attend the Global Conference on Educational Robotics and to represent Austria within this tournament in July. They should develop robots to solve the exercises of the game board and therefore they will have up to three months to find a good strategy, to build up and program their robots for this strategy. Around January the game-set and rules for that year are announced and from that point on the actual game board is ready for the student's first test runs.

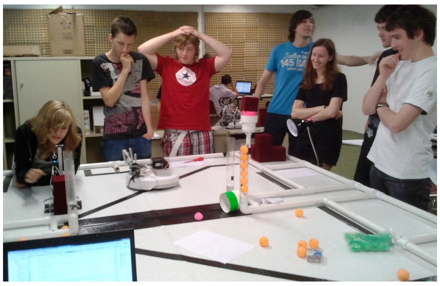


Fig. 7. Students at the game table

4) Research: Using what they have learned in Botball the students can start working on more research oriented projects such as a multi-agent system for DISBOTICS. To facilitate the design of multi-agent control systems, a generic agent architecture [13] [15] [29] was developed. This architecture clearly separates the control software into two layers: the high level control (HLC) and the low level control (LLC) as already explained in the previous chapter. The LLC layer is in charge of controlling the hardware directly. It is responsible for performing all necessary operations in real-time and is based on the IEC 61499 standard [31]. The HLC layer is responsible for more complex tasks such as coordination, monitoring or diagnostic, using multi agent technologies, ontologies, and rule based systems and is implemented on the CBCv2. This allows students to use the same controller and learn actual research technologies in a well-established environment. In cooperation with researchers of the Vienna University of Technology they work on diploma projects in all possible topics for DISBOTICS. If they succeed with good research results they have the possibility to write a paper together with the researchers and present their work at GCER or other international conferences. This year a diploma project group succesfully implemented the HLC-Architecture on the CBCv2 Controller and wrote a paper about it [14], which was presented at the 2011 GCER.

C. *Influence on other subjects*

As described in the introduction, educational robotics brings technology in an easy way to young pupils. This research project brings science into the schools and this has a deep influence on subjects like electronics, English, project management, mechanics and construction as well as math and physics. The knowledge learned by the students in the electronics lessons is used to design and build up their first simple robot. That's a good motivation and enhances - in a pupils view - the status of the subject, so it makes sense to learn electronics. This turns out to be true even for students in informatics or mechanics, who normally are not that interested in electronics. Due to the in English written documentation of the controller, game board, and coding examples the students are forced to read and write English. The online community provides the possibility to get in contact with native speakers in America. The requirement of English documentation within the Botball program and the possibility of writing and presenting a scientific paper at GCER are strong motivation to good students. All these aspects lead to a better understanding and a further usage of the English language outside the classroom.

The curriculum for technical high school education requires a diploma project for students to reach their final exam. Normally this is done in project groups and supported by its own course. This course teaches all kinds of project management as well as social skills. The addition of Botball into the curriculum provides opportunities for the students to gain a deeper knowledge of the technologies and subjects as well as gaining additional project experience.

A robot's behavior is controlled by the design and interaction of its mechanics and software. Robots provide use cases for teaching mechanics or construction and finally leads to a better understanding of why things behave in a special way. Nearly everything in robotics has to be calculated: the radius for a curve; the speed or the time for special operations; the torques of wheels; and the paths of the robot. All these things give opportunities to enhance the mathematics and physics courses. Finally the development of a robot requires a team of students with a diversity of skills. A good team will consist of creative students, students which are good in mathematics or English, good in mechanics, electronics or programming. In Austrian technical high schools, which often have more than one faculty, this is a big opportunity to bring students of different educational departments together to have one common aim and to represent their school as one team.

IV. Austrian Botball Season

Due to the good feedback of pupils, teachers and researchers about the project, the combination with research topics and good impressions about a first small internal Botball competition at the Vienna Institute of Technology, an Austrian competition and conference is planned for the season 2011/2012. The complete season will follow the Botball season in America. It will start with an open showcase in September 2011 and will bring an Austrian workshop at the Vienna University of Technology for every project team to learn the basics about programming the CBC as well as information about the new game board and the rules in February next year. After 2-3 months the European Botball Conference on Educational Robotics (EBCER) will take place. This conference should be the host for the Austrian Competition; this means the regional tournament which supports the winner to go to GCER12 to represent Austria. Furthermore, it should give project teams the opportunity to talk about their robots, research aspects, and their experience in special sessions and in discussion with other teams. Some Botball-Teams from America and Poland will also participate at this Conference. It is also planned to invite researchers from all over the world, to give scientific talks about robotics to pupils.

V. Conclusion

The project DISBOTICS shows the possibility to bridge the gap between research and high school education. Mobile robotics in combination with the disassembly process offer lots of interesting topics for diploma projects with an actual research background. The build process of the tiny bot has led to unexpected creativity of the pupils; many of them improved their robots in different ways. Parents reported an enthusiasm in learning they've never seen before in their kids. The same pupils would like to learn more about robotics and they now plan to do a diploma project with robotics with a research background. The first diploma project in this field together with students of Institute of Technology has led to a research paper about the implementation of the agent based system on the CBCv2 and has shown the possibility to bring high school students into the field of research. They presented their research work, which is a cornerstone in the DISBOTICS project, at an international conference and received an award for it. BOTBALL is evaluated as a good opportunity to interest young students in robotics and to educate them in many aspects of robotics. The first Botball tournament at the Vienna Institute of Technology has shown a big positive impact for the education of pupils and has shown a huge motivation to learn about robotics and new technologies together with other pupils as a team. We are looking forward to proceed with other research topics and to provide more students with the possibility of participation within the first Austrian Botball Tournament in 2012.

Acknowledgment

The authors wish to thank the KISS Institute of Practical Robotics for their support and the Austrian Federal Ministry of Science and Research (BMWF) for their financial support with the programme Sparkling Science. A big thank goes to the first participants of the Austrian Botball program, espacially to the students Clemens Koza, Christoph Krofitsch, Manuel Parg and Antonio Prvulovic as well as to their teacher Michael Matinides.

References

[1] D.J. Ahlgren *Fire-Fighting Robots and First-Year Engineering Design: Trinity College Experience.* Proceedings : ASEE / IEEE Frontiers in Education Conference, Oct 10-13, Reno, NY, 2001.

[2] V. Braitenberg, *Vehicles: experiments in synthetic psychology*, Book, MIT Press, 1986.

[3] H. Eriksson, *The JESS TAB Approach to Protg and JESS Integration.* In Proceedings of the IFIP 17th World Computer Congress - TC12 Stream on Intelligent Information Processing, pages 237-248. Kluwer, B.V., 2002.

[4] Foundation for Intelligent Physical Agents.*FIPA Communicative Act Library Specification* http://www.fipa.org/specs/fipa00037/ and FIPA ACL Message Structure Specification http://www.fipa.org/specs/fipa00061/, 2003.

[5] T. Fong, and I. Nourbakhsh, and K. Dautenhahn, *A survey of socially interactive robots: Concepts, Design and Applications.* Technical Report CMU-RI-TR-02-29, Carnegie Mellon University, Pittsburgh, PA USA, December2002.

[6] H. Goh, and B. Aris *Using Robotics in Education : Lessons Learned and Learning Experiences*, 1st International Malaysian Educational Technology Convention, 2007

[7] Telecom Italia Labs, *JADE - Java Agent Development Framework*, http://jade.tilab.com/, Accessed March 2011.

[8] J. Johnson, *Children, Robotics and Education. in: Artificial Life and Robotics.*, 16-21, Springer, 2003.

[9] H. Kim, S. Kernbaum, and G. Seliger, *Emulation-based control of a disassembly system for LCD monitors.*The International Journal of Advanced Manufacturing Technology, 40(3), 383-392, 2009.

[10] KISS Institute of Practical Robotics *The Botball Season* http://www.botball.org, Accessed May 2011.

[11] KISS Institute of Practical Robotics *The Botball Game Table 2011: Botville Airport* http://exploring.external.lmco.com/Robotics2011/Botball/index.html, accessed June 2011

[12] KISS Institute of Practical Robotics *Hand-Assemble BYO Bot* https://bbstore.mycafecommerce.com/product/hand-assemble-byo-bot, accessed July 2011

[13] G. Koppensteiner, M. Merdan, W. Lepuschitz, I. Hegny, *Hybrid Based Approach for Fault Tolerance in a Multi-Agent System*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM2009, Singapore; 2009.

[14] G. Koppensteiner, M. Merdan, C. Koza, C. Krofitsch, M. Parg, and A. Prvulovic, *Knowledge based agent architecture for high level control of mobile robots* in Proceedings of the 2011 Global Conference on Educational Robotics, 2011.

[15] M. Merdan, M. Vallee, W. Lepuschitz, and A. Zoitl, *Monitoring and diagnostics of industrial systems using automation agents*, International Journal of Production Research, vol. 49, no. 5, p. 1497, 2011.

[16] M. Merdan, *Knowledge-based Multi-Agent Architecture Applied in the Assembly Domain.* PhD Thesis, Vienna University of Technology., 2009 http://www.ub.tuwien.ac.at/diss/AC05040230.pdf.

[17] D.P. Miller, I.R. Nourbakhsh, and R. Siegwart, *Robots for Education* Springer Handbook of Robotics (B. Siciliano and O. Khatib, eds.), ch. 55, Springer, 2008.

[18] D.P. Miller, M. Oelke,M.J. Roman, J. Villatoro, and C.N. Winton, *The CBC: A LINUX-based low-cost mobile robot controller* Robotics and Automation (ICRA), 2010 IEEE International Conference on , vol., no., pp.4633-4638, 3-7 May 2010

[19] G. Mitsuoka, *CBC v2 controller: brings ease of use as well as speed, power and vision to small robots* Robot Magazine, pp. 8285, July/August 2011.

[20] S. Papert and I. Harel, *Situating Constructionism, in: Constructionism*, Ablex Publishing Corp., 1991.

[21] B. Perteet, *A Multi-Vehicle Framework for the Development of Robotic Games : The Marco Polo Case.* Unpublished Master's Thesis, Oklahoma State University, Stillwater, 2005.

[22] T. Pilarsky, and M. Happold, H. Pangels, M. Ollis, and A. Stenz A, *The Demeter system for automatic harvesting*, Autonomous Robots 13, pp. 9-20, 2002.

[23] Sandia National Laboratories, *Jess: the Rule Engine for the JavaTM Platform.*, Available at: http://herzberg.ca.sandia.gov/, last visited May 2011.

[24] K. Schilling, and H. Roth, *Mobile Mini-Robots for Engineering Education*, Global J. of Engng. Educ., Vol.6, No.1., 2002

[25] Stanford Medical Informatics, *Protégé Ontology Editor*, Stanford University. Protégé Website. http://protege.stanford.edu., Accessed May 2011

[26] A. Stentz, *Robotic Technologies for Outdoor Industrial Vehicles*, in Proc. of SPIE AeroSense 2001.

[27] F. Torres, et al., *Automatic PC disassembly for component recovery.* The International Journal of Advanced Manufacturing Technology, 23(1), 39-46, 2004

[28] J.M.M. Tur, and C.F. Pfeiffer, *Mobile Robot Design in Education. A Modular Low Cost Platform*, Robotics & Automation Magazine, IEEE , vol.13, no.1, pp. 69- 75, March 2006

[29] M. Vallee, H. Kaindl, M. Merdan, W. Lepuschitz, E. Arnautovic, and P. Vrba, *An automation agent architecture with a reflective world model in manufacturing systems*, in IEEE International Conference on Systems, Man, and Cybernetics (SMC09), San Antonio, Texas, USA., 2009.

[30] I.M. Verner, and D.J. Ahlgren *Robot Contest as a Laboratory for Experiential Engineering Education.* Journal on Educational Resources in Computing, 4(2), 1-15, 2004.

[31] A. Zoitl, *Real-Time Execution for IEC 61499.*, USA: ISA-o3neidaA, 2009, no. ISBN: 978193439-4274.

Kindergarten Children Programming Robots: A First Attempt

Kerstin Stoeckelmayr, Michael Tesar, Alexander Hofmann

Department of Computer Science, University of Applied Sciences Technikum Wien
Hoechstaedtplatz 5, A-1200 Wien, Austria

kerstin.stoeckelmayr@technikum-wien.at

michael.tesar@technikum-wien.at

alexander.hofmann@technikum-wien.at

***Abstract*— Using robots to teach programming is a method to enthuse young people about computer sciences. They are applied by colleges as well as by schools. To rouse young people´s enthusiasm for technology, the Department of Computer Science – University of Applied Sciences Technikum Wien offers robotic classes at a kindergarten for pre-school children (aged five to six years). Simultaneously, they are given an understanding of scientists´ profession. All activities with the robots are documented by the children themselves, processed and reflected about in complementing talks. To cater to all learning types, a high value is put on using different methods of teaching and the children work actively in workshops. Thus, a strong sense of identification with the project can be ensured in both children and lecturers. The collaterally conducted case study demonstrates the gained findings and enables multipliers to apply this concept adapted to their own needs. Complementing this case-study we recommend using this procedure in kindergartens with a high number of children with migratory background. Also, we point out the limitations of constructivist educational concepts in kindergartens.**

***Keywords*— kindergarten, robotics, programming, children as scientists, finite automata**

I. Introduction

Following the basic idea of familiarizing children with the world of science and technics, the pilot scheme „Wissensakademie" was created in cooperation with "Kinderfreunde Wien" and the Department of Computer Science at the University of Applied Sciences Technikum Wien.

The Department of Computer Science has been an active part of the RoboCup initiative in Austria for many years and hosts the "Regionalzentrum Wien" for the RoboCupJunior initiative.

The RoboCupJunior "Regionalzentrum" offers trial courses for programming Lego Mindstorms® NXT robots, hosts advanced courses, days for practice and coaching-sessions. Additionally, week-long introduction classes in robotics were held for ten to 16-year-old students. The goal of these summer classes is to motivate students to participate in the RoboCupJunior´s Austrian Opens and winning teachers as multipliers to continue the project.

A. Motivation

After building up a stable core of participants for the RoboCupJunior initiative it appears reasonable to us to extend the chance to engage in technics actively and under guidance to children as early as kindergarten. This way they can sample and experience the contact with technics. We hope that in this way, fear of contact with technical products and informatics will be minimized or, ideally, will not arise. Children could link everyday knowledge to technics and we hope to offer them first chances make and check their own assumptions. A high value is put on the method of teaching that allows children to gather knowledge themselves. The robots´ immediate reaction to their actions shows the little scientists the consequences of their acts. Thus, creativity and concentration, which are also called for in numerous other kindergarten projects, can be simultaneously improved. Over the last years, a downright environment of projects has developed in kindergartens and children seem to be used to occupying themselves with new matters continuously. "[…] when children are still in the stage of understanding their surroundings by grasping things but also develop first thoughts about logic relations, the foundations for programming should be set. It is very important to foster this as early as possible, because everything a child really understands can later be applied to similar problems." [1]

B. Advantages

We consider it a great advantage, that children in kindergarten can learn and experiment without exam pressure. This gives them the freedom to try a lot and to identify their own interests. A playful approach to exploring the world of robotics helps them to engage in the subject at ease and with no pressure. Conveying facts is not in the foreground and the acquisition of generic skills in the field of speech, counting, orientation in space, phrasing and checking of assumptions as well as breaking down complex courses of action into single steps happen almost casually and effortlessly. As in Schweikardt and Gross, we are „motivated by the idea that experimentation and play with robots exposes students to many subject areas within science, math and engineering." [2] Rapeepisarn et al. cover the aspect of „learn through play" and „entertainment" at length. [3] Even more, robotics can form the basis for an education in programming and engineering „via the back door". [4], [5] Another advantage lies in our chance to learn from the children. In 2006 Schweikardt and Gross thought that robotics was going to lead an insular existence, and would profit from „increased interdisciplinary collaboration with designers, materials scientists, psychologists, and other creative people." [2]

In addition to an excursion with the children to the college´s laboratories, the variety of learning methods provides a vast support of different learning types. The use of a digital camera that is appropriate for children offers a possibility for them to document their own activities. Drawings can visualize procedures, discussions can activate new links.

C. *Structure of the paper*

At the beginning, we introduce the robots they come to use, as well the underlying educational concepts. Afterwards we present the tutorials held at the kindergarten in the course of the pilot scheme "Wissensakademie" regarding their contents. Following a short description of our research method, we go into detail of our findings and discuss them in context of an extension of the pilot project to a number of kindergartens with a high number of children with migratory background.

II. BACKGROUND

A. *A Short History*

Froebel coined the term kindergarten and already developed strategies for hands-on learning, supported by toys and activities, in the 19th century. [6] [7] According to Kafai et al., analogies to his approach can be found in the concept of Lego Mindstorms®. [6] Based of these experiences, the use of robotics in kindergartens appeared reasonable.

„Currently, interest has shifted from whether technology should be used with young children, to how it should be used in order to provide effective learning experiences." [8] Granting a playful approach, according to Rapeepisarn et al., enhances the development of children in all aspects. [3] „The focus has shifted from technology to pedagogy." [9, p. 11] The right choice of robots that rather support the process of learning than cause frustration through complex handling is anything but trivial. Robotic Toys appear to be suitable for kindergarten children.

B. *Robotic Toys*

Robotic Toys are gaining notice increasingly in IDC-research (Interaction Design and Children), as Fernaeus et al. noted. „We define robotic toys as robots intended for basic leisure activities such as play, creativity, playful learning, entertainment, and relaxation. […] A robot is an active tangible artefact that interacts directly with the world around it." [10]

From a Development Psychology point of view, the use of robots in kindergarten is well justified. Development Psychologists, headed up by Piaget, emphasize the importance of the use of physical objects during childhood for the development of cognitive skills. [11] In doing so, children should in no case be overstrained. As one technological learning goal, Alexander and Rackley name the process of switching on and off a computer. [12]

C. *Bee-Bot*

Simple user interfaces are extremely important for children, since common user interfaces of computers are too complicated and form a barrier in the learning of the technology. [13] The fact that computers can generally be operated only by one or two people blocks the cooperation between children, because they cannot work on solving problems together. [1]

Over the last years, there have been increasingly more robotic toys developed to resemble pets in behavior and appearance. [10] Because of the positive experiences in the work with children (e.g. [14]) and the numerous availability of educational material [15], we decided to use Bee-Bots as robots for our exercises. The possible didactic applications of the Bee-Bot and its functioning can be looked up at Pekárová [14]. For better understanding, **Fehler! Verweisquelle konnte nicht gefunden werden.** shows a state diagram of the functionality of the Bee-Bot.

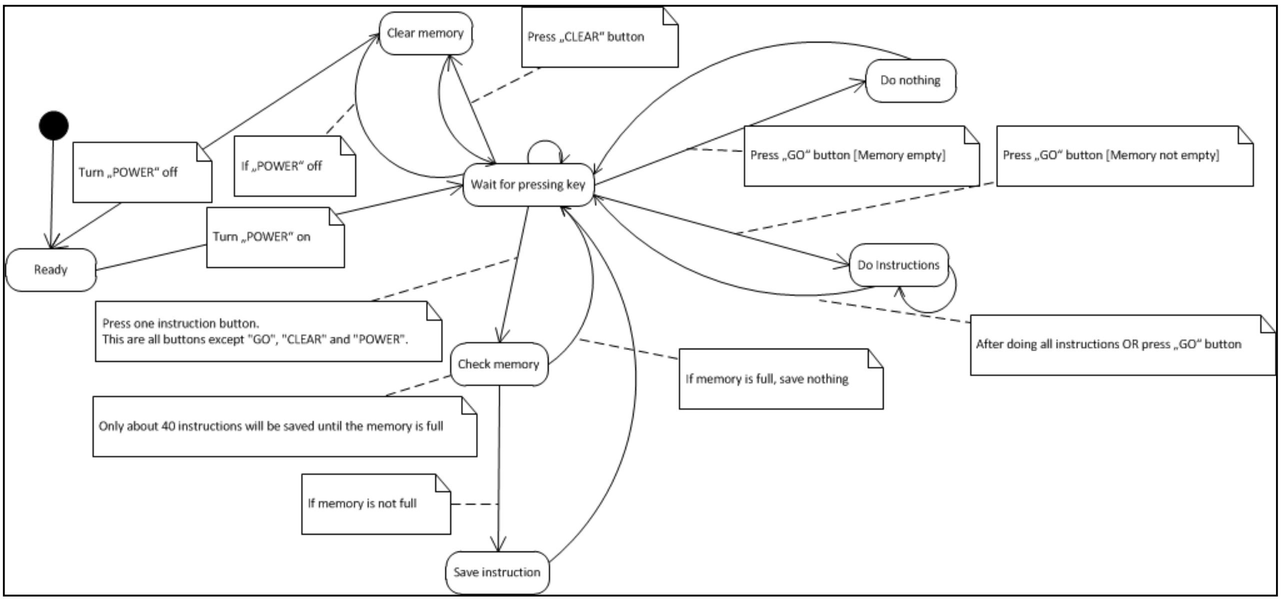


Fig. 1 State Diagram of the Bee-Bot (own modeling)

D. Constructionism

Constructionism was founded by Papert and is based on constructivism. As opposed to constructivism, constructionism only applies to learning and teaching with technologies. Furthermore, Papert is considered to be the developer of the Lego Mindstorms® robotics concept (see [16]). [9, p. 4] The concept of constructionism can be looked up at length at Umaschi Bers [9].

E. Similar Projects in Kindergarten

In 2003, Gibbs and Roberts conducted a project with 10 children aged four to five, in which they were allowed to experiment with computer games developed especially for kindergarten children on CD-ROMs. In doing so they were supported by a scientific team member. With the help of an especially developed smartboard, the children's interactions (or non-interactions) with the CD-ROMs were logged. The project´s goal was to find out how young children interact with technologies, CD-ROMs in particular, and what they learn from their experience with the technology. [8]

At Fernaeus et al., children aged four to 17 were watched handling robotic toys, in particular, Pleo, inside their families. This was primarily to study the interactions and the effects of the use of technology in familiar surroundings. [10]

Because of the complex handling of computers, Khandelwal and Mazalek worked with a „Teaching Table: A tangible mentor for pre-K math education." developed especially for the cause. [13] Scharf et al. developed Tangicons, „non-electronic physical programming cubes", that were suitable for children in kindergarten to gain first experiences in programming. [1]

III. THE PROJECT "WISSENSAKADEMIE"

A. The Project Concept

As described in the introduction, the idea for this project derived from our activities with RoboCupJunior[1]. Children are given the opportunity to get in touch with technics and science by means of fun and playing. To document their experiences, and to internalize the importance of good documentation in a scientist´s daily work, they are given digital cameras suitable for children. The pictures taken during the pilot project are to be put into a research-booklet for documenting the laboratory work and to be complemented by drawings. By becoming engaged intensely in their work, we want to enhance the children's enthusiasm for technics and prolong their questioning of the courses of actions taken.

[1] RoboCupJunior is part of RoboCup, a worldwide initiative focusing on research in the fields of Robotics, Artificial Intelligence and Computer Science. RoboCup provides standardized challenges such as Robotic soccer and the vision of beating the human soccer world champions with humanoid robots by the year 2050 [19]. Until then it is a long period of time and some of the todays' interested children will be the postdocs by then doing research in this field. Therefore the RoboCupJunior initiative was born as a discipline for RoboCup. The mission statement of RoboCupJunior is described by Sklar & Eguchi as follows: "To create a learning environment for today, and to foster understanding among humans and technology for tomorrow." [20]

B. Differences

Pekárová [14] conducted a similar project in Slovakia. She takes role play activities as a basis for her work. In four tutorials, she and her team used Bee-Bots to familiarize 26 children with robotics and programming. The class we offered was different in the following aspects:

- We tried to explain additional topics of computer science like finite automata and algorithmic thinking.
- Each child can be provided with a Bee-Bot.
- Our pilot scheme is meant for ten children instead of 26.
- Each child is provided with a digital camera for documentation.
- The pictures can be printed out by the children themselves on photo printers.
- Each child is encouraged to keep a research-booklet for the lab work and use the pictures for it.
- In our pilot scheme we scheduled an excursion to the robotic laboratories of the University of Applied Sciences Technikum Wien in order to give the children a deeper insight into the matter.
- The contents of our classes are aimed not only at robotics and programming, but at the profession of technical scientist as well.

C. Project Contents

The contents of the project in kindergarten were split into 10 units of 50 minutes each. Referring to Gelderblom and Kotzé, who state „if a child can solve a specific kind of problem in one domain that they cannot necessarily transfer that skill to a different domain" as one of their design-lessons, our pilot project tries to enable the children to internalize knowledge by continuously repeating the most important concepts in various settings. [17]

1) Interaction with children and teachers

One teacher and two to four students supported the kindergarten-class. In this setting, the teacher held the short lectures in each unit, while the students were coaches and tutors to the kindergarten children. They supported the children individually by repeating und explaining the specific content. So, it was possible to realize many learning settings. In single-learning-settings the kindergarten children had to solve individual programming tasks, to draw robot pictures, to take pictures and to keep records of their own learning activities in their research-booklets. Some exercises were authored in pair-work-settings, whereby the children assigned programming tasks for each other. To form a framework for the single and pair-activities, we carried out group activities like discussions and presentations, demonstrations, dancing, singing and playing.

2) Used Concepts of Programming

Hubwieser and Aiglstorfer [21] distinguish between basic process units and combined process units. Basic process units are indivisible process units, which are running without any conditions. There are three kinds of combined process units

- Sequences,
- Conditional process units, and
- Repetitions (loops). [21]

In our course setting we focus on the basic process units and the sequences. Basic process units which are processed consecutively can be combined to a sequence. Conditional process units and repetitions could not be part of a users` Bee-Bot program but the state diagram in figure 1 shows that these two concepts are part of the finite automat Bee-Bot. While working with the Bee-Bots children use these two principles automatically and sometimes unconsciously.

Examples for tasks where children need basic process units are in the context of the Bee-Bot: moving one step forward, one step backward, turn right or turn left. In order to move one step right or left we need a sequence of the following two instructions: turn right/left and move one step forward. Another simple sequence: two steps forward.

3) Tutorial Contents

The following table lists contents and goals of each tutorial. It aims to offer an insight into the work with children and the tasks, they should solve.

TABLE I
TUTORIAL CONTENTS

Unit	**Contents**	**Goals**
1	Initiation, introduction of the research-booklets and activities; The children are to draw a robot, explain, what they know about it and finally enact the behaviour of robots in a „play“.	Activating previous knowledge, gaining knowledge about the build-up of a robot (sensors, computer, mechanics, electronics); Expressing heard information creatively by drawing and acting; Presenting and defending one´s own conclusions.
2	Introducing the work and activities of a scientist; Giving out digital cameras to become acquainted with without guidance of instructors! Presentation of a vacuum cleaner robot and explanation of its operating mode (sensors, drive mechanism etc.); Taking pictures as scientists and presentation of conclusions.	Recapitulating of what was learned in one´s own words Getting to know the profession: Scientist; Use of digital cameras; Getting to know various areas of use for robots.
3	Handing out digital cameras for documenting the entire tutorial; Presentation of a lawn mower robot and explanation of its operating mode (sensors, drive mechanism etc.); Taking pictures as scientists and presentation of conclusions.	Recapitulating of what was learned in one´s own words; Use of digital cameras; Getting to know various areas of use for robots.
4	Visit at the University of Applied Sciences Technikum Wien – explanation of NAOs (humanoid toy robot) and viewing of other robots on the premises	Recapitulating of what was learned in one´s own words; Getting to know various areas of use for robots; Getting to know a university
5	Introduction of the Bee-Bots (without live demonstration); Presentation of their operating mode; Letting the children try out the Bee-Bots without guidance or assignment. First exercises on a CVC-Mat (Consonant Vowel Consonant Mat)	Recapitulating of what was learned in one´s own words; Getting to know various areas of use for robots; Handling of a simple robot
6	Personalization of the Bee-Bots (naming, dressing up etc.); Letting the children tell their Bee-Bots´ story; Singing a Bee-Bot song; Dancing along with the Bee-Bot song; Each child is in turn assigned a task of programming their Bee-Bots; Meanwhile, the others are training or taking pictures	Recapitulating of what was learned in one´s own words; Working with various materials and techniques; Training of (linguistic) creativity; Handling and programming of robots; Working autonomously with technic; Getting to know creative work with technic
7	Building groups with free choice of a mat and coach; Coaches assign problems to be solved by the children; Holding a competition on one mat with identical requirements for all	Recapitulating of what was learned in one´s own words; Handling and programming of robots; Working autonomously with technic; Playful interaction in a competition

8	Entrance talk: The children explain the Bee-Bot and its operating mode themselves; The children work in pre-defined pairs and assign problems to each other; Building new pairs to construct a mat for each Bee-Bot.	Recapitulating of what was learned in one´s own words; Constructing solvable problems; Autonomous solving of problems in teamwork; Creative work with materials.
9	Completion of the Bee-Bot mats; Constructing problems for one´s own mat and having them solved.	Recapitulating of what was learned in one´s own words; Creative work with materials; Constructing solvable problems and solving problems constructed by others.
10	Retrospect; Collective viewing and completion of the research-booklets; Outlook on computers and programming.	Recapitulating of what was learned in one´s own words; Presentation of one´s own work; Getting to know computers.

The ten units were held at weekly intervals. Merely between units seven and eight there was a gap of three weeks because of the Christmas holidays. At the beginning of each unit we spoke about the last unit. While each unit children gained skills of autonomous documentation of events.

D. Participants

Nine children, four girls and five boys, all in pre-school age (aged five to six years) participated in the project at hand. All children are part of the same organizational group in kindergarten and are familiar with each other. The leader of the involved group was interviewed and both she and the kindergarten administration supported us in the execution of the project as well as in the care for the children from the beginning. [18]

The "Wissensakademie" was conducted by a scientific member of the Department of Computer Science in the role of teacher, coach and presenter. In doing so she was supported by six students in their first and third semester of their study of computer science. They filled out the observation templates as well as act as personal coaches for the children. The presenter took care of the organizational progression of events and the contents and execution of each unit.

E. Empirical Study Procedure

1) Our main questions

a) Can handling robots create a wish to be a technician already in children in kindergarten?

b) Does handling robots have an influence on children´s estimation of their own skills?

c) To what extent can children in kindergarten already grasp the basic ideas of programming?

2) Methods

As scientific methods, we chose interviews with guidelines as well as observation templates. Short interviews with questionnaire A were meant to be held with all nine children before start of the ten units and to be compared afterwards. Since unfortunately some children were missing already during the first unit, unfortunately only seven children could be interviewed in the beginning.

After a 4-week-window after closure of the ten units, the participating children were again questioned with questionnaire A, immediately followed by another questionnaire B. In addition, individual interviews were held. Unfortunately only four children could be interviewed because the others were not present for various reasons. For organizational reasons, the missing interviews could not be held later on. Since it is hard to make appointments with five-year-olds, the appointments were arranged with the kindergarten administration, which could not always make sure that the children to be interviewed were present at the appointed time.

Of the other 15 children in pre-school age of this kindergarten who did not participate in the project „Wissensakademie", eleven children – nine girls and two boys – could be questioned as a control group.

In addition, as scientific method of qualitative recording, standardized observation templates were filled out during the tutorials and complemented with photographic documentation.

3) Interviews

Besides collecting demographic data, the following four questions from questionnaire A were important for this case-study:

1. What do you especially enjoy doing?
2. What are you especially good at?
3. Have you ever played games on a computer?
4. What do you want to do later? (profession)

In questionnaire B, eight questions of evaluation were asked, one half referring to the works of scientists, the other to the discussed contest.

1. What is a robot?
2. What are robots needed for?
3. Do you believe that we will soon have more robots in our everyday life? What do you think they will do?
4. What do you think you learned in the workshop „Wissensakademie"?
5. What does a scientist do? (Do you want to engineer, invent something someday?)
6. Do you want to work as a scientist?
7. What is the research-booklet for documentation of the laboratory work used for?
8. Which way of recording conclusions do you find the most practical?

IV. Results and Discussion

Question a) and b) were dealt with by the means of the interviews held. The answering of question c) is based partly on the outcome of the interviews and the first partial results of the observation templates. The other results are based mostly on the presenter´s and students´ observations.

A. Question a)

Before start of our class, none of the children in the participating group expressed the wish to one day become a technician respectively scientist. Two of the girls in the control group expressed a wish to work in technics. One even said she wanted to „make computers“.

After taking part in the „Wissensakademie” class, none of the children expressed a wish to be a technician respectively scientist as an answer to the question about their career aspirations on questionnaire A. Two of the children said they „didn´t know”. The answers given on questionnaire B partly contradicted this. One child who according to questionnaire A wanted to be a „football star“, answered to questionnaire B he wanted to be a scientist. It should be mentioned that unlike on questionnaire A, in the course of this talk, the job of a scientist was worked out. After talking about questionnaire B, children liked the idea of becoming a scientist.

We noted that four out of eleven children (36%) of the control group owned a computer. One of them was the girl who´s wish was to work in the computer sector. 54% of children in the control group wanted to be doctors. According to the head of this kindergarten, this was probably due to a health care project held shortly before. Why this development did not take place as such in our analysed target group can only be guessed. To evaluate the power to influence the children’s career aspirations it is necessary to observe their engagement during the next 10 to 15 years. At this moment, we develop a concept to realize this study. But abstract terms like technician or scientists – unless practiced by someone in their personal surroundings – seem to take time and continuous repetition to be internalized by children.

B. Question b)

The questionnaire A contains two questions, which are used to find out children´s estimation of their own skills. Furthermore, we tried to find out, if there is a correlation between the mentioned skills and talents and their favoured activities. Before taking part in our course program, most of the interviewed children answered with “playing with friends” to the question “What do you especially enjoy doing?” six out of seven interviewed children answered to the question „What are you especially good at?” - “I do not know”.. Only one girl answered with “I am good in drawing” These results suggests that on the one hand, children do not know what skills are and on the other hand they do not know to articulate their talents. It was very interesting to see, that children did not see any coherence between these two questions. The children mentioned social activities as their favourite activities. They seemed to have the opinion, that these social activities – like “playing with friends” – are not an answer in the right meaning, when talking about “What are you especially good at?”. After closure of the ten units, 100% of the participating children answered to the same question immediately. Mostly they mentioned sportive or creative activities. One child mentioned it was good at „calculating“, and said “thirty times thirty gives nine hundred”.

It appears that the participating children became aware of what they were subjectively good at and were able to articulate this. Increasing self-confidence seems to be a result of handling robots.

C. Question c)

Two of the children of the participating group stated that they owned their own computer. In the control group, four children own one. Five children of the participating group and nine of the control group play computer games.

The observation templates show that all of the participating children at once had a feel for the programming of the Bee-Bots. Also, all basic process units the Bee-Bot offers (e.g. first steps backward and forward) were no problem after a few tries. However, the changes of direction of the Bee-Bots presented a challenge. There was no problem understanding the concept of sequences – as described above – but children did not understand the usage of the turn right/left button.

A change of direction of a Bee-Bot is made of two commands at least. A step to the left is made of a turn left followed by one step forward. This led to the biggest problems in understanding the Bee-Bots´ programming. The children mostly triggered steps into a different direction manually: The Bee-Bots were turned around by hand. None of the children could handle more complex motion sequences consisting of at least three different directions.

Grasping and autonomously developing simple courses of program presents a serious problem for most children. An even bigger one was the language, especially when explaining more complex motion sequences. It appears to be possible to convey more complex sequences, but calls for a lot of time and patience in doing so.

D. Further findings

1) Self-confidence

Generally we noticed that during the interview after taking the class, the children were a lot more eager to talk and give information. This is partly due to the now familiar surroundings (coach respectively presenter). These findings correspond with those of Alexander and Rackley. [12] The active participation in class – presentation of tasks and the reflecting talks – surely help to enhance linguistic skills.

2) Paradox

The children themselves pointed out the paradox of the Bee-Bot – a robot in form of a bee that can neither fly nor sting. Quite contrary, it rolls on the floor, is a lot bigger and friendly.

3) Cameras and Printers

The use of cameras was accepted very positively by the children – this also showed in the big number of very good pictures. The children were able to handle the cameras by themselves almost without mistakes and in course of time found all the cameras´ functions. However, the childrens´ attention was often so consumed by the cameras; they were often distracted from the workshop. In the future, we will only provide cameras at specified times in order to be able to better channel their attention.

Unfortunately, the printing of the pictures failed because, unforeseen, the printers were incompatible with the cameras. The children understood the simple handling of the photo printers at once, but unfortunately the photos on the memory card were not shown on the display. Thus, navigation and selection of the wanted pictures for printing was impossible. The printing was done by the students after each tutorial.

4) Absence

It was very problematic for us, that some of the children did not show up at the workshops regularly and thus missed essential contents. The absence of the children was in the responsibility of the parents or legal guardians, since they have to see that the children appear at the kindergarten regularly and timely. It is necessary to find strategies to work against these circumstances. The kindergarten direction confirmed that the missing or being late of children is not a problem of the services the kindergarten offers. Often it is because of misunderstandings between the kindergarten direction and the parents, because they often read and understand information in form of letters only partly because of language barriers.

5) Pedagogical Considerations

The sixth unit should be closer looked at from a pedagogy point of view. First signs indicate that the boarders of the constructivist paradigm of learning are reached very soon in kindergarten – at least in the area of technics. Autonomous learning without guidance or assignments with new and unknown technologies seems to be almost impossible.

Something similar can be gleaned in Gibbs and Roberts: „It was found that though the children enjoyed themselves, they appeared to learn very little, particularly in terms of content. The most significant factors influencing this outcome were the pedagogy […] and the scaffolding support provided." [8] For older children, especially at secondary school, it is easy to work on RoboCupJunior-assignments successfully in teams under the constructivist paradigm of learning. [4]

But the example of the digital camera mentioned above shows, that familiar technologies can indeed be used autonomously and intensively.

In conclusion it should be mentioned that during our class, there were no differences between the approach to technics of girls and boys worth mentioning.

V. CONCLUSION AND FUTURE WORK

The project „Wissensakademie" has – from our point of view – proven to be very successful. The children were enthusiastic, interested in programming and robots and are looking forward to a sequel to this project.

The small group of nine children was ideal to work (inter-) actively with the children and explore the world of robots playfully. The children were easy to be enthused about going on this new exploration. Under the presented setting, a top quality promotion of the children was possible.

We do not think that the children can be cared for individually in bigger groups with the same resources of support. But the positives of our setting are at the same time limitations: we cannot give any general statements for the following reasons. First of all: Our group of children was too small – nine children are to less, for making general conclusions. Also a problem: Only sometimes we were lucky, to see all children in our robot-sessions. Too many times children were absent. Even in the small group it was not always easy to take all the language barriers, attention deficits, different speeds of learning and the varying previous knowledge into account, in order to offer the ideal amount of promotion to each child. Also, we were not always able to fully consider all cultural differences of acting and social interaction.

Animated by the kindergarten educationist, we developed learning materials for reinforcement in order not to decelerate the process of curiosity and the eagerness to experiment with technics in the children. These materials are not only meant for revising the last workshop-unit, but also to prepare for the next one. This way, a continuous gain of knowledge or a consistent occupation with the new filed of knowledge can be secured. In addition to this it is important to mention that we did not notice differences in the approaches to or handling of the robots between boys and girls.

One critical point of our study could be the circumstance that we used many different materials and methods in our classes – considering the duration of each session (50'). This fact and medium- and long-term suitability need to be subject to further research.

Besides a detailed analysis of the observation templates, the program will be continued with the same children by means of an advanced class in the summer of 2011. After a few revising units to consolidate their knowledge, ten more units with easy programming exercises with the popular Lego Mindstorms® NXT's will be held in summer. For this, the children will be allowed to build up the robots in order to establish a personal relationship to them as well as to gain their first experiences in programming with the graphic user interfaces.

Another kindergarten could be won for a repetition of the program. This time, we will use our gained experience, offer even more intensive single coaching.

ACKNOWLEDGMENT

Thanks to our students, all the children and staff taking part in this project.

REFERENCES

[1] F. Scharf, T. Winkler and M. Herczeg, Tangicons: "Algorithmic Reasoning in a Collaborative Game for Children in Kindergarten and First Class." *Proc. of the 7th International Conference on Interaction Design and Children* (Chicago, Illinois, USA, June 11-13, 2008). IDC'08. ACM, New York, NY, pp. 242--249. ISBN 978-1-59593-994-4, 2008.

[2] E. Schweikardt and M. D. Gross, "roBlocks: A Robotic Construction Kit for Mathematics and Science Education." *Proc. of the 8th International Conference on Multimodal Interfaces* (Banff, Alberta, Canada, November 2-4, 2006). ICMI'06. ACM, New York, NY, pp. 72--75. ISBN 1-59593-541-X, 2006.

[3] K. Rapeepisarn K. W. Wong, C. C. Fung and A. Depickere, "Similarities and differences between "learn through play" and "edutainment"." *Proc. of the 3rd Australasian Conference on Interactive Entertainment*, 4-6 December 2006, Perth, 2006.

[4] M. Petre and B. Price, "Using Robotics to Motivate 'Back Door' Learning." *Education and Information Technologies* 9:2. Kluwer Academic Publishers, pp. 147-158, 2004.

[5] M. Bers, C. Rogers, L. Beals, M. Portsmore, K. Staszowski, E. Cejka, A. Carberry and B. Gravel, "Innovative session: early Childhood Robotics for Learning." *Proc. of the 7th international conference on learning sciences* (ICLS '06). International Society of the Learning Sciences 1036-1042, 2006.

[6] Y. B. Kafai, K. A. Peppler, Q. Burke, M. Moore and D. Glosson, "Fröbel's Forgotten Gift: Textile Construction Kits as Pathways into Play, Design and Computation." *Proc. of the 9th International Conference on Interaction Design and Children* (Barcelona, Spain, June 9-12, 2010). IDC'10. ACM, New York, NY, pp. 214--217. ISBN 978-1-60558-951-0/10/06, 2010.

[7] H. Raffle, "Kinesthetic media: touch, toys & interactive materials." *ACM SIGGRAPH 2006 Educators program* (SIGGRAPH '06). ACM, New York, NY, Article 8, 2006.

[8] D. Gibbs and S. Roberts, "A Jump-start in Learning? Young Children's Use of CDRom Technology." *Proc. of the IFIP Working Group 3.5 Conference: Young Children and Learning Technologies.* (Parramatta, USA, July, 2003) IFIP'03, pp. 39-47, 2003..

[9] M. Umaschi Bers, *Blocks to Robots: Learning with technology in the early childhood classroom.* New York, NY: Teachers College Press, Columbia University. ISBN 978-0-8077-4847-3, 2008.

[10] Y. Fernaeus, M. Hakansson, M. Jacobsson and S. Ljungblad, "How do you Play with a Robotic Toy Animal? A long-term study of Pleo." *Proc. of the 9th International Conference on Interaction Design and Children* (Barcelona, Spain, June 9-12, 2010). IDC'10. ACM, New York, NY, pp. 39--48. ISBN 978-1-60558-951-0/10/06, 2010.

[11] J. Piaget, *Play, dreams, and imitation in childhood.* New York: Norton, 1962.

[12] T. Alexander and C. C. Rackley, "Integrating Information Assurance (IA) Into K-5 Curriculum." *Proc. of the Information Security Curriculum Development Conference* (Kennesaw, GA, USA, September 23-24, 2005). InfoSecCD'05. ACM, New York, NY, pp. 1--3, ISBN: 1-59593-261-5/05/0009, 2005.

[13] M. Khandelwal and A. Mazalek, "Teaching Table: A tangible mentor for pre-K math education." *Proc. of the 1st International Conference on Tangible and Embedded Interaction* (Baton Rouge, Louisiana, USA, February 15-17, 2007). TEI'07. ACM, New York, NY, pp. 191--194. ISBN 978-1-59593-619-6/07/02, 2007.

[14] J. Pekárová, "Using a Programmable Toy at Preschool Age: Why and How?" *Workshop Proceedings of SIMPAR 2008, Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS,* Venice(Italy) 2008 November,3-4, ISBN 978-88-95872-01-8, pp. 112-121, 2008.

[15] L. Turja, "Best practices for technology education in early childhood – background and some examples." *Project Report D2.2 Description of best pedagogical practices of the Project UPDATE*, Project no. 042941, 6th EU Framework Programme, 2008.

[16] S. Papert, *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books, 1980.

[17] H. Gelderblom and P. Kotzé, "Ten Design Lessons from the Literature on Child Development and Children's Use of Technology." *Proc. of the 8th International Conference on Interaction Design and Children* (Como, Italy, June 7-9, 2009). IDC'09. ACM, New York, NY, 52--60. ISBN 978-1-60558-395-2, 2009.

[18] M. Tomitsch, T. Grechenig, G. Kappel and T. Költringer, "Experiences from Designing a Tangible Musical Toy for Children." *Proc. of the 5th International Conference on Interaction Design and Children* (Tampere, Finland, June 7-9, 2006). IDC'06. ACM, New York, NY, 169-170, 2006.

[19] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa and H. Matsubara, "RoboCup: A Challenge Problem for AI" *AI Magazine*, 18(1):pp. 73-85, 1997.

[20] E. Sklar and A. Eguchi, "RoboCupJunior - Four Years Later." *RoboCup 2004: Robot Soccer World Cup VIII*, number 3276 in Lecture Notes in Computer Science. Springer, 2005.

[21] P. Hubwieser and G. Aiglstorfer *Fundamente der Informatik Ablaufmodellierung, Algorithmen und Datenstrukturen*. München, Wien: Oldenbourg, 2004.

Mixed-Reality Robotics – A Coherent Teaching Framework

Reinhard Gerndt[1], Jens Lüssem[2]

Ostfalia University, Wolfenbuettel, Germany

University of Applied Sciences Kiel, Kiel, Germany

[1]r.gerndt@ostfalia.de

[2]jens.luessem@fh-kiel.de

***Abstract*— Robotics evolved as a central issue in teaching for scientific and engineering disciplines. Robotics inherently encompasses a spectrum of sciences and technologies and qualification levels. However, most current teaching approaches, related to robotics, concentrate on individual aspects or small student groups. In this paper we present the mixed-reality robotics educational approach. With our mixed-reality robotics teaching system we reached a true interdisciplinary setup, addressing different qualification levels. The system allows for aspects like peer education, learning-by-teaching, problem-based learning and competition-based (self-) assessment. It consists of the mixed-reality robotics platform and a teaching concept. The overall approach has been used in teaching robotics at secondary school, undergraduate and graduate level. Student and instructor feedback is very positive.**

***Keywords*—**Robotics education, Mixed-Reality, problem-based learning, learning-by-teaching, robotic platform, European qualification framework, robotic programming tools, artificial intelligence, swarm robotics

I. Introduction

Robotic problems offer a system-level approach to teaching. The overall objective, a functional robotic device, can only be accomplished if all components, i.e. mechanics, electronics, and computational intelligence, interoperate properly. Success or failure becomes immediately obvious when the robot is set to work. Not only theoretical competencies, but also practical skills are required to reach a functional system. Robotics thus addresses different disciplines from mechanical engineering, over electrical engineering to computer science, just to mention the most prominent ones. It also addresses different qualification levels, starting with school children, spanning the undergraduate level and finally reaching the graduate level.

However, many current robotic teaching approaches fail to use the full potential of robotics in education. In our view, this is often related to the robotic platform used and the classical lecture approach taken. Robotics kits often have a too narrow focus towards a specific qualification level. At entry level many sophisticated platforms are too complicated to operate and too costly to purchase and to maintain. At advanced level many low-cost platforms lack the opportunity for serious research. Classical lectures typically are focused to a specific faculty, like mechanical engineering or computer science, to a specific student set, like undergraduate or graduate students and towards a specific subject like robotic locomotion or vision. They fail to integrate all the aspects into a teaching framework that addresses robotics as a coherent learning framework. Therefore, with the mixed-reality approach we address both aspects of robotics in education, a robotic platform and a teaching approach.

Whilst all students typically are very enthusiastic about robotics, school students typically lack the skills and their schools the financial background to do an immediate transition from the entry-level domain to the senior domain. In their robotics biography there often is a gap between graduating from school and advancing in the course of studies to a level to re-enter robotics at a late stage at university. The presented robotic platform and the educational approach intend to close the gap between both domains. The mixed-reality robotic platform allows easy access to a basic control of the robots and opens a path to control of complex behavior of cooperating real robots. The basic idea of the platform is based on the EcoBe! micro-robots as presented in [1], [2] and [3]. It was designed with demonstrating, teaching and research in the domain of advanced robot cooperation and swarm robotics in mind. Figure 1 shows a setup of the system in a five-vs.-five robotic mixed-reality soccer setup.

Fig. 1. Mixed-Reality Setup.

At entry level, the system is used as an easily operated robotic platform with a few robots. At a higher level it can be used as a robotic kit for the domain of a large number of cooperative robots and swarm robotics. In both situations the cost of the individual robot is a crucial aspect, due to the small budget when entering robotics and due to the large number of robots required in the field of cooperative and swarm robotics. Size of the individual robot also is very important due to the space required to operate the robots. Versatility of the environment and thus the spectrum of possible applications are allowed for by using a horizontally placed screen as robot arena, to display a virtual environment and virtual objects. Real objects, like the robots, just can be placed on the screen. This way, mixed setups with real and virtual objects can be realized. With easy programming of the robots at entry level in mind, a graphical programming environment was introduced in addition to the already available programming in C and JAVA.

The overall system has been used for hands-on teaching at secondary school level and at university level [4], [5]. One of the main objectives of using robots in teaching was raising awareness on the interfacing between computers and the real world. Another aspect was to provide a test bench for requirements engineering, project management, software engineering, robotics and artificial intelligence teaching.

The educational concept involves learning-by-teaching or peer instruction as well as competitive aspects, both within the same university and among different universities. Over the time, at a number of universities in the world a student community was established to work on and with the mixed-reality robotic platform. For learning by teaching, senior students pass on their knowledge to less mature students and by this also gain deeper insight into technological aspects and improve their social competencies (fig. 2). This takes place within robotic classes and spans over different faculties as well as different universities. Typical methods for this are *joint student workshops* to work on specific aspects of the system or applications. During workshops students may discuss their individual findings and research results with their fellow students and eventually compare their development results with others. They may also join forces to address more complex objectives.

Another aspect of the teaching concept is *competitions*. During competitions students are exposed to certain aspects of robotics. Often, competitions are handled by senior students. A typical entry-level competition setup is a two-vs.-two robotic soccer game. At the advanced level teams compete in currently seven-vs.-seven mixed-reality robotic soccer games within the RoboCup, robotic soccer championship. However, the set of applications is not limited to robotic soccer.

The remaining paper is organized as follows: After this introduction, we will shortly describe three other robotics platforms for teaching robotics at school and university level. In the then following section we will present the hardware and software architecture of the mixed-reality robotic platform with its graphical programming environment. Then we present our teaching approach. Following that, we present first results and finish with an outlook to future work.

Fig. 2. Peer teaching at workshop with competition.

II. Educational Robotic Platforms

There exist numerous robotic kits for education and research. Many teaching approaches are based on a specific hardware platform. Therefore, short descriptions of three robotic platforms that are used in teaching robotics or teaching with robots are presented. The platforms typically are targeted for a distinct aspect of teaching and impose specific constraints on the teaching approach.

A. Lego Mindstorms

The best-known programmable robot kit probably is the „Lego Mindstorms" construction kit. This system is available on the market since 1998 [6]. Based on the Lego construction bricks it requires building a robot prior to working with it. On the other hand it opens a vast design space for different types of robots that may be optimized for specific tasks and may be built to individual creative ideas of the users. Many students are familiar with Lego bricks since their early childhood days.

The central control device is the Robotics Command System (RCX) or the NXT for more recent kits. The NXT encompasses a microcontroller that can be connected to a number of drives and sensors. Typical sensors are touch, ultrasonic and light sensors. With the mechanical components, mostly based on the versatile Lego Technic kits, almost any robot or other automated device can be built (Fig. 3).

The RCX programming environment is based on "RCX Code", a brick-oriented graphical programming language. In RCX, the graphical blocks represent instructions and control elements. To form a program, blocks can be directly attached to each other in a graphical programming environment. It is also possible to group multiple blocks in order to form new blocks to improve readability of the program. After putting together the program on a PC, the code is then transferred to the RCX brick, where it is executed autonomously. The more recent NXT versions use national Instruments LabView-based

NXT-G for programming. In addition there is a wide variety of development kits for different levels of programming skills.

The Lego system is also used by the Roberta approach [7]. Roberta is a project to expose school children, especially girls to robotic technology in a playful manner. Within the Roberta project a set of training examples and material was developed and training for school children was carried out.

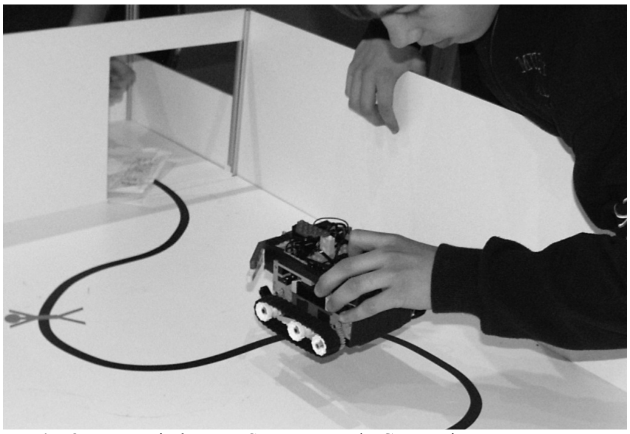


Fig. 3. Lego Mindstorms System at RoboCup Junior Rescue.

B. MA-VIN Robot Kit

The MA-Vin robot kit is based on a differential drive robot with a size of roughly 12 by 10 cm. There is almost no construction work that needs to be done and thus little flexibility in the individual design of the robots.

The robot is controlled by ATMEGA64L microcontroller [8]. Unlike Lego, the mechanical construction can be altered only marginally. In its basic configuration the MA-VIN (fig. 4) is equipped with 6 optical sensors for collision avoidance and to survey the ground. Additional sensors can be added to any of five terminals. The robotic kit comes with several I/O modules, like a touch sensor, a light sensor, a buzzer and an LED line, to mention just some.

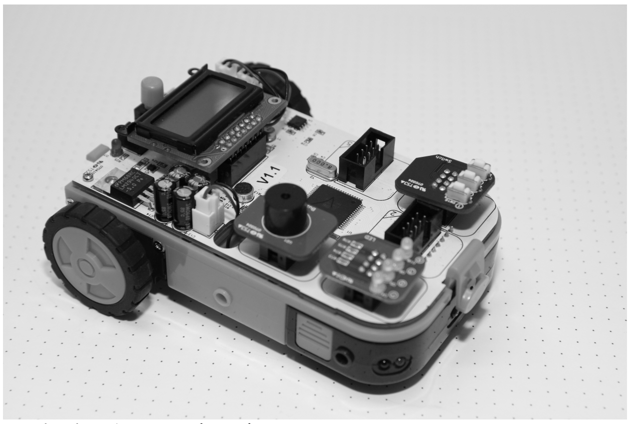


Fig. 4. MA-VIN Robot Kit.

Programming at entry level is done with a graphical programming interface. Programs are developed on a PC and then transferred to the robot, such that the robot operates autonomously. More advanced users may also use the programming language C.

Typical educational applications are a line tracer, pathfinder in a maze or robotic sumo, with the objective to push an opposing robot out of a starting circle.

C. Jasmine Open-Source Micro-Robots

When searching for a price worthy small robot, scientific swarm robotic projects offer a good entry point. There exist a number of activities related to swarm robotics. The open-source micro-robotic project of the universities of Stuttgart and Karlsruhe [9] is one of them.

The size of the Jasmine robots (fig. 5) is roughly 2.5 cm cube. Figure 3 shows the most recent version of Jasmine III with the typical on-board sensors. The locomotion principle of Jasmine, as of most other swarm robots is a differential drive.

Expandability, e.g. to add additional sensors or actuators, typically is allowed for by means of stacking PCB boards on top of the robots. Being targeted for applications with large numbers of robots, similar to the MA-VIN robots, only few different configurations are available. However, the overall robotic platform is kept open, in order to allow individual configurations by special of special add-on boards, which then requires circuit design skills.

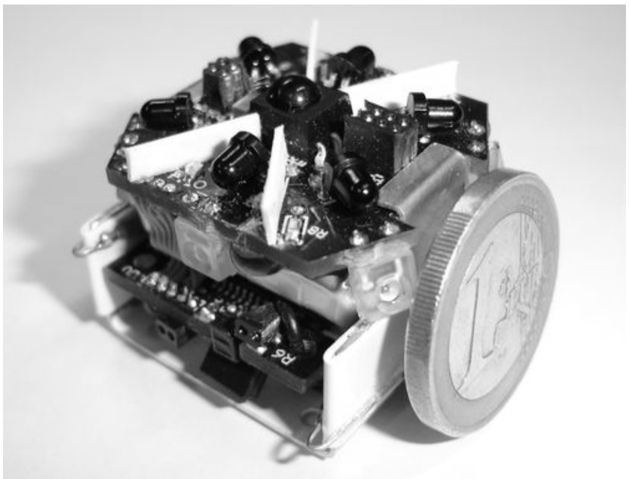


Fig. 5. Jasmine III Open-Source Micro-Robot [5].

Programming is done in C. Programs have to be loaded to the robots. The robots are typically self-contained and operate autonomously. The kit does not foresee any specific environment, as long as locomotion is feasible.

III. The Mixed-Reality Approach at a Glance

This section outlines the structure of the Mixed-Reality Robotic Kit.

A. Hardware Setup

The Mixed-Reality hardware setup consists of the micro-robots, the augmented reality display or screen and a tracking camera system. The robots size is 2.5 cm cube (fig. 6). Differential drive was chosen as locomotion principle. From a mechanical point of view, the robots consist of a body with two stepper drives and the rechargeable batteries. It may be equipped with different controller PCB.

The controller board encompasses the motor drivers, the IR communication link and the battery electronics, as well as an extension connector. They typically differ with respect to the microprocessor or set of microprocessors. A basic controller board uses a single small AVR microprocessor. In a more advanced configuration an ARM-7 processor provides computational power for independent operation of the robots.

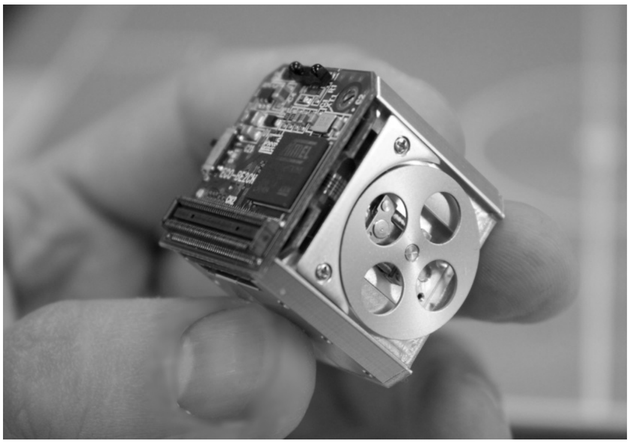


Fig. 6. ARM-7 version of EcoBe! micro-robots.

An essential part of the mixed reality environment is the horizontally mounted augmented reality display. Robots and real objects can be placed on the screen. Virtual Objects are displayed on the screen. If required, virtual objects can be virtually ‘attached’ to the robots. A tracking camera overhead captures any object placed on the screen and provides a global system view (fig. 7). For identification purpose, robots and other real object can be equipped with optical markers. Currently a variety of markers can be used. The spectrum ranges from the markers initially used, over VR markers, e.g. according to ART toolkit, to a miniature version of the RoboCup small size league markers.

B. Software Setup

The Mixed-Reality software framework consists of a number of modules for in- and output, simulation and control (fig. 8). The most central components of the system are the software agents. Each agent controls a single robot. The agents implement the intelligence of the micro-robots. They may also take control of virtual objects, e.g. kick a virtual ball, or move a virtual object, ‘attached’ to the robot. Agents may connect remotely to the framework.

Fig. 7. Camera – display setup.

The vision-tracking module captures the camera output and determines positions and poses of the robots equipped with markers and possibly other real objects. The world state generator generates an individual view for every single robot in the system. The individual ego-views are then communicated to the agents that control the robots.

The switch module separates the commands issued by the agents into commands that affect virtual objects and real robots. The robot control module takes care of interfacing and communicating with the robots. The ODE container wraps the physics engine and takes care of simulation of the virtual objects. It processes data of real objects, like position and space occupied, and commands that affect virtual objects. The physics of virtual objects can be defined freely. In the soccer system for example, different friction profiles can be implemented for the virtual ball, to account for different fields. The graphics module displays all virtual objects on the screen as defined by an XML data set.

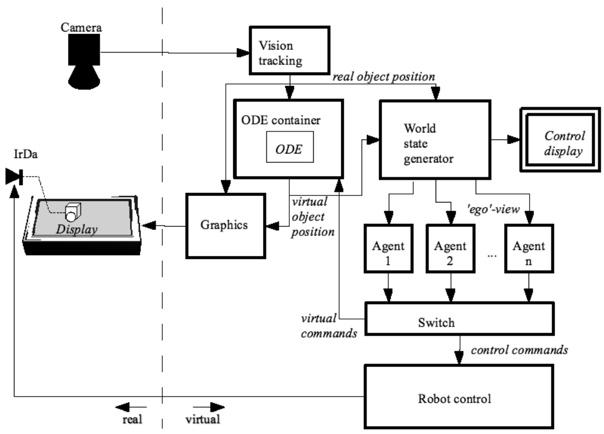


Fig. 8. Structure of the Mixed-Reality system.

The Mixed-Reality kit significantly differs from many other kits. PC-based software agents control the robots. Even though the robots do not have any on-board sensors, any kind of (virtual) sensor can be foreseen for the robots. By processing the camera image, any sensor data stream can be generated for the robot agents. It also differs with respect to the versatility of the environment. As long as it remains virtual, any kind of environment can be generated and displayed on the horizontally mounted screen. If required, interdependencies between robots and environment can be implemented in software. Real objects, placed on the screen can be identified by image processing or with the help of optical markers, attached to them.

C. Graphical Programming Environment

Easy access to programming the robots is considered as a key aspect for further activities in robotics. Therefore, aside from standard ways of programming in JAVA and other programming languages, we implemented a graphical programming environment, we named Be!Brick. The objective was to enable inexperienced students to implement cooperative and swarm robotic behavior [10]. Inspired by its success among children, we had chosen the Lego "Robotics Invention Studio" as a starting point for the development of Be!Brick.

Be!Brick therefore also uses graphical instruction and control bricks (fig. 9). A screen may contain multiple programs, which may be assigned to different robots. Communication among robots is assured by means of global variables that are accessible by all programs and event messages, similar to sensor events. The names and identification details for the available robots are read from a configuration file and thus only have to be prepared once.

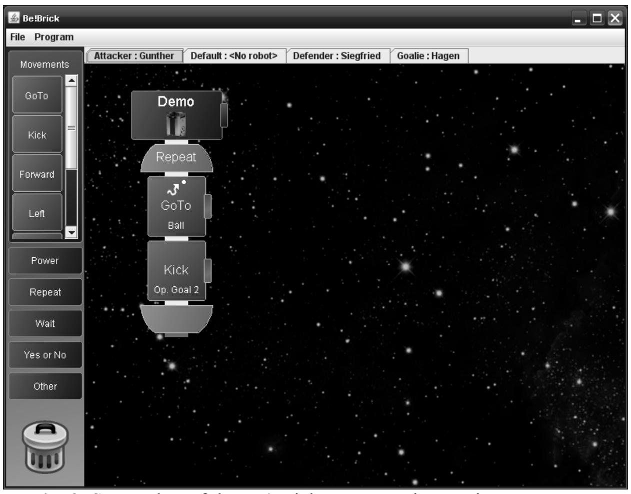


Fig. 9. Screenshot of the Be!Brick programming environment

An important issue in robot programming is localization with respect to other robots and possible landmarks in the environment. The proposed system readily provides bearings and distances with respect to the robots position and pose (fig. 10). In the soccer exercise, goal poles and corner flags are typical landmarks. Localization thus can be reduced to simple vector calculations. Similarly, target positions can be calculated, like a halfway distance between two robots or, in the robot soccer benchmark, the center of the goal line between the two goal posts as an advantageous position for the goalkeeper or a goal kick. However, if required, a more advanced scenario can be generated and robots may carry out localization based on the data from any virtual sensor that can be calculated by the server.

Robot movements can be controlled by low-level commands, controlling direction and speed of the two differential drives. A more convenient way is to use high-level commands like "drive forward" or "drive a right curve". However, the proposed system also foresees an even more abstract "go-to" instruction to directly drive to certain positions on the field. The „go-to" instruction-brick takes the target position as a parameter. The command remains active over multiple cycles and tries to minimize the distance between the target position and the robot. The execution terminates when the distance is small enough. Then the next instruction block will start.

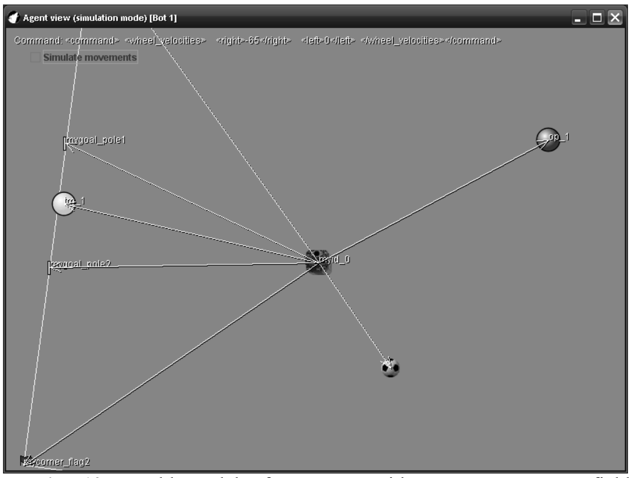


Fig. 10. World model of an agent with vectors to soccer field landmarks

Be!Brick allows for concurrent control of multiple robots. This makes it an ideal platform for exploring cooperative behavior with robots. In order to aid exploration of swarm behavior, Be!Brick contains instruction blocks that represent the basic rules used for the "Boids" system developed by Craig Reynolds in 1986. Namely these are "cohesion", "separation" and "alignment" [11]. The instruction blocks can be added to any Be!Brick program and can be configured to allow experimenting with different configurations. As the "Boids" simulation consists of multiple concurrent rules, a special brick allows merging the results of the behavioral bricks.

IV. Cross-University Teaching Approach

A. Goals of our teaching approach

The main goal of our teaching approach was to reach a higher level of qualification than typically reached. For this we set up robotics courses in a systemic way, which enables us to link courses, within different faculties, within different universities and on the secondary school, bachelor and master level [12], [13], [14] and [15].

We therefore had to combine "traditional" learning methods with problem-based learning and learning-by-teaching approaches. Using this combination of learning approaches we are able to reach the dedicated learning outcomes for the levels 6 and 7 (first and second cycle in the Framework for Qualifications of the European Higher Education Area) in the European Qualifications Framework (EQF) [16].

B. Content of our teaching approach

In terms of content we focus on:

- Robotics
  - Locomotion
  - Kinematic and dynamic motion
  - Trajectory control
  - Robot Vision
  - Localization
- Artificial Intelligence
  - Agents
  - Heuristic Search Strategies
  - Path planning
  - Machine learning
  - Swarm Intelligence
- Systems Engineering
  - Requirement Analysis
  - Software / Hardware Development
  - Software / Hardware Testing
  - Refactoring
  - Software / Hardware Maintenance
- Project Management
  - Project Planning
  - Project Management
  - Project Controlling

Further on, aspects like event management, human resources management, logistics and a few others are also touched.

C. Learning features of our teaching approach

Core features of our approach (fig. 11) are:

- Integration of students on different levels from secondary school over Bachelor on to Master level
- Development of competences in accordance with the European Qualification Framework
- Cross-qualification learning communities
- Cross-university learning communities
- Integration of short feedback cycles to foster learning success
- Integration of competitions
- Integration of lecturers in the learning process (also within the cross-university-approach)

Nevertheless, a large part of our teaching activities is done in a rather "traditional" way – we offer lectures on specific topics with additional exercises. The robotics teaching approach then helps to correct misconceptions. During early programming lectures students typically develop a false understanding of reliability of programs. They seem to believe that no difference between intended and real behavior is possible. Working with real robots results in questioning this believe. Experience from working with real robots leads to developing a concept of reliability. With respect to sensor data students (re-) develop a basic understanding of filtering, for example by means of averaging. When Boolean variables, e.g. on presence of a certain state that is derived from sensor data is concerned, implementation of counters to count presence respective absence of a specific perception during subsequent cycles is a typical approach. These concepts not only apply to robotics, but also to communication protocols over unsecure channels and real-world data bases, just to mention a few.

The robotics approach also helps to introduce some concepts in a natural way. A multitude of independent robot entities also physically demonstrates the concept of parallelism and the need for synchronization, which is an issue in larger and many real-world programs, to which students will only be exposed later in their university career. Distributed computing systems also require an understanding of parallelism and synchronization. A hands-on challenge for the students in undergraduate AI and control theory classes is implementing a two-versus-two robot soccer game with the mixed-reality system. Having two robots that need to co-operate as a team, requires parallel operation and specialization at the same time. Both robots need to operate simultaneously but also need to specialize, e.g. as a goalie and a field player at the same time. We established regular block courses to enable learning-by-teaching among students of the participating universities, based on the robotics approach.

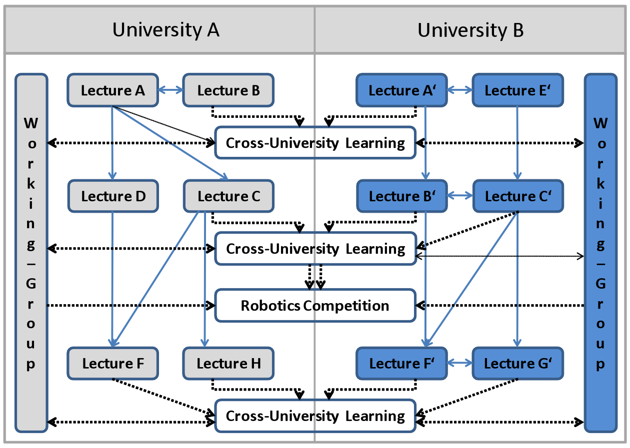


Fig. 11. Components of our learning approach.

The robotic soccer game also introduces a competitive element among individual students. In case of competitions, the evaluation of the learning effort and the solution to the specific problem is left to the experiment of a game. It is independent of subjective assessment and has a much higher acceptance by students. Competitions also play an important role in our cross-university-approach. As part of the regular block courses, cross-qualification teams are competing with each other playing robot soccer games. Complexity starts with two-robot soccer teams and reaches up to eleven-robot soccer teams.

Additionally, we established cross-university working groups in which students on different levels work together in a – nearly – self-organized way. The cross-university working groups are based working groups at each participating university. Working groups often meet during joint lectures and organize own workshops. The workshops with student groups from several universities are used to advance the overall system and if special competencies are needed that may be available only by one university or individual participant. Roles of participants typically evolve by means of experience or commitment. More mature students typically take the role of advisors to the others. Overall objectives are typically defined by the goal to improve the robotic system at hand and to apply concepts presented in lectures. After reaching consensus on the next partial goals, students decide on the general approach, on the work plan and the team structure. Faculty staff provides assistance and guidance only if necessary. Cross-university learning based on the block lectures and on the (self-) organized working groups can be seen as the backbone of our approach.

The two-versus-two game also is used for workshops with students from secondary school as part of the recruitment activities of the participating universities or students at early undergraduate level. Typically these students are able to implement a basic two-versus-two robotic soccer exercise within a few hours. The most common approach is a specialization of the two robots in each team, with a goalie and a field robot. Initially the typical behavior paradigm is reactivity. Other groups, typically with a higher degree of experience and confidence in their skills implemented dynamically changing goalkeepers. Implementing basic cooperation in a team of five robots typically takes them a little longer. Students, even with little programming experience, successfully use JAVA, C or C++ as a programming language for the robot agents. Students with even more advanced experience often start working on basic principles of swarm behavior, typically with up to 10 robots, and define individual robotic projects to be implemented with the Mixed-Reality system.

D. Evaluation of our teaching approach

Evaluations [4], [5] show that students are very motivated by these learning-by-teaching approaches. Over all, more than 300 students participated in the activities over a period of over 5 years.

In the center of our attention were the students' expectations from a cross-university workshop. Interestingly, students focus most on information exchange. Realization of new ideas and having fun when meeting with fellow students appears to be less important (fig. 12). Students are interested in having a possibly different approach to a subject, when presented by a lecturer or mature students from a different university and are eager to discuss their own ideas actively with their fellow students.

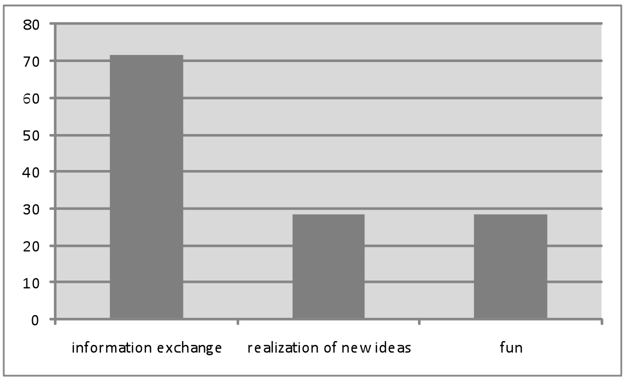


Fig. 12. Expectations from a cross-university workshop

We were also interested on what conditions a sustainable students network can be established (fig. 13). Besides the argument that the number of workshop participants should be reasonably small, students ask for regular cross-university projects and specific projects to tackle on. A clear framework, as set out by specific subjects and supported by social media, as well as financial aid or awarding credits for participation appears to play a less important role.

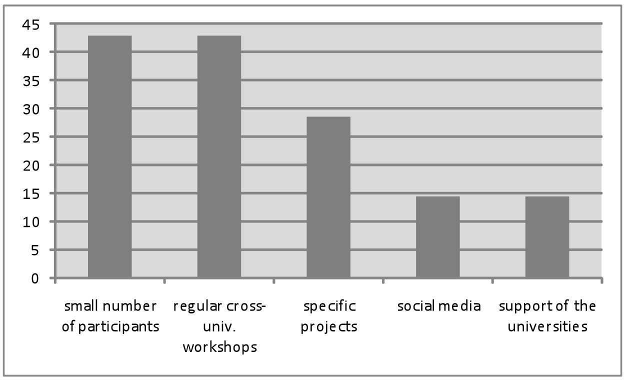


Fig. 13. Conditions for a sustainable students network

V. Conclusions

In this paper we presented an educational system, suitable for teaching and experiencing cooperative and swarm robotics. It consists of a robotic platform and a teaching framework. It combines elements from many fields of computer science, electrical engineering and mechanical engineering. A graphical programming environment provides access to novice users. Advanced users may use high-level programming languages to implement the robotic behavior by means of agents. The system thus is well suitable for students with very little programming skills as well as for serious research. The teaching approach enables cross-university and cross-qualification learning. Students invested significant resources to address subjects independently of their course of studies. Participating students evaluated it very positively.

An objective for the further improvement of our teaching approach is the better use of distance learning methods and technologies.

REFERENCES

[1] Guerra, R. da S., Boedecker, J., Yanagimachi, S., Asada, M.: Introducing a New Minirobotics Platform for Research and Edutainment. Symposium on Autonomous Minirobots for Research and Edutainment, Buenos Aires (2007)

[2] Guerra, R. da S., Boedecker, J., Mayer, N. M., Yanagimachi, S., Hirosawa, Y., Yoshikawa, K., Namekawa, K., Asada, M.: Introducing Physical Visualization Sub-League. Lecture Notes in Computer Science (2008)

[3] Gerndt, R. Bohnen, M., Guerra, R. da S., Asada, M.: The RoboCup Mixed Reality League, in Dubois E., Gray P., Nigay L. (eds.) The Engineering of Mixed Reality Systems 2010. pp. 399 – 418 Springer, (2010)

[4] Gerndt, R., Lüssem, J.: RoboCup – Wettbewerb als ein Ausbildungskonzept?, Norddeutsche Konferenz für Informatik an Fachhochschulen, Wolfenbüttel 2010

[5] Gerndt, R., Lüssem, J: Robotik in der standortübergreifenden Informatik-Ausbildung, Norddeutsche Konferenz für Informatik an Fachhochschulen, Elmshorn 2011

[6] The Lego Mindstorms webpage, mindstorms.lego.com

[7] The Roberta webpage, www.roberta-home.de

[8] The MA-VIN robot kit, www.robonova.de

[9] The Open-Sorce Micro-Robot project, www.swarmrobot.org

[10] Krupop, S.: Be!Brick Graphical Programming System for Multi-agent Systems, Ostfalia University, Computer Science Department, Bachelor Thesis 2009

[11] Reynolds, C., Flocks, herds and schools: A distributed behavioral model., *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (Association for Computing Machinery), pp. 25–34, (1987)

[12] Aebli, H.: Zwölf Grundformen des Lehrens: Eine Allgemeine Didaktik auf psychologischer Grundlage. Medien und Inhalte didaktischer Kommunikation, der Lernzyklus. Stuttgart: Klett-Cotta. 2006

[13] Lüssem, J. et al.: Combining Learning Paradigms to Ensure Successful Learning Outcomes in the Area of Software Development, EDULEARN 2011, Barcelona 2011

[14] Bielaczyc, K., Collins, A., “Learning communities in classrooms: A reconceptualization of educational practice”. In C. M. Reigeluth (Ed.), Instructional design theories and models. Volume II: A new paradigm of instructional theory. Mahwah, NJ: Erlbaum. 1999

[15] Winkler, K., Mandl, H., “Learning Communities”. In P. Pawlowsky & R. Reinhardt (Ed.), Wissensmanagement in der Praxis. Neuwied: Luchterhand. 2002

[16] N.N.: Recommendation oft he European Parliament and oft he Council of 23 April 2008 on the establishment of the European Qualifications Framework for lifelong learning. Official Journal of the European Union, Bruxelles 2008

A Contribution to the Discussion on Informatics and Robotics in Secondary Schools

Alberto Barbero[#1], Barbara Demo[*2], Francesco Vaschetto[#3]

[#] *Istituto Tecnico Superiore "G. Vallauri"*
via San Michele 68, Fossano (CN) – Italy
[1]barbero@Vallauri.edu
[3]vaschetto@Vallauri.edu
[*] *Dip. Informatica, University of Torino*
c.so Svizzera 185 - 10149 Torino – Italy
[2]barbara@di.unito.it

Abstract—**The approach to Informatics in Italian secondary schools is being reshaped after the Reform effective from autumn 2010. Educational robotics can contribute to this reshaping because it allows students acquire a technological competence in Informatics as recommended by the Computer Science teachers and researchers in universities with the "Manifesto for Informatics in secondary school" issued in May 2010. During the school year 2010-2011, first year students of a technical institute, i.e. students about fourteen years old, have developed programming activities using Scratch and S4A, Scratch for Arduino. This experience is proposed as a reference activity for a new Informatics curriculum in technical schools and, integrated with other components, to all types of secondary schools.**

Keywords—**Educational robotics, Pragmatic robotics, Manifesto for Informatics, Scratch, Arduino, S4A.**

I. Introduction

Several robotics activities are carried out in k-12 education. For primary and middle school students, reports generally describe typical educational robot activities, [1], [2], while also more specialised and pragmatic robot activities are offered to older students. Indeed, the aim of vocational and partly of engineering (or technical) secondary schools is directing students toward their working life thus they shall offer pragmatic robotics experiences supported by robotics industries also. In Torino (Piedmont, Italy) we have headquarters of COMAU, the Italian industry among world leaders in industrial robotics, and in the area several other companies are active in the same field. A recent market research by Piedmont Region and other local administrations points out that these industries will offer a relevant number of specialized technical jobs in the next years [3]. According to this market research these future jobs will concern industrial robotics but many of them, even most of them, will be in the so called service robotics where we have domotics and medical robotics, to name two of the best known areas. These areas are still in a creative and innovative phase. Hence robotics in secondary schools naturally has and will continue to have a double-face presence: the pragmatic robotics, introducing to the robotics industries above, and the educational robotics such as the one cultivated in projects like Terecop and Roberta [4],[1].

Multidisciplinarity, problem solving, programming and Informatics in general are peculiar components of educational robotics activities. Indeed, in primary, middle and non vocational secondary schools, robotics experiences described in the literature are so strictly interwoven with Informatics that we can see Informatics and robotics as joined together. Thus, for an analysis of robotics in schools we suggest considering the "Manifesto for Informatics in secondary schools", published in May 2010 by the main Computer Science academic associations in Italy [5]. The Manifesto points out that in today society people have quite different perceptions of what Computer science is. One is the pragmatic perception of people that consider Informatics to be a set of hardware and software systems, the second is the technical perception shared by people that conceive Informatics as a set of technologies to be used to implement system and applicative packages. Third is the cultural perception conceiving Informatics as the scientific discipline founding computer technology.

The Manifesto analysis fits robotics at least for the pragmatic and the technological perceptions. Indeed the cultural component in the case of robotics likely concerns the several disciplines converging into robotics and deserves a different discussion. But when we mostly consider computer components of robotics we must take into account also the third facte concerning the scientific aspects of Informatics necessary to all students because shall allow them to be conscious or at least to have a general idea of the possibilities (and likely of the limits to the possibilities) of the automated tools they will deal with or will design in their future life. This is particularly true for service robotics that appears to be the best promising field as for jobs offer. Obviously an education considering all three perspectives will provide different levels of competence in each component depending on the school type as it is for other disciplines.

In Session 2, we sketch the Manifesto and its claim for extending students vision to perceive all different conceptions of Informatics in order to acquire a computational thinking mindset [6].

In Italy educators have the chance to define curricula inflected according to the above remarks also because of the secondary school Reform effective from autumn 2010. This Reform has reshaped all secondary school types. There is a larger presence of computer science related subjects: for example Informatics appears both in technical institutes and in Applied Sciences lyceum type since the first year. Previously they were explicitly present from the third year only. The Reform does not rigidly define the approach to Informatics and, consequently, schools must give it a new shape.

Indeed, for all subjects, the Reform only defines "guidelines" for each school year in every school type leaving to teachers and to single schools defining their education paths with an autonomy and responsibility level higher than with previous school curricula, precisely defined. This autonomy is both an opportunity and a possible critical point for Computer science related disciplines since they do not have a tradition concerning contents and teaching methodologies teachers may refer to or feel sort of mandatory referring to. Besides, due to reasons not to be discussed here, in Italy, teachers of Informatics related disciplines may have many specializations different from Informatics (and these teachers are not going to soon retire). Aware of the chance of proposing new approaches to computer science, several groups of Informatics teachers in schools and researchers from universities are working together in order to design (part of) curricula implementing the Manifesto.

In Session 3 an example is sketched of the activities proposed by one of these groups. The activities have been carried out with first year students during the 2010-2011 school-year at the Technical Institute Vallauri in Fossano (Cuneo), Piedmont, Italy, where two of the authors are teaching. The students are introduced to programming using the Scratch language and its extension S4A (Scratch for Arduino) when the open hardware Arduino is to be used. In these activities multidisciplinary aspects are present that allow students to experience concepts introduced in different subjects of their normal curricula. Teachers also considered that at the end of the first two years in a polytechnic secondary school Italian students must choose among Informatics, Electronics, Mechanics and others, the specialization characterizing their final three years. The curriculum of the beginning two years is thus crucial because, obviously, the choice for the last three years is mostly based on first experiences. Scratch and Arduino activities combine Informatics and robotics allowing students to come into contact with several of the disciplines offered to them as a specialization from the third year. Besides, also for students that shall quit the secondary school at the end of the compulsory beginning two years, it is important they are introduced to software and hardware technologies they will be confronted with in their future life.

The Informatics curriculum sketched in this paper can be proposed also to students in different types of secondary schools integrated with other disciplines for example with philosophy, logics and science history in the lyceum type of school. The conclusion is that we must consider proposing a blend, appropriate to the school-type, of pragmatism with knowledge of the technological components of a robot system and of their scientific foundations.

II. THE MANIFESTO FOR INFORMATICS IN SECONDARY SCHOOLS

In Italy there are three main national associations of computer scientists: the National Consortium for Informatics Inter-universities (CINI), the Group of Informatics Engineers (GII) and the Group of researchers in Information Science from Italian universities (GRIN). While the Education Ministry was developing the Secondary School Reform, these associations contributed to the discussions in different ways. A common contribution is the "Manifesto for Informatics in secondary school" [5].

The Manifesto points out that in Italian schools Informatics is almost only present as learning technologies to implement some software in technical schools of some specialised type or, most generally, as practicing specialised software such as Office and Open Office, GeoGebra or other such softwares certainly of big help in learning Mathematics or other subjects. In the Manifesto we read that:

"Informatics is becoming the kernel of our modern world both because it is needed to the normal development of our everyday duties and because its development shapes and directs the advancement of our whole society.

Nowadays, in all areas of human activities we can find the influences of digital discoveries and achievements. Indeed, the computer is no more used for the traditional scientific calculus only, yet it is used in all areas of industrial production, medicine, publishing and communication to name only some of its applications. Two billions of people have at least one contact on the net each day. We have around us products full with hundreds of millions of billions (no typos here) of transistors– elementary hardware components supporting information technology – in our cars, in domestic appliances, inside the gas pumps, in our videogames, and they are half of the financial value of the products. Hundreds of billions of software instructions, expressions of human intelligences, give life to these components and, through them, to all processes peculiar to our modern society.

In our everyday language the expression Computer Science or Informatics refers to three different perceptions related but quite different. A person can have a

1. Pragmatic perception of Informatics and see it as a set of hardware and software systems.

2. Technological perception and conceive Informatics as a set of technological tools to be used to implement system and applicative packages.

3. Cultural perception and see Informatics as a scientific discipline founding, thus making possible, computer science technology.

The common man has the first perception of Informatics and sees it as a set of applications. Thus in his point of view knowing computer science means knowing how to use software packages and what digital devices it is reasonable to buy. On the contrary, for technicians, knowing Informatics means to know how to develop software systems."

What must be like the Informatics discipline in schools is a largely debated question. Likely there are different answers depending on the point of view one looks at computer science and the above three different interpretations inspire different ways of dealing with Informatics in schools. Different types of secondary schools can plausibly have different aims in making their students competent in Informatics and consequently they may decide to stress one of the above perspectives most. For a cultural perspective of Informatics we should mainly address its epistemological aspects and focus on its connections with Mathematics, in particular Logics, with Philosophy and History: this is particularly suggested to Italian classical, scientific and pedagogical lyceum type of schools. But the Manifesto concludes that, unfortunately, Informatics as a science is missing in almost all of our schools while it should be an aspect of digital literacy possessed in some form by everyone at the end of any type of secondary school.

III. Programming with a cat and a king as companions

Due to the very different levels of students familiarity with computers and computer science, in Italian technical schools the guidelines for the first year include the European Computer Driving License (ECDL) syllabus [7]. It turns out that most of the schools find appropriate dedicating to the ECDL certification the entire first year in order to ensure a basic common level of digital competencies. But, as we said, in the reformed school teachers have the responsibility of defining a curriculum from their class guidelines. At the Vallauri Technical Institute in Fossano, Italy, in school-year 2010-2011, teachers decided to advance to the first year part of the competencies of the guidelines for the third year in order to introduce students to algorithms and to programming from the beginning. They aimed at gaining the interest of students entering the secondary school by proposing them computer science motivating activities and at the same time allowing students to acquire abilities required by the first year Reform guidelines. Also, teachers considered important that students could have a concrete grasp on what it is going on from the formal specification of a program to its effects on something concrete, with similar motivations as those introduced in [8]. For this they chose to also propose some experiences with the Arduino board.

A. Introduction to programming

Teachers decided to look for a programming language different from Visual Basic, C, C++ or Java, scheduled to be used in last years of secondary schools and in universities, judged too difficult and requiring too much time to obtain motivating results. Scratch is a visual programming language due to Mitchell Resnick and his Lifelong Kindergarten Group of the M.I.T. MediaLab in Boston [9]. Scratch and its open source integrated development environment (IDE) suit the curriculum teachers were thinking of because it was specifically created for introducing basic concepts of problem solving and programming to very young or inexperienced students. In Scratch, programmers can easily implement animations, simultaneously execute different processes and make them interact, use events. The IDE provides a visual block editor where a program is specified by dragging the chosen icons or blocks, each corresponding to a primitive written on the block, having a different colour and shape depending on its function. The blocks have jigsaw puzzle shapes in order to limit language components that can go together. The choice of the Scratch language and its IDE is motivated and detailed in [10].

For sake of space here we describe only the activities around the conversion of a decimal number into a binary notation because it can be used for showing both aspects of the programming and robotics activities considered here. Students worked on different number notations during Informatics and Mathematics hours discussing about numbers, digits, fractions and algorithms to convert a number from one notation to another. After students worked out the mathematical aspects of the problem, each group was asked to specify a conversion algorithm from decimal to binary. The Scratch stage of one of the programming activities is shown in figure 1. The stage is the "window" where sprites are shown. In figure 1 we have the following sprites: the cat Garfield, sprites corresponding each to a bit and the Arduino sprite, not used in the first steps of the conversion activity.

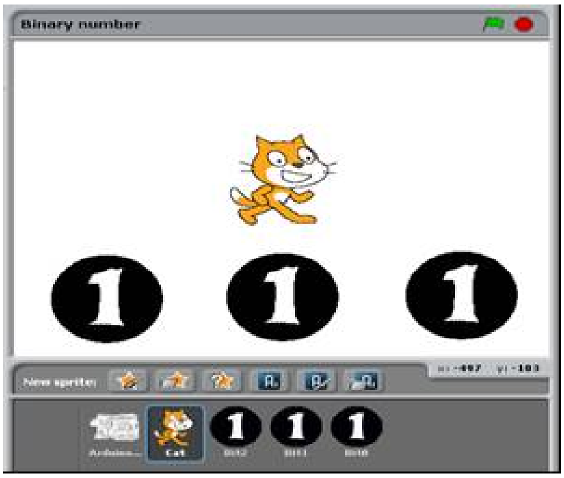


Fig. 1 The Scratch stage for converting from decimal into binary

Indeed each group of students worked on its own solution of the given problem. After each group had found an algorithm and implemented it, students were introduced to comparing different solutions of the same problem and to understand that a solution of a problem by means of an algorithm and a program has properties such as complexity and generality. Thus, though this experience is quite elementary, students could discover at an introductory level properties like complexity and generality by comparing their algorithms.

The Scratch experience has been more than a programming experience: students went from operative system features such as components and standard configurations to writing reports or presentations of their different solutions. Teachers verified that through the programming activities students also achieved most of the competences required by the first grade guidelines and were enabled to pass ECDL-like tests that, as we wrote above, in technical secondary schools are often offered at the end of the first or of the second year.

B. S4A: Scratch for Arduino

The binary notation for numbers also has different Arduino versions. Arduino is a well known open board easy to enrich with sensors and actuators suitable to the interactive environments where we want to use it [11]. The interest on Arduino is because, as we said, we consider important that students experience concrete programming, that they can see the immediate result of their code and verify what it is going on from the formal specification of a program to its effects on something "touchable", in this case Arduino. For Piedmont people Arduino board has a special meaning because it was conceived in Ivrea (Piedmont, Italy), the legend says at the Arduino bar of the Arduino main street. We had an Arduino (955–1015) Margrave of Ivrea and King of Italy. In Ivrea the Olivetti Company was based that is the company many elderly Italians owe for different, not only technological, reasons.

Fig. 2 The S4A code sequence for switching off & on "bits" on Arduino

We consider programming Arduino using the S4A (Scratch for Arduino) environment developed by Citilab in Barcellona (Spain) [12]. S4A is an open Scratch extension supporting simple programming of Arduino. It extends the Scratch IDE with new blocks for programming the sprite Arduino having sensors and actuators connected to the real board. A firmware was developed that, by means of the Picoboard protocol, allows S4A to send commands to the actuators, to get sensors status and to make run the (up to four) engines. S4A and the described firmware are downloadable at http://seaside.citilab.eu/scratch/downloads.

The activity went on through several steps summarized as follows:

a. making a led blink,
b. managing a red-light changing colours either by fixed intervals of time or by touching the sprite corresponding to a color,
c. using activities a and b to show the binary conversion of a decimal number in the range [0,7].

Figure 2 shows the S4A code for switching on and off the three leds connected to Arduino ports number 10, 11 and 13. The resulting binary notation for a decimal number ranging between 0 and 7 is shown on the same stage of Figure 1 and on Arduino using the leds corresponding each to one digit in the binary notation, see figure 3.

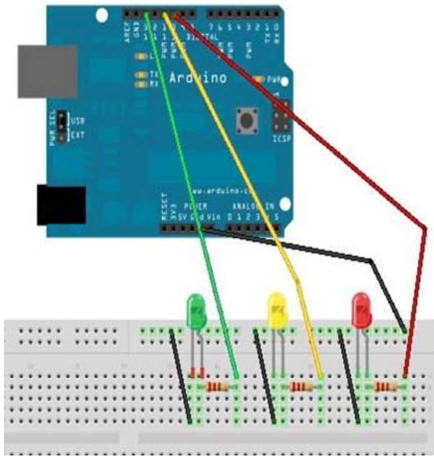


Fig. 3 Arduino and leds used for the red-light activity and for converting into binary

IV. Conclusions

In approaching robotics activities all types of secondary school shall propose a blend, depending on the school-type, of pragmatic experiences and acquisition of knowledge about the technological components of robot, of their programming and of their scientific foundations. With the enlarged presence of Informatics in the reformed secondary school some schools plan to maintain a mostly pragmatic approach to Informatics and even enrich their offer in teaching how to use packages and automatic systems of several kinds. The pragmatic approach is important to acquire basic competence in dealing with a computer, it is mandatory when we use fantastic applications to experience mathematics, physics, chemistry and the other disciplines. It is necessary for vocational schools where training courses have a relevant presence. In all cases we must consider that the practice is as important as the methodology to acquire practice.

During school professional training on technological tools such as robots we must consider the fast evolution of the technology. The risk is making room for ideas like “abstraction shall not be taught to students” heard in a recent workshop on vocational and technical schools. All types of secondary school adopting this motto are condemning its students to find difficult changing from one system to another, from one robot to another, in general when they have to update their knowledge by using their competences. Also in vocational school the three approaches of the Manifesto must be present.

Here we presented a preliminary contribution for a first year curriculum of technical schools implemented during this 2010/2011, first year of the secondary school Reform in Italy. By beginning new experiences in technical schools, where computer science is not a new presence, we try to define original educational objectives and activities for sort of testing them in environments already introduced to the discipline. The aim is to introduce basic technological concepts of Informatics also in all other types of secondary schools where it has not been present till now as, for example, in so called Lyceum, with cultural integrations, as for logics and philosophy.

With our computer science activities in secondary schools we do not aim at educating all students to become good programmers rather to introduce them to the computational way of thinking that many researchers consider essential for our next generations can take advantage of the computational power they will have at disposal.

Jeannette Wing, the President's Professor of Computer Science and head of the Computer Science Department at Carnegie Mellon, during her presentation in the Computer Science Distinguished Lecture Series at Carnegie Mellon in Qatar, said: "Computational thinking is a fundamental skill used by everyone in the world, and should be incorporated into educational programs along with reading, writing and arithmetic to grow every child's analytical ability” [6].

The example here described is a possible component of a curriculum for the first year of secondary school where Scratch is used for the programming and graphics part of activities and Arduino is used for the control part. We designed it guided by the Manifesto for Informatics and knowing that the relationships between robotics and Informatics particularly in schools allows us to avoid reducing robotics in education to a mechanistic exercise.

Acknowledgment

Many thanks to all our colleagues of the Association DSchola. Thanks also to the researchers of the TERECOP project, particularly to Michele Moro of the University of Padova. Many discussions have been fostered by the unvaluable context of the Meeting “Education and Robots” organized by the Museo Civico of Rovereto (Trento) under the responsibility of Franco Finotti and Nello Fava. Stefano Monfalcon’s students activities shown during that meeting are best and inspiring examples of educational robotics: the Depero’s puppet moved by two Lego NXT robots was surprising.

References

[1] (2004) The Roberta project website. Available: http://www.roberta-home.de/en
[2] M. S. De Michele, B. Demo and S. Siega, “A Piedmont SchoolNet for a K-12 Mini-Robots Programming Project: Experiences in Primary Schools”, in *Proc. Terecop Workshop, First SIMPAR Conference*, Nov. 2008.
[3] (2011) The Piedmont Region website. “Competenze e professioni nel futuro del Piemonte”. Torino, June, 9th 2011 Available: http://www.regione.piemonte.it/europa/notizie/comp_prof.htm
[4] (2009) The TERECOP project website. Available: http://www.terecop.eu/
[5] (2010) GRIN website, The Manifesto for Informatics in secondary school”, http://www.grin-informatica.it:8080/opencms/opencms/grin/infoescuola/riformascuola/
[6] Wing, J., Computational thinking is a fundamental skill, Presentation in the Computer Science Distinguished Lecture Series at Carnegie Mellon in Qatar, April 2011, http://www.cmu.edu/news/blog/2011/Spring/computational-thinking.shtml
[7] ECDL website, Available: http://www.ecdl.com/
[8] B. Demo, G. Marcianò and S. Siega, “Concrete Programming using Small Robots in Primary Schools” in *Proc. ICALT 2007*, p. 124
[9] Resnick, M. and all., Scratch: Programming for All, CACM, Vol. 52, No. 11, pp 60-67, November 2009
[10] A. Barbero and B. Demo, “The Art of Programming in a Technical Institute after the Italian Secondary School Reform,” in *Proc. ISSEP 5th, 2011*, (to appear).
[11] Arduino Open hardware Project, www.arduino.cc
[12] S4A - Scracth for Arduino, http://seaside.citilab.eu/scratch/arduino

Cross-Curricular Approach to Robotics in Interactive Museum-Pedagogy Environment

Nikolaos Fachantidis[#1], Vassiliki Spathopoulou[*2]

[#]*Educational Dept., University of Western Macedonia*
Florina, Greece
[1]nfaxanti@uowm.gr

[*]*12st Ephorate of Byzantine Antiquitites*
Kavala, Greece
[2]vassiliki.spath@gmail.com

***Abstract*— In the scientific field of Museum Pedagogy, a construction and its interpretation, constitute the signifier and the signified for the educational procedure. It is a common sense that the presence of new technologies, by tangible or virtual means, should contribute in a useful and functional manner in order to preserve the museological structure respecting educational aims without underestimating the value of the museum exhibits. Sometimes it is common knowledge that the technological and the interactive museum exhibits could distract the visitor from cancelling the possible experiential learning.**

The implementation of the project "Ancient Greek theater" showed that the exploitation of robots in museum activities can contribute to the achievement of museum-pedagogical aims. The cross-curricular design is an important factor for the balance between social-humanities and technological studies. Also the intuitive control and the playful character of the interactive environment create and support engaged and experiential learning, which lead to better comprehension.

***Keywords*— Cross-curriculum, Educational Robotics, Interactive Robots, Learning Environment, Museum-Pedagogy, Edutainment.**

I. Introduction

In the scientific field of Museum Pedagogy, a construction and its interpretation, constitute the signifier and the signified for the educational procedure. The interpretation of the museum exhibits is a complex procedure and the educational and museum pedagogical structures compose the appropriate means to bridge the gap between the museum exhibit and the museum visitor.

The polysemy of the museum exhibits is formatted by the interconnections of the external shape with the cultural symbols, ideals and the values which the exhibits represent [1]. When the prior knowledge, the empirical, cognitive and the aesthetic background of the museum visitors are taken into consideration in combination with the social changes as they are reflected in the museum exhibits [4], [6] could make the interpretation complete, unprecedented and unique.

II. Museum-Pedagogy

A. *Learning and Interaction*

The museum constitutes a learning, communication and entertainment environment. In the frame of the educational structure planning in a museum-pedagogical program, the concept of the experiential learning is disposed as a possible strategy bridging the gap between the transmitter and the receiver and creating accessible the complete interpretation of the museum exhibits. The museum visitor learns by doing, raising his interest for a creative expression. Through interactive methods like texts 'dramatization, drama and theatrical plays with the rest members of the museum visitors' group, having as a springboard the museum exhibits, the social interaction also could be enhanced [2]. The concept of interaction consists of animation as a catalytic factor and motive which could elect the interpretation and the museum exhibits' learning. In this way knowledge is constructed gradually while the museum visitor participates directly in the experiential procedure.

The interaction could also be the social interaction between museum visitors as members of a whole museum group and determines for a thorough accessing of the museum exhibits, the museum research and the possible results which could be formatted through that experiential procedure [8]. An important and functional point which could be part of an interactive planning is the feedback. Providing the museological planning with feedback is taken into consideration the fact that the direct springboard is the obtained cognitive background in the museum space, the experiential background of the museum visitor and as well as the possible reproduction of all this obtained information in order to obtain a further research and a continuous learning [10].

B. *Interaction through technology*

Initially, technology contributes in the digitization of the museum material collections, enhancing the necessary and appropriate points in order to preserve the documentation, the projection and the research with direct access in the wide public [11]. Terms which are stated such as digital, virtual and cybermuseum determine a part of the whole wide frame of technological contribution in a museum organization [7], as concerns content, context and space of existence.

Digitization of museum material collections offers the opportunity to immerse museum visitors (users) in a virtual museum tour, where visual experience of the museum content is possible [3].

Other interactive forms are through mechanisms such as wireless sensors, audiovisual devices-speakers/projectors (information overlay in smart rooms) where technology is part of the museum space but also the specialized interactive narrative with smart clothes in the form of a jacket or a vest with a small embedded computer and a lightweight headmounted display or glasses (where the technology is part of the museum visitor) providing the visitor with the opportunity to recall information from his experiential cognitive background in order to create a unique personal narration [14].

In the game interaction, the museum visitor has the chance to interact with virtual or real museum exhibits and constructions in a playful manner. The game is chosen as a method which could be a useful springboard under an educational and museum-pedagogical point of view accessible to the museum exhibits through technology [13].

Finally, the most frequent form of robot interaction in museums is the robot – guided tour, which either guides the museum visitors or performs the commands which are being given by the visitor [5], [9], [12], [15], [16], even with the support of augmented reality [17].

In order to create user friendly and functional activities, the robots' operation should be achieved through intuitive and human based interactive operation [18], [19].

III. Project "Ancient Greek Theater"

Based on the above museum-pedagogical principles about interactive learning activities, but also on research results about proper robotic lesson construction [20], [21], cross-curricular activities constitute a meaningful learning approach. The "Ancient Greek Theater" is a cross-curricular project, which aims to offer a study environment on Humanities and Technology topics, as shown in Table I.

TABLE I
CROSS-CURRICULAR TOPICS

subject	**topics**
Culture, Sociology, Literacy, Art, Geography	comedy and tragedy - spiritual values through ages politics and Democracy religion and ancient gods apparel & life habits ancient Greek theatres at the Mediterranean Sea
Sciences, Technology	acoustic marvel (filter and reflector) optics (theatre = viewing place) stable construction - static mechanical manipulators, cranes and other artefacts dimensions and analogies

A. Operation and Scenario

The Ancient Greek theater project (Fig. 1) is a representation of the ancient theater enhanced by robots. The involved robots automate the theater's functions (robot-crane), but also interact with the user as actresses (robot-dolls). Two cameras are being used to recognize the body motions of the participant students, who wear blue and yellow theater suits. Every motion of the student with blue suit causes the corresponding motion to the blue robot actress. The same operation of distance control works also for the yellow pair (student-actress and robot-actress). When the students manage to bring the robot-actresses in front of the scene, with proper motions, then the robot-crane brings a god onto the stage from above.

Fig. 1 Two students participate in the activity

The scenario of the activity is based on the play "Iphigenia in Tauris" by Euripides. It is a tragedy that was written between 414 and 412 BC. The work unfolds in scenes "in Tauris" where priestess Iphigenia offers libation to goddess Athena and begs for help.

The author, Euripides, at this point uses the story telling technique “Deus ex Machina” (god out of the machine), at which a seemingly inextricable problem is suddenly and abruptly solved with the contrived and unexpected intervention of a god. In ancient Greek theater this was carried out by the use of a crane which was bringing an actor-god onto the stage from above (Fig. 2).

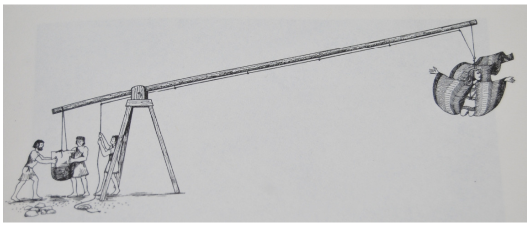


Fig. 2 Crane was used to bring actors (gods) onto the stage

B. Construction

The project has been constructed according to ancient theater’s analogies. Greek theater buildings were called a Theatron. The theaters were large, open-air structures constructed on the slopes of hills. They consisted of three principal elements: the Orchestra, the Skene (stage), and the Audience. The centrepiece of the theatre was the orchestra, or "dancing place", a large circular or rectangular area. Behind the orchestra was a large rectangular building called the skene. It was used as a "backstage" area where actors could change their costumes and masks, but also served to represent the location of the plays. Rising from the circle of the orchestra was the audience. The audience sat on tiers of benches built up on the side of a hill.

Fig. 3 Robot actress

In “ancient Greek theater” project, three robots have been involved. All robots are constructed with the use of Lego NXT Mindstorm educational sets. The two robots represent the priestesses (Fig. 3, 4) and the third materialize the crane “Deus ex Machina” (Fig. 5). In Fig. 4 we can see the mobile robot structure. It has three degrees of freedom: body straight movement, waist and hands rotation.

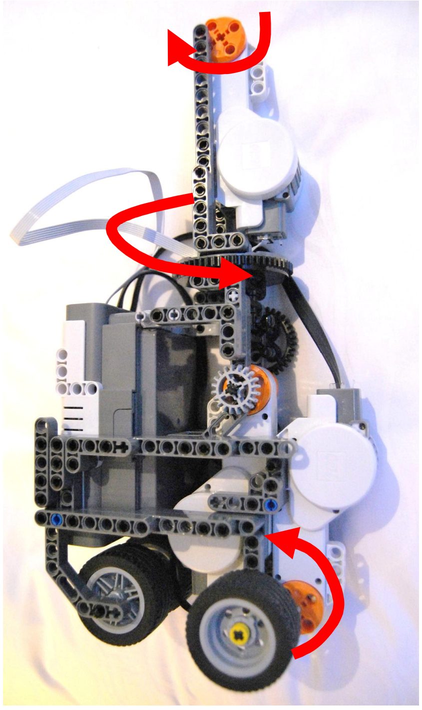


Fig. 4 Robot actress – 3 degrees of freedom

The robot-crane (Fig. 5) should be able to hang and transfer the god in any place above the stage. For this reason the architecture of an industrial robot arm has been adopted.

Fig. 5 The robot crane “Deus ex Machina”

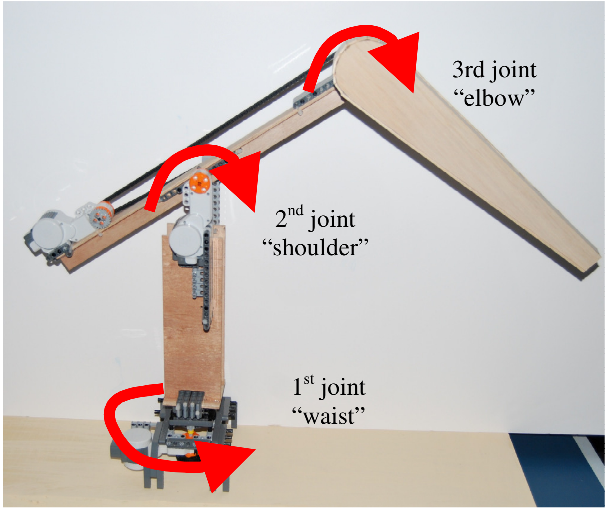


Fig. 6 The robot crane - architecture

The PUMA Unimation robot type (Fig. 7), is a 6 degree of freedom articulated robot, popular in industry and research institutes. The theater’s crane implements the first three joints and links of PUMA robot (Fig. 6).

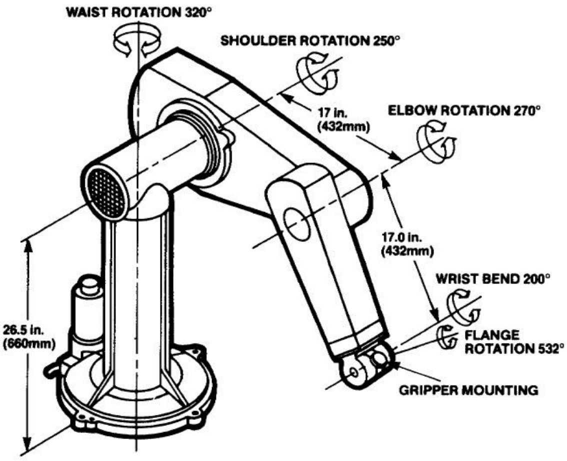


Fig. 7 Joints and rotation axes of PUMA

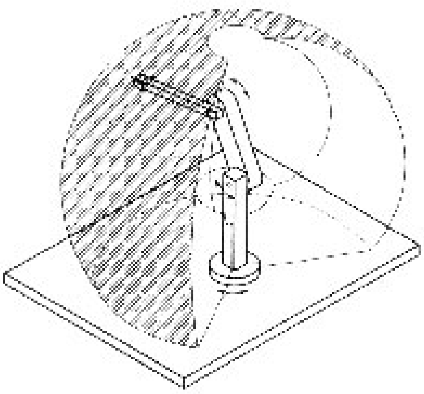


Fig. 8 Workspace of an articulated 3 DOF robot arm

The first three joints’ axes, give three degrees of freedom (waist, shoulder, elbow), which means free positioning in the space (independent X,Y,Z coordinates) (Fig. 8).

The system encompasses two NEXTcam v 3.0 Midsensors (Fig. 9). These cameras support real time image processing for identification (up to 8 objects) and communicate directly with the I2C bus of NXT.

The two cameras identify the bodies with the blue and yellow kirtles (clothes). The coordinates of the body are then extracted and filtered. Based on this information a forth NXT calculates in real time the path of the two mobile robots (dolls) and in this way the system follows the movements of the users.

The four NXTs, that the project encompasses, communicate with Bluetooth technology.

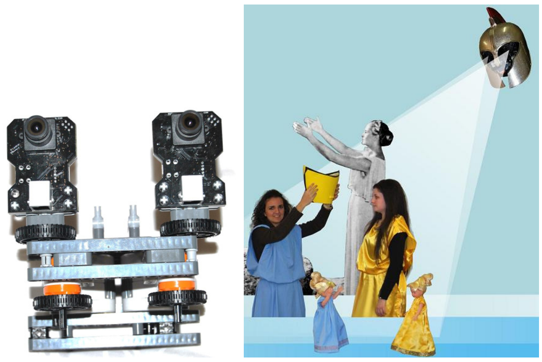


Fig. 9 The two NEXTcams and the vision system

IV. Learning Goals

The “ancient Greek theater” project implements many issues of the robotics technology which can be considered as the learning goals in the Science and Technology curriculum (Table II). The project includes four microcontrollers (NXTs), which are used in the control of mobile robots (the 2 dolls), in the control of a robot arm (the crane) and in the image processing and path creation (the eyes unit). All the 4 units collaborate and communicate, exploiting the Bluetooth ver. 2.0 capabilities of NXT. Control theory implemented in the closed loop control of the mobile robots (dolls), for real time participant’s motion following.

The robot arm has been developed based on the industrial PUMA Unimation robot arm and enables motions of three degrees of freedom (3 DOF), having the capability to reach any point in its working area. The project also uses position-rotation, distance and audio sensors, as well as two cameras.

TABLE II
Learning Goals as Technological Project

Robotics and Control	Cognitive goals
Robot design-construction	Industrial robot architecture (PUMA-Unimation) Balanced robot arm Mobile robots Cooperative robots
Control	Control theory – position control Closed loop feedback

Electronics	Optical sensors Position sensors
Communication	Bluetooth ver. 2.0 Multiple communication (4 NXT interaction)
Signal Processing	Real time image process Motion detection & following Multiple objects' motion following 3D view capability
Intelligence	No preprogrammed motion Real-time motion creation based on user's movements Body gesture recognition

V. Cross-Curricular goals and learning approach

As mentioned above, the "ancient Greek theater" offers a learning environment for the students, to work on cultural (play spiritual values), sociological (politics, life habits, religion and ancient gods), literary (drama, comedy), artistic (scenery, apparel, masks), geographical (ancient theatres at the Mediterranean Sea) topics. This cross-curricular approach has the advantage of the global view and also facilitates the investigation of topics connection.

The presentation of the project is based on active participation of the students (Table III). Students assume the role of actor of tragedy, wearing ancient Greek tunics, holding ancient religious object (amphora) and following the Greek ancient theatrical ritual, in front of the stage and the "viewers" and provoke the "Dues ex Machina".

The presentation of the cultural and technological aspects of the ancient Greek theatre is been achieved not in a passive way, in which the robots acts and the people just watching. In this performance type presentation, the person gets involved in a dialogic relation with the subject and the robots become the tools that support a cognitive process. The prevalent idea in the contemporary research of museum-pedagogy considers as major factor the personal engagement in a dialogic process [20]. Students take advantage of the dynamic environment of the robots, the "hand on" capability, the user driven action, the realistic and dramatised scenario and scene, to impart the cultural ethics, the literature aspects and the every day habits (like apparel) to the participant.

The participants have the opportunity to integrate with their heroes through the man-machine interaction experience of the ancient Greek tragedy. The two participants are asked to represent one of the most critical stages of the play (tragedy), in which two priestesses make supplication to the goddess. Specifically, Iphigenia with the escort of another priestess, beg the goddess Athena to help her come out of the moral dilemma. This learning approach and the project context also ensure that the impact of traditional gender roles are neutralized, promoting girls' engagement [20].

The movements, which are described by the project and participants are asked to implement, are natural, human, and realistic, supporting intuitive operation. The motion of robots is driven by the participant (through cameras usage), but also the robots shows to the participants if their effort has positive evolution (raising the hands, appearing the god out of the machine, etc), offering the necessary feedback. The whole interaction system relies on wireless communication and the cameras which are placed inside an ancient helmet, making the scene more natural and aesthetically beautiful. As no remote control used or any inelastic pre-programmed robot movements, the presentation gets a deep interactive format. Also, by an accurate representation of the scene and getting experience through the role playing game, we have the opportunity to incorporate an edutainment framework [22]. The achievement of edutainment is important especially for fields which are considered borrowing (by the students), like the ancient literature and ancient languages. In this project, participants come in direct contact with the ancient culture and understand the ancient ethics and principles advocated by the project, in a playful learning environment [23].

TABLE IIIII
Learning approach characteristics

Aspect	Approach
Presentation - Participation	Dynamic - "Hands on" type User based Realistic Scenario Theater's scene - Dramatized presentation Cultural ethics presentation Literature presentation (myths, tragedies) Apparel presentation
Man-machine interaction	Intuitive control Natural motions User friendly No remote control device
Understanding – Cognitive	Engaged learning Role playing game Accomplishment of objectives Experiential type comprehension Creative type comprehension Edutainment

VI. Educational Implementation

In the current pilot implementation, students from 4^{th} to 6^{th} grades (10-12 years old) of elementary school and 11^{th} grade (16 years old) of high school have participated in educational activities, based on the "Ancient Greek theater" project (Fig. 10). In the beginning, a museum-educator presents the ancient Greek theater topic, through discussion with students. This stage aims also to extract students' prior knowledge, in order to design and adapt properly the following activities. Then the students participate in pairs, trying to represent the ancient tragedy, by telling the story dialogues and wearing theater masks and suites. Their steps and movements aim to drive the robot actresses.

Exploiting the intuitive manipulation of the robots, students show a fluent operation of the system, with no orientation or kinetic problems. They were able to focus on the play and to cooperate with their teammates, in order to represent the play scene successfully. In this way, as they expressed, they learned the humanity topics of the project, in a playful and edutainment manner. In this environment they show continuing interest about topics, which in general have lack of participation in a typical classroom course.

Regarding the Science and Technological issues, curiosity and robots' interactive response drive them to participate enthusiastically in the investigation of the relative aspects. At the end of the activities they were able to recognize and explain project technology and to compare and correlate it with other industrial or even every day technologies.

Fig. 10 Elementary students participate in "Ancient Greek Theater"

VII. Conclusions

It is a common sense that the presence of the new technologies, by tangible or virtual means should contribute in a useful and functional manner in order to preserve the museological structure respecting educational aims without underestimating the value of the museum exhibits. Sometimes it is common knowledge that the technological and the interactive museum exhibits could distract the visitor from cancelling the possible experiential learning. Furthermore, the virtual or tangible technology representations sometimes are not able to transfer the meaning and the aims which are represented by the real exhibits and also they cannot replace the experience which could be obtained through the direct access with the real museum exhibits.

As the implementation of the project "Ancient Greek theater" showed that the exploitation of robots in museum activities can contribute to the achievement of museum-pedagogical aims. The cross-curricular design is an important factor for the balance between social-humanities and technological studies. Also the intuitive control and the playful character of the interactive environment create and support engaged and experiential learning, which lead to better comprehension.

Acknowledgment

The authors would like to thank the teachers and students of elementary and high schools in Florina, and Thermi for their participation to the project and their acceptance for interviewing and answering the research's questioner. This project has partially founded by the ACHLADA mining Ltd.

References

[1] N. Nikonanou, "The role of the museopedagogist in the contemporary museums," *Museologists Books*, vol.2, pp.18-25, 2002

[2] N. Nikonanou, "The organization of the communicative museum's policy and the role of the museopedagogy," *in Proc. 1st Int. Conf. on Museology: Museum, Communication and New Technologies*, Mitilini, Greece, 2004, paper p.77-92

[3] C. Dallas, "The presence of visitors in virtual museum exhibitions." *e-Museology: International scientific electronic journal*, 4: Performativity, interactivity, virtuality and the museum, p.5-18., 2007

[4] E.Nakou, Ed, *Museums : We, the Material (objects) and the Civilisation,* Athens, Greece: Nisos, 2001

[5] (2010) Sacramento Discovery Museum Science & Space Center to Unveil New Robotics Lab, [Online]. Available: http://museumpublicity.com/2011/02/16/sacramento-discovery-museum-science-space-center-to-unveil-new-robotics-lab/

[6] Roschelle (1995) Learning in Interactive environments-Prior Knowledge and New Experience, homepage on Association of Science-Technology centers. [online] Available: http://www.exhibitfiles.net/resource/education/priorknw.htm.

[7] C. Arvanitis, *Digital, Virtual, Cybermuseum or Network Museum? Searching for the Definition*," in Proc. Museum, Communication and New Technologies, First International Conference of Museology, Department of Cultural Technology & Communication, University of the Aegean, Greece, May 2002, p.183

[8] C. Heath and D. vom Lehn, "Misconstruing Interactivity," in *Proc. Interactive Learning in Museums of Art and Design, London: Victoria and Albert Museum*, 2002

[9] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G.Lakemeyery, D. Schulz, W.Steiner and S. Thrunz, "The Interactive Museum Tour-Guide Robot" in *Proc. AAAI-98*, 1998, p. 11

[10] S. E. Koester, "Interactive Multimedia in American Museums," Archives & Museum Informatics, Tech. Rep. ISSN 1042-1459 N0.16, 1993

[11] A. Bounia, The multimedia as an interpretative tool in Greek museums : General principles and issues, in Proc. Museum , Communication and New Tecnologies, Greece 2004, paper p.21.

[12] M. Oikonomou "New Technologies and Museums: a Tool, an Obstacle or Fashion?," *Museology International Scientific Electronic Journal*, vol.1, pp.1-14, 2004.

[13] R. Wakkary, K. Muise, K. Tanenbaum, M. Hatala and L. Kornfeld, "Situating Approaches to Museum Guides for Families and Groups," in *Proc International Cultural Heritage Informatics Meeting (ICHIM07)* J. Trant and D. Bearman (eds). Toronto: Archives & Museum Informatics. Published September 30, 2007 [Online]. Available: http://www.archimuse.com/ichim07/papers/wakkary/wakkary.html

[14] F. Sparacino, K. Larson, R. MacNeil, G. Davenport, A. Pentland, (1999) Technologies and Methods for Interactive Exhibit Design: from Wireless Object and Body Tracking to Wearable Computers, International Conference on Hypertext and Interactive Museums, ICHIM99, Washington, DC., 1999 [Online]. Available: http://en.scientificcommons.org/42903048

[15] C. Delgado-Mata, R. Velazquez, R. J. Pooley, R. Aylett, J. Robertson, MPISTE: A Mobile, "Personalised, Interactive Story Telling Environment", in *Proc. 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*, cerma, 2010, pp.243

[16] K. Fischer, Why it is interesting to investigate how people talk to computers and robots: Introduction to the special issue, Journal of Pragmatics, vol. 42, issue 9, September 2010, Pages 2349-2354

[17] B.-O. Han, Y.-H. Kim, K. Cho, H.S. Yang, "Museum tour guide robot with augmented reality," in Proc. VSMM'10, 2010 16th International Conference, Oct. 2010, pp. 223

[18] M. Nieuwenhuisen, J. Gaspers, O. Tischler, S. Behnke, "Intuitive multimodal interaction and predictable behavior for the museum tour guide robot Robotinho," in Proc. *10th IEEE-RAS Int. Conf. on Humanoid Robots, Humanoids 2010* , art. no. 5686848, pp. 653-658

[19] A. Yamazaki, K. Yamazaki, M. Burdelski, Y. Kuno, M. Fukushima, "Coordination of Verbal and Non-verbal Actions in Human-robot Interaction," *Journal of Pragmatics*, vol. 42, 9, p. 2398-2414, Sept. 2010

[20] I. Nourbakhsh, E. Hamner, B. Dunlavey, D. Bernstein and K. Crowley, "Educational Results of the Personal Exploration Rover Museum Exhibit," in *Proc. 2005 IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, 2005, pp. 4278-4283

[21] A. Wiesner-Steiner, H. Schelhowe and H. Wiesner, "The Didactical Potential of Robotics for Education with Digital Media," *Int. J. of Information and Communication Technology Education*, vol. 3,1, 2007

[22] O. Zühal, "Edutainment: is learning at risk?," *British Journal of Educational Technology,* vol. 34,3, 2003, pp. 255-264.

[23] M. Resnick, "Edutainment? No Thanks. I Prefer Playful Learning." In *Associazione Civita Report on Edutainment,* 2004

Index of authors

The conference is jointly organized by:

innoc

ÖSTERREICHISCHE GESELLSCHAFT FÜR INNOVATIVE COMPUTERWISSENSCHAFTEN

AUSTRIAN SOCIETY FOR INNOVATIVE COMPUTER SCIENCES

BMWF[a]

Bundesministerium für Wissenschaft und Forschung

Technically sponsored by:

IEEE

AUSTRIA SECTION

www.rie2011.org

ISBN: 978-3-200-02273-7