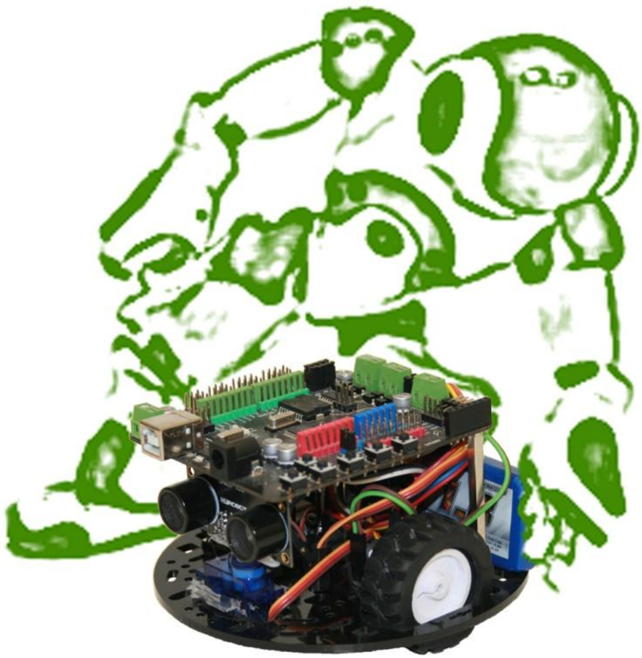
4th International Conference on Robotics in Education

RiE 2013

September 19-20, 2013

Łódź, Poland



4th International Conference on Robotics in Education

RiE 2013

19-20 September 2013, Lodz University of Technology,

Poland

Conference proceedings

Integrated part of this proceedings is an electronic material available on CD or pendrive

Published by the Faculty of Electrical, Electronic, Computer and Control Engineering, Łódź 2013

Editor: Grzegorz Granosik

ISBN: 978-83-937551-9-6

Patronage:

Robotix Week Welcome

It is my pleasure to introduce you to the Technical Program of the Robotix Week and the Proceedings of the 4th International Conference on Robotics in Education (RiE 2013), Łódź, Poland, September 19-20, 2013. All contributed papers are included in this publication as an electronic attachment.

After successful editions in Bratislava, Wien and Prague this Conference is organized by the Institute of Automatic Control at the Lodz University of Technology, Poland. With the Honorary Patronage of Rector of the Lodz University of Technology and technical co-sponsorship of IEEE Poland. With great support from Partners: National Instruments, KUKA Roboter CEE, RoboNet, SEP Students Group and Sponsors: Ericpol, International Visegrad Fund, City of Łódź and Młodzi w Łodzi.

This year we received a total of 23 submissions from 14 countries. Each manuscript had two reviewers. Twenty two original works have been selected for oral presentation on RiE 2013. The best 11 papers will be published in the Journal of Automation, Mobile Robotics & Intelligent Systems JAMRIS.

I am grateful to the invited distinguished plenary speakers: Prof. Andrea Bonarini, Politecnico di Milano, Italy; Prof. Edward Jezierski, Lodz University of Technology, Poland; and representatives of our Partners, who prepared a total of 5 plenary talks.

I would like to express my thanks to all members of the International Program Committee who did very hard work in evaluating all papers. Many thanks to members of Organizing Committee for all help and constructive criticism. Finally, I would like to express my thanks to all participants for their help in keeping high standards of the Conference.

RiE 2013 is traditionally followed by the robotic competitions Robotour (September 20-22) where autonomous mobile robots travel in park on the distance of about 1000 m. Additionally, this year Conference is accompanied by the Visegrad Robotics Workshop (September 17-18) providing hands-on experience with mobile robots. Therefore, the Robotix Week is full of activities with robots.

I wish you a memorable stay in Łódź and Poland.

Welcome to Robotix Week, RiE 2013 and the Lodz University of Technology.

Grzegorz Granosik

International Program Committee

- Dimitris Alimisis, School of Pedagogical and Technological Education, Athens, GR
- Richard Balogh, Slovak University of Technology in Bratislava, SK
- Jenny Carter, De Montfort University in Leicester, UK
- Jean-Daniel Dessimoz, Western Switzerland University of Applied Sciences, Yverdon-les-Bains, CH
- Hugo Ferreira, Instituto Superior de Engenharia do Porto, PT
- Grzegorz Granosik, Lodz University of Technology, PL
- Alexander Hofmann, University of Applied Sciences Technikum Wien, AT
- Peter Hubinský, Slovak University of Technology in Bratislava, SK
- Boualem Kazed, University of Blida, DZ
- Tomáš Krajník, University of Lincoln, UK
- Miroslav Kulich, Czech Technical University, CZ
- Dariusz Marchewka, AGH University of Science and Technology in Kraków, PL
- Martin Mellado, Universitat Politècnica de València, ES
- Michele Moro, University of Padova, IT
- David Obdržálek, Charles University in Prague, CZ
- Pavel Petrovič, Comenius University in Bratislava, SK
- Colin Sauzé, University of Wales in Aberystwyth, UK
- Alexander Schlaefer, Universität Lübeck, Deutschland, DE
- Dietmar Schreiner, Vienna University of Technology, AT
- František Šolc, Brno University of Technology, CZ
- Gerald Steinbauer, Graz University of Technology, AT
- Roland Stelzer, INNOC - Austrian Society for Innovative Computer Sciences, AT
- Francis Wyffels, Ghent University, BE

Robotix Week Organizing Committee

- Grzegorz Granosik, Chairman
- Artur Gmerek
- Jacek Kabziński
- Martin Dlouhý, Robotour
- Igor Zubrycki, Visegrad Workshop
- Michał Maciejewski, Visegrad Workshop
- Marek Gawryszewski
- Przemysław Dawid & Piotr Kmiecik

date	17.09.2013	18.09.2013	Thursday 19.09.2013	Friday 20.09.2013	21.09.2013	22.09.2013
time	Visegrad Robotics Workshop		Conference on Robotics in Education 2013		Robotour Competitions	
9:00	Micromouse I	LabVIEW Robotics I	opening ceremony	plenary session III Prof. Edward Jezierski	Warm up	Robotour Workshop
9:20						
9:40			plenary session I Prof. Andrea Bonarini	plenary session IV Kuka Roboter		
10:00					Robotour round I	
10:20				coffee break		
10:40	coffee break	coffee break	coffee break	plenary session V NAO robots		
11:00	Micromouse II	LabVIEW Robotics II	plenary session II National Instruments		Robotour round II	
11:20				session D		
11:40			session A			
12:00					Robotour round III	
12:20				session E		
12:40	lunch	lunch				
13:00			lunch		Robotour round IV	
13:20				lunch		
13:40	Micromouse III	LabVIEW Robotics III				
14:00			session B		Awards	
14:20				session F		
14:40						
15:00			session C			
15:20				session G		
15:40						
16:00			coffee break			
16:20			city tour	closing ceremony		
16:40				coffee break		
17:00				Robotour open tryouts		
17:20						
17:40						
18:00	dinner	dinner				
18:20						
19:20			Conference dinner	Get together party		

Thursday 19.09.2013				
Plenary Session I				
9:40	Andrea Bonarini	Robotics and Design: interdisciplinary courses	Italy	001_P1
Plenary Session II				
11:00	Maciej Antonik	From Robotics Theory to Practice: Graphical System Design with NI LabVIEW	Poland	002_P2
A	**Robotics in primary curriculum**	Chair: **Andrea Bonarini, Italy**		
11:40	Dave Catlin	A Day in the Life of an Educational Robot	England	003_A1
12:00	Mícheál Ó Dúill	Robotics when I want my two front teeth: Issues around the core curriculum of primary	Bulgaria	004_A2
12:20	Jarosław Kotliński and Grzegorz Troszynski	Will robots take over Polish gymnasiums?	Poland	005_A3
12:40	Dorit Assaf	Building a Rube Goldberg Machine using Robotic Toolkits. A Creative, Hands-on Exercise for	Switzerland	006_A4
B	**Robotics at university**	Chair: **David Obdržálek, Czech Republic**		
14:00	Cem Avsar, Walter Frese, Thomas Meschede and Klaus Brieß	Developing a Planetary Rover with Students: Space Education at TU Berlin	Germany	007_B1
14:20	Nelson David Munoz, Jaime Valencia and Alessa Alvarez	Simulation and Assessment Educational Framework for Mobile Robot Algorithms	Colombia	008_B2
14:40	Nelson David Munoz, Jaime Valencia and Alessa Alvarez	Suggestions for the assessment of navigation algorithms in educational robotics.	Colombia	009_B3
C	**New methods of teaching robotics at schools**	Chair: **Dorothy Langley, Israel**		
15:00	Martin Kandlhofer, Gerald Steinbauer, Sabine Hirschmugl-Gaisch and Johann Eck	A cross-generational robotics project day: Pre-school children, pupils and grandparents learn together	Austria	010_C1
15:20	Amy Eguchi	Educational Robotics for Promoting 21st Century Skills	USA	011_C2
15:40	Dorothy Langley, Yair Zadok and Rami Arieli	Exploring Spatial Relationships: A Strategy for Guiding Technological Problem Solving	Israel	012_C3
Friday 20.09.2013				
Plenary Session III				
9:00	Edward Jezierski	The balance between mechanical, electrical/ele-ctronic and control aspects in robotics education	Poland	013_P3
Plenary Session IV				
9:40	Janusz Jakieła	KUKA Education Bundle: Know-how for future professionals	Poland	014_P4

Plenary Session V				
10:40	Damian Derebecki	NAO robot in Education	Poland	015_P5
D	**General robotic education**	Chair: **Edward Jezierski, Poland**		
11:20	Daniel López and Martha Ivón Cárdenas	Programming in the Real Word: Initiation and Motivating Challenges of Entrepreneurship	Spain	016_D1
11:40	Reinhard Gerndt, Ina Schiering and Jens Lüssem	Elements of Scrum in a Students Robotics Project - A Case Study	Germany	017_D2
12:00	Elisa Buselli, Francesca Cecchi, Giacomo Santerini, Pericle Salvini and Paolo Dario	Building a network of schools on Educational Robotics in Tuscany area	Italy	018_D3
E	**Educational robotic technologies**	Chair: **Michele Moro, Italy**		
12:20	Thomas Kittenberger, Andreas Ferner and Reinhard Scheikl	A simple computer vision based indoor position-ning system for educational micro air vehicles	Austria	019_E1
12:40	Cesar Vandevelde, Jelle Saldien, Maria-Cristina Ciocci and Bram Vanderborght	Overview of Technologies for Building Robots in the Classroom	Belgium	020_E2
13:00	Witold Pawlowski, Michal Krepski and Slawomir Gabara	Didactic Automated Assembly Stand	Poland	021_E3
F	**Educational Robotic Systems**	Chair: **Thomas Kittenberger, Austria**		
14:20	Yuri Okulovsky, Pavel Abduramanov, Maxim Kropotov and Anton Ryabykh	CVARC: an educational project for a gentle introduction to autonomous robots' control	Russia	022_F1
14:40	Stefano Michieletto, Stefano Ghidoni, Enrico Pagello, Michele Moro and Emanuele Menegatti	Why teach robotics using ROS	Italy	023_F2
15:00	Igor Zubrycki, Grzegorz Granosik	Introducing modern robotics with ROS and Arduino	Poland	024_F3
G	**Robotic competitions**	Chair: **Richard Balogh, Slovakia**		
15:20	Wojciech M. Czarnecki, Krzysztof Szarzyński and Andrzej Wójtowicz	Designing a competition for autonomous robots with a restricted set of sensors with a case study of LEGO NXT	Poland	025_G1
15:40	Amy Eguchi and Luis Almeida	RoboCupJunior: Promoting STEM Education with Robotics Competition	USA Portugal	026_G2
16:00	Marco Cigolini, Alessandro Costalunga, Federico Parisi, Marco Patander,Isabella Salsi, Andrea Signifredi, Davide Valeriani, Dario Lodi Rizzini, Stefano Caselli	Lessons Learned in a Ball Fetch-And-Carry Robotic Competition	Italy	027_G3

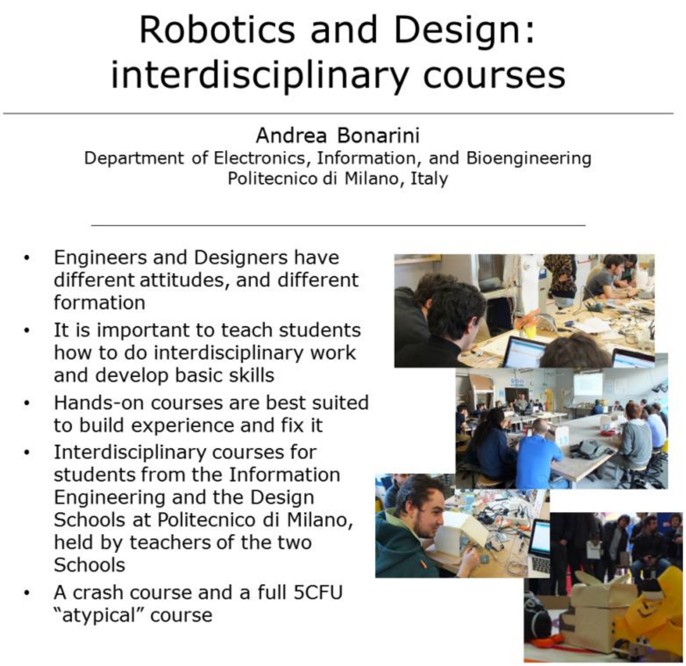
Plenary Session I

Robotics and Design: interdisciplinary courses

Prof. Andrea Bonarini
Artificial Intelligence and Robotics Lab
Department of Electronics, Information, and Bioengineering
Politecnico di Milano
Milan, Italy

Thursday, 19.09.2013 9:40-10:40

Andrea Bonarini is full professor at the Politecnico di Milano, Department of Electronics, Information, and Bioengineering. Coordinator of the Artificial Intelligence and Robotics Lab (AIRLab) since 1990. He is among the founders of the Italian Association for Artificial Intelligence (AI*IA) and the Italian Chapter of the IEEE Computational Intelligence Society (Chair from 2008 to 2010). Since 1997 he participated to the Robocup initiative (member of the Executive Committee from 2002 to 2010). He has participated and led several EU, national, and industrial projects. Since 1989, he has developed with his collaborators and students more than 30 autonomous robots. His research focus is currently on interactive, autonomous robots, in particular for Edutainment and Robogames. He has published more than 140 papers on international journals, books, and proceedings of international congresses.



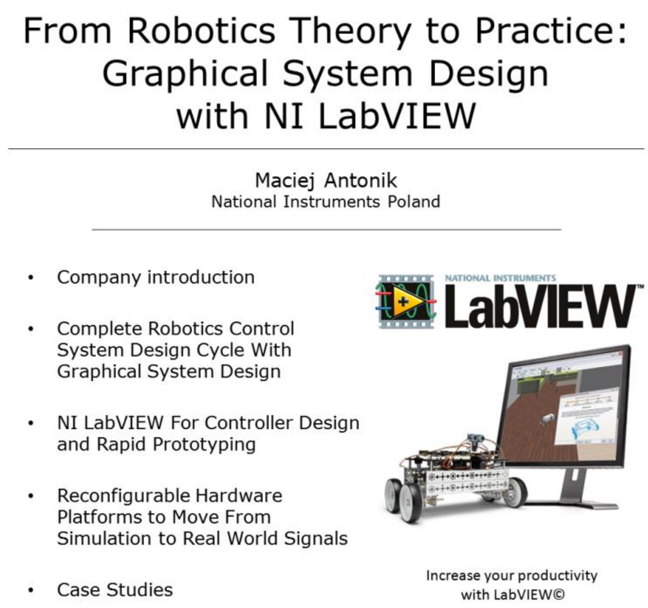
Plenary Session II

From Robotics Theory to Practice: Graphical System Design with NI LabVIEW

Maciej Antonik
Academic Engineer
National Instruments Poland

Thursday, 19.09.2013 11:00-11:40

Robots mean many things to many people, and **National Instruments** offers intuitive and productive design tools for everything from designing autonomous vehicles to teaching robotics design principles. The NI LabVIEW graphical programming language makes it easy to build complex robotics applications by providing a high level of abstraction for common tasks, such as sensor communication, obstacle avoidance, path planning, kinematics, steering and more. All of above mentioned features are accessible even for the beginners due to the simplifying factor brought by the graphical nature of NI LabVIEW. Engineers are able to import code from other languages, such as C, MATLAB or HDL – and perform communication external device while focusing strictly on the application, not necessarily on the way of achieving the specified functionalities. Open RIO architecture gives the power of programming the reconfigurable FPGA devices to everyone. From simple thermocouple measurements up to ultra-fast FPGA-enabled camera systems – this holistic approach is the key to highly efficient and robust development.
National Instruments Poland is bringing all these tools to the Polish market.



Session A: **Robotics in primary curriculum** Chair: **Andrea Bonarini, Italy** Th. 11:40-13:00

A Day in the Life of an Educational Robot

Dave Catlin
Valiant Technology Ltd, UK

- A Report and Analysis of a School Working with Educational Robots
- The project looks at a Star Wars Project in a UK Primary School
- It summarises the ERA (Educational Robotic Activity) Principles
- It demonstrates how ERA can provide a meta analysis of the value of educational robots

Students celebrating the successful completion of a Roamer robot Task.

Robotics when I want my two front teeth

Mícheál Ó Dúill (Mike Doyle)
Logios.Org (& Euro Lyceum, Sofia, Bulgaria)

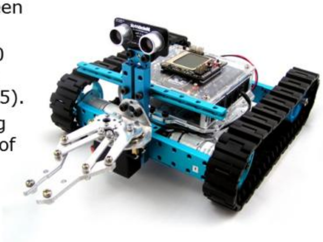
- Papertian objects are cognitively more powerful than natural language.
- Primary education builds the trunk of the learning tree, the branches follow.
- Robotics must fully support the core of literacy, numeracy and science.
- Comparison of pictographic, kit-based short-course robotics teaching with a curriculum approach shows the latter is more supportive of core objectives.
- The curriculum approach revealed a significant weakness in the traditional method of teaching mathematics.
- In the light of the technicity proposal, should construction and programming be given a higher priority in primary education? If so, how?

Primary Education is the trunk of a tree that supports the subjects of academe.

Will robots take over Polish gymnasiums?

Jarosław Kotliński
Faculty of Education, University of Lower Silesia in Wrocław
Grzegorz Troszyński
robotyedukacyjne.pl

- An attempt at implementing mechatronics as regular compulsory classes in gymnasiums (students between the ages 14-16).
- The project in numbers: 2700 students, 32 schools, 2 years (September 2013 – June 2015).
- New paradigm of the teaching model and the changing role of the teacher.



M-system robot

Building a Rube Goldberg Machine using Robotic Toolkits

Dorit Assaf
Artificial Intelligence Laboratory
Department of Informatics, University of Zurich, Switzerland

- A creative, hands-on exercise for secondary school students.
- Focuses on free exploration and experimentation. It is encouraged to use any material that can be found.
- An alternative to game-like robot competitions such as robot soccer, sumo, etc.
- Enables open-ended play to foster creativity.
- Trains important skills such as team work, problem solving, and time management.

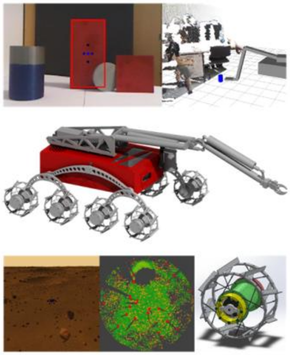
Students building a Rube Goldberg machine

Session B: **Robotics at university** Chair: **David Obdržálek, Czech Republic** Th. 14:00-15:00

Developing a Planetary Rover with Students: Space Education at TU Berlin

Cem Avsar, Walter Frese, Thomas Meschede and Klaus Brieß
Chair of Space Technology, Technische Universität Berlin, Germany

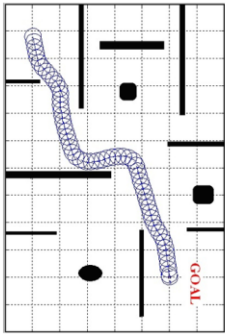
- An outline on TU Berlin's teaching activities in hands-on space education
- Participation in the DLR SpaceBot Cup 2013, a space robotics competition in the context of planetary exploration
- Development of a high-level autonomous rover with a team of undergraduate students
- System concepts for autonomous navigation, object recognition and manipulation



Simulation and Assessment Educational Framework for Mobile Robot Algorithms

Nelson Munoz Ceballos[1,2], Jaime Valencia[3], Alessa Alvarez Giraldo[1,3]
[1] *Colombian Politechnic Jaime Isaza Cadavid,* [2]*Metropolitan Technological Institute,* [3]*University of Antioquia, Medellín-Colombia*

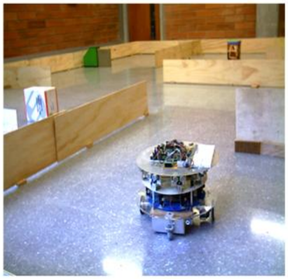
- A mobile robot simulator useful in research and education was implemented in Matlab, it models the differential kinematics as well as proximity sensors of the robot.
- It allows the performance assessment of navigation algorithms through various quality metrics that are useful for comparing and analyzing navigation algorithms of mobile robots.
- An example simulating and comparing two autonomous navigation algorithms is presented.



Suggestions for the assessment of navigation algorithms in educational robotics

Nelson Munoz Ceballos[1,2], Jaime Valencia[3], Alessa Alvarez Giraldo[1,3]
[1] *Colombian Politechnic Jaime Isaza Cadavid,* [2]*Metropolitan Technological Institute,* [3]*University of Antioquia, Medellín-Colombia*

- A simple way of teaching the topics performance metrics and benchmarks in educational robotics is presented.
- Aspects related with the performance assessment of navigation algorithms in mobile robotics are described.
- An example and the results obtained with a mobile robot in simulation and in a real environment are presented.



Robot Giraa_02

Session C: **New methods of teaching robotics at school** Chair: **Dorothy Langley, Israel** Th. 15:00-16:00

A cross-generational robotics project day

Martin Kandlhofer and Gerald Steinbauer
Institute for Software Technology, Graz University of Technology, Austria
Sabine Hirschmugl-Gaisch and Johann Eck
University of Teacher Education Styria, Austria

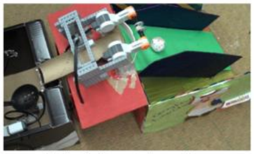
- Pre-school children, young students and grandparents learn together
- Integration of different age groups, different robotics platforms and different scientific institutions
- Robotics day in kindergarten offering eleven different hands-on experiments
- First qualitative evaluation investigating impact on young students

Educational Robotics for Promoting 21st Century Skills

Amy Eguchi
Division of Education, Bloomfield College, USA

- This paper introduces an educational robotics course offered as an Interdisciplinary Studies Course under General Education category at a liberal art college that serves predominately underprivileged population of students from neighboring communities in New Jersey, USA.
- It also presents the case study to examine participated students' learning from the course.
- The results show that the participating students identified their learning of collaboration and cooperation skills as well as communication skills as one of the best learning outcomes from the course.

Robotic Vending Machine

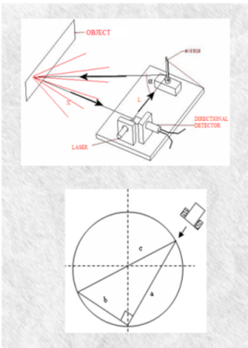


Robotic Donuts Launcher

Exploring Spatial Relationships: A Strategy for Guiding Technological Problem Solving

Dorothy Langley, Yair Zadok and Rami Arieli
Holon Institute of Technlogy, Israel; College for Academic Studies, Israel;
Weizmann Institute of Science, Israel.

- Instructional strategies are required to facilitate the transition of science majors' knowledge and skills from traditional school settings to project based learning environments.
- Visualization of the problem space and guided exploration of its spatial relationships can promote the elicitation of relevant formal knowledge and lead to creative solution design.
- These methods are described in the contexts of designing and programming robot navigation and in developing remote distance sensors.



Plenary Session III

The balance between mechanical, electrical/electronic and control aspects in robotics education

Prof. Edward Jezierski
Robotics Group, Institute of Automatic Control
Faculty of Electrical, Electronic, Software and Control Engineering
Lodz University of Technology
Łódź, Poland

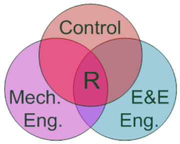
Friday, 20.09.2013 9:00-9:40

Edward Jezierski is the head of the Robotics Group at the Lodz University of Technology (TUL), Poland. His research activity is connected with dynamics and control of robots. Since 1990 he is offering courses in robotics at all three levels at TUL. He also has some international experience in education of this field. Prof. Jezierski is the author of two books devoted to students: Robotics – Basic Course (2002) and Dynamics of Robots (2006), both in Polish.

The balance between mechanical, electrical/electronic and control aspects in robotics education

Edward Jezierski
Robotics Group, Lodz University of Technology, Poland

- Different background of students starting to learn robotics cause differentiating the approach to built and perform the robotic course.
- Core components a robotic course: mathematical preliminaries, kinematics for position and velocity, static forces, dynamics of the manipulator, robot components, trajectory planning, basic control rules, the structure of control system, robot programming.
- A unified approach to analysis of mechanical and electrical subsystems is possible using similarities in transfer and accumulation of energy.



Basic components of Robotics (R)

Plenary Session IV

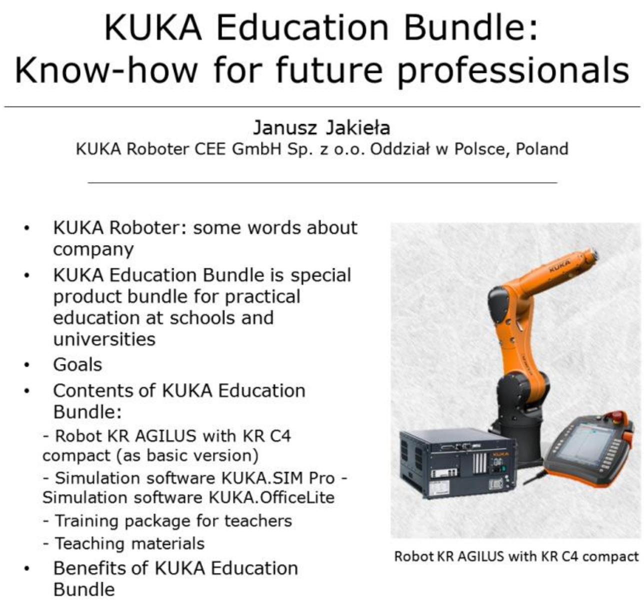
KUKA Education Bundle: Know-how for future professionals

Janusz Jakieła
KUKA Roboter CEE GmbH Sp. z o.o. Oddział w Polsce
Poland

KUKA

Friday, 20.09.2013 9:40-10:20

KUKA Roboter
Industrial robots, control systems and software. Creativity, Reliability, Dynamics. The widest range of products on the market.
KUKA Roboter is one of the world's leading suppliers of industrial robots - number one in Europe, number one in automotive industry worldwide and second place worldwide.
KUKA has already over 150 thousand robots with payloads from 5 to 1300 kg working in various applications.
Robots supplied by KUKA are most commonly used in industrial applications: spot/arc welding, handling, palletizing, packaging, machining and other automated processes.
Kuka is applying the expertise it has been acquiring for over 30 years in the automotive industry to develop innovative automation solutions for other segments, for example medical technology, the solar industry and the aerospace industry.
KUKA in Poland offers: robot sales, consulting, training, technical support and service.



Plenary Session V

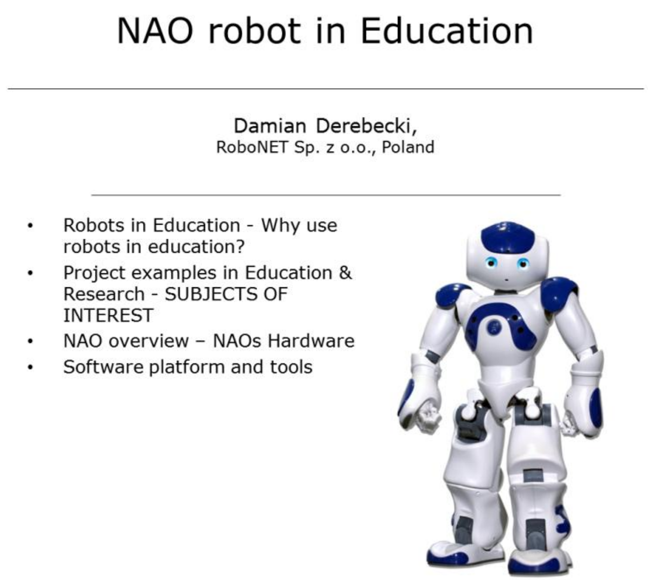
NAO robot in Education

Damian Derebecki
RoboNET Sp. z o.o., Gdańsk, Poland

Friday, 20.09.2013 10:40-11:20

RoboNET group is composed of three independent, yet complementary brands. **RoboFACTORY** is a team of physicists, mathematicians, computer programmers and electronics and robotics specialists. Together they under-take the most ambitious technical challenges to create solutions that will make the world a better place. **RoboCAMP** is a bold idea for a change in the education system of Poland with the help of an innovative educational program for children – it proves that kids are able to learn science while having a great time building robots. **RoboSHOP** is an online store, selling advanced educational toys for children, as well as professional mechatronic, electronic and measuring devices for universities and scientific institutes. From 1st July **RoboSHOP** is a distributor of the Humanoid robot **NAO**, one of the most advanced and versatile robot destined for educational and research purpose.

The humanoid platform NAO gained it's great popularity by wide variety of applications. The thing that amazes the most in NAO is the way how it links people of highly different abilities and interests, by giving them possibility to learn various issues: mechanics, computer sciences, control engineering, and mechatronics. The humanoid robot NAO gives everyone the opportunity to program and create realistic behaviors, thanks to its Multilanguage cross-platform, and programs such as Choreograph and Webots for NAO. This Robot is not only easy to program and control, but also monitor and draw conclusions from his numerous sensors.



Session D: **General robotic education** Chair: **Edward Jezierski, Poland** Fr. 11:20-12:20

Programming in the Real World: Initiation and Motivating Challenges of Entrepreneurship

Daniel López
Robotics Laboratory, Dept of Technology, Institut Font del Ferro, Spain
Martha I. Cárdenas
Department of Languages and Computer Systems (LSI), UPC, Spain

- Since nowadays the Enterprise and the High Technology go together hand to hand, trying to reach the control and understanding of real world processes, high school students may recreate it with robotic simulations and enhance their practical training in entrepreneurship.
- Educational Robotic projects within the field of Technology, Science and Maths, may encourage them to create and develop solutions for many real applications which includes their design, monitoring and control.

Robotic process applied to logistics

Elements of Scrum in a Students Robotics Project - A Case Study

Reinhard Gerndt, Ina Schiering
Ostfalia University of Applied Sciences, Germany
Jens Lüssem
University of Applied Sciences Kiel, Germany

- Robotics competitions allow self-organised learning
- Increasing complexity of competitions induces the need for an adequate project organisation
- We adopted Scrum and tried to adapt this agile methodology to student projects
- First experiences, feedback of the students and ideas for adjustments are discussed

Project management in Robotics

Building a network of schools on Educational Robotics in Tuscany area

Elisa Buselli, Francesca Cecchi, Giacomo Santerini, Pericle Salvini and Paolo Dario
The BioRobotics Institute, Scuola Superiore Sant'Anna, Italy

- The work presents the creation of a network of schools on Educational Robotics in Tuscany, Italy.
- The network started from 6 classes and it expanded up to 22 classes and 500 students.
- Robotics Laboratories involved Primary, Secondary and High school students with different activities for each level.
- This approach gives emphasis to engaging students, feeding curiosity, encouraging autonomous exploration and experimental activities.
- Preliminary results reveal that students appreciated the activities (89%); teachers evaluated the activity as very efficient and suggest to organize courses to improve their autonomy.

Regional Network of school and students during laboratories

Session E: **Educational robotic technologies** Chair: **Michele Moro, Italy** Fr. 12:20-13:20

A Computer Vision Based Indoor Positioning System for MAV's

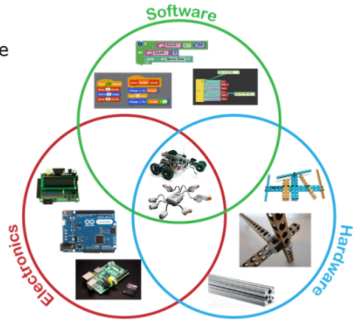
Thomas Kittenberger, Andreas Ferner, Reinhard Scheikl
Department of Embedded Systems,
University of Applied Sciences Technikum Wien, Austria

- Computer Vision System with Gumstix Overo Firestorm computer and Gumstix Caspa FS camera, 40 grams
- Three wide angle high power IR LED's on a tripod for position reference
- Camera resolution 752x480
  Frame rate 12 fps
  Range 10 m
  Accuracy 2%
- Geany IDE
  Ångström Linux
  OpenCV Library
  Custom algorithm for three point pose estimation

Overview of Technologies for Building Robots in the Classroom

Cesar Vandevelde, Jelle Saldine and Maria-Cristina Ciocci
Howest University College, Ghent University, Belgium
Bram Vanderborght
Dept. of Mechanical Engineering, VUB, Belgium

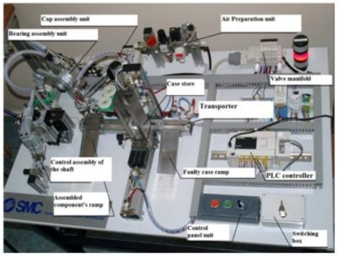
- Which complete robotics systems are available for use in education?
- Which options exist for the hardware, electronics and software when building a robot from scratch?
- DIY / Maker Movement in classrooms as a way to build robots.



Didactic Automated Assembly System

Witold Pawłowski, Michał Krępski
Institute of Machine Tools and Production Engineering, Faculty of Mechanical Engineering, Lodz University of Technology, Poland
Sławomir Gabara
SMC Industrial Automation Polska Sp. z o.o., Poland

- Assembly System of:
  - the case,
  - the bearing,
  - the shaft,
  - the cap.
- Drive of the system: pneumatic.
- Control of the system: PLC and Profibus.
- Failure simulations: the switch box.
- The students learn how to:
  - construct the assembly line,
  - analyze the work cycle,
  - program the PLC with Profibus,
  - failure check.



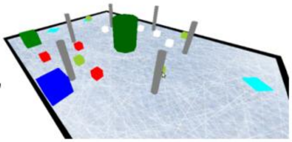
Automated assembly system

Session F: **Educational Robotic Systems** Chair: **Thomas Kittenberger, Austria** Fr. 14:20-15:20

CVARC: a gentle introduction to autonomous robots' control

Yuri Okulovsky, Pavel Abduramanov, Maxim Kropotov, Anton Ryabykh
Institute for Mathematics and Mechanics, Ural Federal University, Russia

- CVARC meant to introduce software engineering students into robotics
- CVARC is a system that allow creation of virtual robotics competitions
- No software to install and no ne programming languages – just join and try!
- The software is tested on the first virtual robotics competitions in the Ural Federal University
- New competitions with gradually growing complexity will start at October!

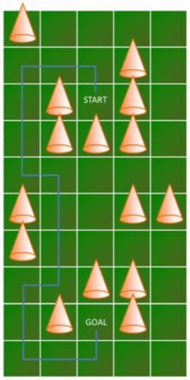


Robots competing in CVARC competitions

Why teach robotics using ROS

Stefano Michieletto, Stefano Ghidoni, Enrico Pagello, Michele Moro, and Emanuele Menegatti
Department of Information Engineering, University of Padova, Italy

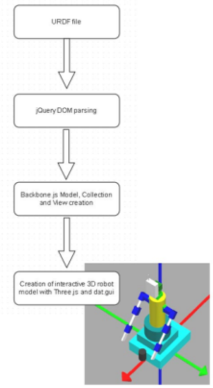
- A set of experiences for teaching robotics to master students is presented
- The use of ROS for developing the experiences was introduced in the last two years
- A comparison of the different skills acquired by students (with and without ROS) was performed
- Students' satisfaction was measured using a detailed questionnaire



Introducing modern robotics with ROS and Arduino

Igor Zubrycki and Grzegorz Granosik
Institute of Automatic Control, Lodz University of Technology, Poland

- Paper describes our experiences with introducing modern robotics through Robot Operating System to students, that are from different disciplines than Computer Science.
- We have designed a framework that makes the learning process easier as well as gives realistic examples of using ROS.
- Our online application: *MyModel Robot* provides students with easy way to experiment and learn URDF language used to describe robots in ROS



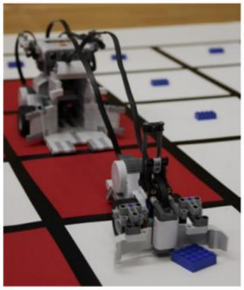
MyModel Robot application diagram

Session G: **Robotic competitions** Chair: **Richard Balogh, Slovakia** Fr. 15:20-16:20

Designing a competition for autonomous robots with a restricted set of sensors with a case study of LEGO NXT

Wojciech Czarnecki, Krzysztof Szarzyński, Andrzej Wójtowicz
Faculty of Mathematics and Computer Science,
Adam Mickiewicz University in Poznań, Poland

- The paper discusses issues and solutions of designing competitions for robots with a restricted set of sensors.
- We present limitations of the LEGO NXT sensors and their usage in competitions.
- We propose rules for ensuring equal chances as well as lack of randomness.
- We show evaluation of our ideas through three years of PozRobot competitions and introduce one of the most successful tasks.

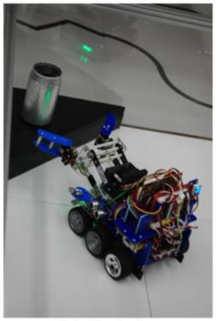


A LEGO NXT robot participating in PozRobot competition

RoboCupJunior: Promoting STEM Education with Robotics Competition

Amy Eguchi
Division of Education, Bloomfield College, USA
Luis Almeida
Dep. de Eng. Electrotécnica e de Computadores, Universidade do Porto/Faculdade de Engenharia, Portugal

- This paper addresses RoboCupJunior, an educational robotics initiative that aims to enhance learning through educational robotics competitions around the world.
- RoboCupJunior is a division of RoboCup, a robotics initiative that aims to promote Robotics and AI research.
- This paper reports the most recent study conducted during RoboCupJunior 2012 competition and discusses the positive results that this study shows. The results of the study highlighted that the participation in RoboCupJunior competitions enhances the participating students' interests in STEM fields and studying further in post-secondary education.

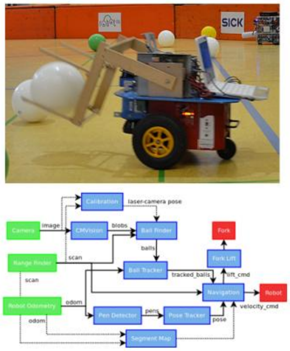


RoboCupJunior Rescue A Robot

Lessons Learned in a Ball Fetch-And-Carry Robotic Competition

Marco Cigolini, Alessandro Costalunga, Federico Parisi, Marco Patander, Isabella Salsi, Andrea Signifredi, Davide Valeriani, Dario Lodi Rizzini, Stefano Caselli
University of Parma

- Robotic competitions allow learning of engineering design
- Foraging task was the theme of the Sick Robot Day 2012 competition
- Robots had to perceive colored balls, grasp and transport them, localize and navigate to assigned areas
- Lessons learned through experiments allowed *Red Beard Button* team to achieve first place



The robot grasping the ball (top) and its control architecture (bottom)

Authors index

Robotour competitions

The objective of the Robotour contest is to encourage the development of robots capable of transporting you to work in the morning or delivering the building material you have just purchased in an online shop. Reaching this goal is neither easy nor short, but we believe that the outcome is worth all.

The task for the robots is to deliver payload in given 1 hour time limit to destination as far as 1 km. Robots must be fully autonomous, not leave the road and choose the correct path on junctions. The place of start and destination will be the same for all robots.

The robots can only use Open Street Map. The key concept of this map is its verifiability. Anything that is verifiable and is described in map features can be used by the teams to update the map of the contest area. Note, that Open Street Map is primarily used for people, and certain rules have to be respected.

Each team can deploy only one robot. Every robot must have EMERGENCY STOP button, which stops its motion. The button must be easily accessible, red and must be a fixed part of the robot (Big Red Switch), so it could be used in case of danger. The minimum size of the switch is defined by an inscribed circle with diameter of 2 cm. The team must show that it is easy to manipulate the robot — two people must be able to carry it several tens of meters. There is also a minimal size — robot has to carry a full 5 l beer barrel.

The robots are expected to stay “on the road” which means to stay on the paved passage ways. If any robot leaves the road, the trial ends. The team has to take care of their robot and remove it immediately.

There could be obstacles on the road. Besides natural obstacles like benches there may also be artificial obstacles. A typical (artificial) obstacle is for example a figurant, a banana paper box or another robot. Robots may not touch the obstacles. Contact with an obstacle means end of the trial. The robot may stop in front of the obstacle and visually or acoustically give notice. Note, that the robot has to detect, that the obstacle is no longer present.

The cases where a faster robot catches up a slower one won't be explicitly handled. The faster robot can handle the slower robot as an obstacle, i.e. avoid it or wait until the „obstacle” disappears. In general the road rules will be respected: right of way, avoidance to the right, passing on the left.

from www.robotika.cz

City of Łódź

Łódź is called the city of four cultures to remind the industrial heritage created by the Polish, German, Jewish, and Russian inhabitants in the XIX century. Although it has been granted a town charter in 1423 from king Władysław Jagiełło only the years 1870–1890 marked the period of most intense industrial development in the city's history. Because of the growth in textile industry, the city has sometimes been called the "Polish Manchester". As a result, Łódź soared from a population of 13 000 in 1840 to over 500 000 in 1913. The beautiful traces of this golden age can be found all around the city.

Poznański Palace (today it holds the Museum of the City of Łódź) – a beautiful XIX century building that was built by Izrael Poznański – a textile magnate and a philanthropist. His factory – the largest XIX century textile production complex has been turned into a shopping centre called "Manufaktura" which is an example of modern business which operates in restored nineteenth century buildings. Piotrkowska Street is the main artery and attraction stretching north to south for a little over five kilometers, making it one of the longest commercial streets in the world. Many of the eclectic buildings have been renovated and date back to the XIX century. Księży Młyn – a large complex of XIX century textile factories, blocks of flats for workers, residences, schools, hospitals, firefighters house, gasworks and clubs. Jewish cemetery – the largest in Europe. Old cemetery with the Karol Scheibler's Chapel – one of the most exquisite sacral buildings in Europe and in the world. Museum of Art was one of the first museums of modern art in Europe and it has a really impressive collection of Polish and international 20th century art. Central Museum of Textiles is located in the White Factory of Geyer vividly lit during the evenings.

Today Łódź has around 740 000 inhabitants. The city benefits from its central location in Poland. Numerous firms have located their logistics centers in the vicinity. Two motorways, A1 spanning from the north to the south of Poland, and A2 going from the east to the west intersect northeast of the city. Recent years have seen many foreign companies opening offices in Łódź. ABB, Bosh, Dell, Gillette, Indesit, Infosys, and Samsung are some examples. Łódź is also a cradle of several innovative companies: Ericpol, Transition Technologies and TME just to name a few. Łódź is a thriving center of academic life with major state-owned universities and academies: University of Łódź, Lodz University of Technology, Medical University of Łódź, National Film School in Łódź, Academy of Music in Łódź, Strzemiński Academy of Art Łódź, and dozens of other schools and academies.

Organizers:

Institute of Automatic Control
automatyka.p.lodz.pl

Lodz University of Technology
Faculty of Electrical, Electronic, Computer and Control Engineering
www.weeia.p.lodz.pl

Contact:

Lodz University of Technology, Institute of Automatic Control
Stefanowskiego 18/22, 90-924 Łódź, Poland
Ph: +4842 631 2554, Fax: +4842 631 2551
Email: i-13@adm.p.lodz.pl

Partners:

Sponsors:

www.rie2013.eu

A Day in the Life of an Educational Robot

A Report and Analysis of a School Working with Educational Robots

Dave Catlin
Valiant Technology Ltd
London, England
dave@valiant-technology.com

***Abstract*—Educational robots are powerful tools with the potential to make regular contributions to the educational life of a school. For robots to fulfill this role they need integrating into everyday teaching practice. This paper is the first in a series that looks at that process in a specific school. It reports a first encounter of pupils with Roamer robots as part of a week long project and highlights some assimilation issues. The Educational Robotic Application (ERA) Principles is used as an evaluation framework and to provide data for the e-Robot project, a long term programme aimed validating the ERA Principles.**

***Keywords*—Teaching with Robots, TWR, Roamer, Educational Robot, ERA Principles, e-Robot, Valiant Technology**

I. INTRODUCTION

This report is the first in a series aimed at discovering what happens to robots when they go to school. What does it take for a school to embrace the potential of the robots to enrich the educational experience of the students on a regular basis? It is part of a wider project that is interested in discovering strategies that will ensure that the technology is successful as well as the nature of potential pitfalls and how these may be avoided.

The report looks at data accumulated by Valiant Technology over the last 30 years. This presents an interesting perspective on the conditions that may deter the long term use of the technology. The plan is to monitor adoption of the technology at a single school, Maple Cross Junior Middle and Infant school in Rickmansworth, England. Therefore, the report presents baseline data about the school gathered from public records supplemented by interviews with various staff members. The school conducted a week long ICT/English/DT[1] project in which they planned to use newly acquired Roamer robots. Observations were gathered through unstructured observation by two non participating onlookers and supplemented by interviews with participating teachers and a review of the comments made by students.

II. THE PROJECT, ERA AND E-ROBOT

The data is analysed using the Educational Robotic Application (ERA) Principles, which provides a framework for evaluating the effectiveness of educational robotics [1]. This is done as part of the e-Robot project, a longitudinal research programme aimed at gathering evidence to verify, modify or refute ERA [2].

According to ERA Pedagogy Principle, the Maple Cross activity is classified as a "Project"[2]. Generally, Projects are unique. How do we extract meaningful data from such one-off events? How can we compare the results gathered with other work? The premise is the ERA Principles summarise the value of educational robots. Any educational robot activity that delivers results that correlate well against these principles:

- Demonstrates the positive educational value of the specific activity - even unique projects like this
- Helps substantiate the principles as a true summary of the educational value of educational robots

The aim of the e-Robot project is to gather data from many different activities using ERA as an analytical tool. The more activities affirm ERA (or modify it) the more we are able to use this tool with confidence.

This process is in its infancy. In fact only one formal presentation of ERA Analysis has ever been published [3]. The efforts here should be perceived as fledgling exploration of gathering and analysing ERA data. Increasing mastery and competence will eventually lead to a substantial growth in the e-Robot database.

III. COMMERCIAL PERSPECTIVE

Valiant Technology have sold Turtle type robots to schools for 30 years. Data from the company's customer support and sales enquiry desk provide an interesting perspective on the fate of several hundred thousand robots. Virtually, 100% of British Infant and Primary schools possess educational robots. An estimated 90% of these own a Classic Roamer, similarly 90% have Beebots and about 10 to 15% have Pip or Pixies. Around 12% purchased Valiant Turtles[3]. It is estimated that about after five years two thirds of these robots spend most of their time in the cupboard and are rarely and sometimes never

[1] ICT - Information Communications Technology and DT - Design Technology.

[2] See Section 4 in B. ERA Analysis below.

[3] BeeBot http://www.kenttrustweb.org.uk/kentict/kentict_ct_bee.cfm
Pixie and Pip http://www.swallow.co.uk/
The Valiant Turtle
http://roamerrobot.tumblr.com/post/23079345849/the-history-of-turtle-robots

used in the classroom. Table 1 shows a number explanations for this state of affairs.

TABLE I. WHY SCHOOLS STOP USING ROBOTS

Reason	*Explanation*
Motivation	The school purchased the product for the wrong reasons. a) Schools purchased the product because they felt it would get them a better Ofsted Report[4], there was no real motivation or strategy to use the robots. b) Schools needing urgent repairs to robots because they "planned the use of the robot next semester." This usually means the school is focussed on the technology not how the robots can help students understand difficult concepts - which is usually occurs throughout the academic year.
Staff Movement	The teacher who motivated the purchase and used the robot moved schools and the other staff had never used it.
Poor Training	The teachers do not really know how to use it beyond a few traditional Turtle type activities.
Fading Intrest	People focussed on the technology and not effective applications, after a while some new "gizmo" captures their attention and they move their focus. This does not mean that the older technology stops being effective, it simply loses its novelty value.

There is a flip side to this. There are schools that have been using the technology on a regular basis for one, two, even three decades. These schools have managed to fully embrace the technology. An ad hoc survey of these schools reveals that schools who use the technology regularly have one or more of the following:

- Teaching staff who who use the technology and have been in post for a number of years
- More than one teacher uses the technology
- An approach to teaching that is energetic and generally compatible with constructionvist teaching methods
- Personal, positive experience of the effect of using the technology with students
- A creative vision on how to use the technology to support the teaching of the curriculum
- A supportive school environment to this general approach

IV. MAPLE CROSS JMI

Maple Cross is a small village of about 2,000 people situated just inside London's orbital Motorway. It is essentially a working class area on the borders of Greater London [4].

Maple Cross Junior Mixed Infant and Nursery School is well below average in size for a school of its type. It has an above-average proportion of girls. A large majority of pupils are of White British origin and the proportion of pupils from minority ethnic groups is average. However, the proportion of pupils who speak English as an additional language is below average. The proportion of pupils known to be eligible for free school meals is above average The proportion of pupils with special educational needs and/or disabilities is above average[5]. Since the last inspection the number of pupils at the school has increased significantly [5].

Tables II, III and IV present the government data relating to the school and its performance.

TABLE II. SCHOOL CHARACTERISTICS

Characteristic		*Maple Cross*	*National*
Enrollment	Boys	69	
	Girls	82	
	Total	151	
% With SEN[a] Statements		7.3	7.9
% Pupils with English as a 2nd Language		10.8	17.5
% Pupils Eligible for FSM[b]		29.5	19.3

a. a SEN - Special Educational Need

b. FSM- Free School Meals

c.

The link between FSM eligibility and underachievement is very strong and data on FSM is easily collected and updated annually [6].

TABLE III. OFSTED REPORTS

Summary Ofsted Reports: Grades - 1 Outstanding, 2 Good, 3 Satisfactory [c] and 4 Unsatisfactory			
Criteria	*2006*	*2009*	*2011*
How well do learners achieve?	3	3	2
The standards reached by learners	3	3	3
How well learners make progress, taking account of any significant variations between groups of learners	3	3	2
How well learners with learning difficulties and disabilities make progress	3	2	2

d. The term satisfactory is a misnomer. It indicates a poor school which requires to take energetic action and will be subject to extra monitoring.

Clearly Maple Cross deals with difficulties typical of impoverished communities. They are making admirable progress. Will the regular use of Roamer help to support and enrich this effort?

At the end of the academic year the deputy head will be promoted to head teacher and the ICT Coordinator who invested in the robot will also leave. Will this trigger the type of issues listed in Table II?

4 Schools in England are regularly inspected by the Office for Standards in Education (Ofsted). Ofsted reports are public domain and make statements on the performance of the school and recommendations for improvement. Poor Ofsted reports lead to parents sending their children to other schools and even dramatic action including the dismissal of staff. Since the use of Educational Robots (programmable toys) have been enshrined in the National Curriculum since 1998 Ofsted would view their absence in a school as a demerit.

5 This seems to contradict the Department of Education Data in Table II.

TABLE IV. NATIONAL TEST RESULTS

KS1 (5 - 7 years old) comparison with all schools				
Quintile	*English*	*Reading*	*Writing*	*Maths*
Highest				
2nd Quintile				
3rd Quintile				
4th Quintile				
Lowest				
KS2 (7 - 11 years old) comparison with all schools				
Quintile	*English*	*Reading*	*Writing*	*Maths*
Highest				
2nd Quintile				
3rd Quintile				
4th Quintile				
Lowest				
KS2 progress comparison with similar schools				
Quintile	*English*	*Reading*	*Writing*	*Maths*
Highest				
2nd Quintile				
3rd Quintile				
4th Quintile				
Lowest				
KS2 progress comparison				
	All Schools		*Similar Schools*	
Quintile	*English*	*Maths*	*English*	*Maths*
Highest				
2nd Quintile				
3rd Quintile				
4th Quintile				
Lowest				

e. Similar schools are those where the students started the Key Stage with the same attainment level

V. ROBOTS IN MAPLE CROSS SCHOOL

The school has 2 Classic Roamers and a set of Beebots when they decided to purchase a set of 5 Roamers. This new purchase inspired ICT Coordinator Nick Flint to initiate a school project.

VI. THE PROJECT

The following was the initial project concept:

This project is part of our joint ICT/Design Technology (DT) week. For the DT element - the rest of the school will be making musical instruments out of junk, but my year 5/6 class thought that sounded 'boring!' So - the compromise we have made is that on Monday they will be making 'musical buildings', and I envisage them using the Roamers to navigate through the buildings.

There will be a link to literacy at some stage in the week. Primarily, I would like them to write out pseudo code for the instructions they give their Roamer. I would then like them to build on this code to write a short story using instructional language, based on their code. I should also add that their theme for the term has been Star Wars, so a nod to the genre would be useful.

Nick Flint

This evolved into the idea of using the Roamer music capability to mimic R2D2 type communication. Various unavoidable non-teaching disruptions took place, which upset the general plan.

The students constructed a "Star Wars" City (Fig 1) out of junk material. This was intended as a backdrop for the Roamer stories.

Figure 1 The Star Wars Cityscape created as part of the project

This seemed as much an art and crafts as a DT Project. The city was impressive and clearly the students had explored a range of materials, textures, fastening methods, shapes and forms.

The students completed a various writing assignment, which the teacher described as fabulous. They had to compose a persuasive piece of writing for Vader & Son Estate Agents. It was written in the style of housing particulars. Some clever, humorous examples of work appeared on the school blog.

A. Using the Robot

Because of the disruptions, they were not able to use the robots until the last day. The day started with students sharing their story ideas with members of their group, then deciding what plot to develop. In the end, there were two groups with stories involving two robot characters and a single group with a story based on a Roamer and a human character. This process took about 30 minutes.

They then started to learn how to program the robots. This process consisted of them working in small groups listening to instructions from the teacher, experimenting and listening to the Roamer's help files which told them how to correct their programming errors (Fig 2).

Figure 2 One of the groups engaged in experimenting with Roamer Programming.

The students then developed their stories writing out a narrative using words like '*went forward to the bar, turned to meet each other, beeped at each other before…'* The pupils then programmed the enactment of their stories, tested and debugged the programs.

The Star Wars City had been moved from the classroom to the school hall. Each team took turns to go on set and film their story. Meanwhile the other groups stayed in the classroom and carried on experimenting with the programming capabilities of the Roamer.

The 3 stories produced and enacted by the students were:

- **We Don't Serve Your Kind Here**: A version of a scene from Star Wars where the robots are refused entry into a bar
- **Good Versus Evil**: the robot meeting with Darth Vader
- **Eastenders**: A romance story based on the BBC TV Soap Opera

VII. ERA Analysis

In this section each ERA Principle will stated and then collated to the observations of the Star Wars Project

1) Intelligence: *Educational Robots can have a range of intelligent behaviours that enables them to effectively participate in educational activities.*

The students used a Standard Primary Roamer. This gave them the capability of working with movement, wait, speed, music and volume. Even though they were beginners who had had only 30 minutes of experience with the robot, they used all of these features. They did not use the repeat and procedure functions and could not use the robots extended speech capability or its control features (inputs and outputs).

Bransford et. al. claimed that students' ability to benefit educationally from programming was restricted by their knowledge of the programming language [7]. Clearly, the basic programming capacity of the robot was easily learnt. The resulting dramas were limited, but this was more a matter of the time spent on the task, than the potential of the robots to provide the students with valuable experiences.

2) Interaction: *Students are active learners whose multimodal interactions with educational robots take place via a variety of appropriate semiotic systems.*

The traditional Turtle type robot semiotic systems were in play in this activity: the symbols used to programme the Roamer and visual response of the students to the robots movement in space. This clearly connected students to various mathematical ideas (see the Curriculum and Assessment Principle).

The semiotic of the Star Wars story was also present in the activity. Two of the three groups responded to this. The third did not. (see Conclusions for further discussion of this point).

3) Embodiment: *Students learn by intentional and meaningful interactions with educational robots situated in the same space and time.*

This is always the critical question with robot activities - why a physical robot - could we do this with a virtual robot?

As reported in the initial ERA Paper [1], all the adults involved in this project believed that the robot was offering a unique experience that would not be the same on screen. Moreover, they all felt that this experience was valuable. However, they all found it difficult to provide a concrete explanation of why they found this so.

Body syntonic activity was observed on several occasions [8]. This was spontaneous. Students got up and walked through the robots movements, using their bodies to solve the problems.

Another aspect of embodiment was observed, which has been noticed before, but not recognised as an aspect of embodiment deriving from work with physical robots. This is the animation of the students as the robot acts out their instructions. Their physical response was often lively and indicates a connection to the outcome (Fig 3).

An interesting observation is student reaction to mistakes. It almost universally, humorous. It is not their mistake. The blame is projected onto the robot. It got it wrong. Even when the mistake is revealed publically, the atmosphere is more like a TV Outtake Programme; the audience share the joviality and spare the student from what is normally a humiliation.

Papert [9] claimed the debugging process was a positive way of discovering mistakes. Indeed it forms an iterative process of "getting the right result". However, what has never been noted is the student reaction to success. In all these activities, students demonstrated their success. In particular, the Eastender's team danced in celebration. This is another indication of their physical connection to the challenges and their outcome.

Figure 3 Student animation as the robot acts out its program indicates embodiment in action.

4) Pedagogy: The science of learning underpins a wide range of methods available for using with appropriately designed educational robots to create effective learning scenarios.

This principle is about understanding the structure of the activity. Twenty-eight different methods have been identified as ways of using the robot. Most activities involve more than one. Table V describes the methods used in this project.

TABLE V. MAIN PEDAGOGICAL ELEMENTS OF THE PROJECT

	Pedagogy	Description
1	Project	Projects are spread over several lessons. The robot cotributes to the project outcome. This can be a major or a minor role.
2	Cooperation	The way the robot day was set up required students first in small and then larger groups.
3	Creative	The project involved creative opportunities in the creation of the sets and in writing opportunities and the plotting of the drama.
4	Experimentation	The students experimented in learning how to program the Roamer.
5	Group Task	It was a group based activity.
6	Presentation	The outcome of the project was a video presentation of their story.

5) Curriculum and Assessment*: Educational Robots can facilitate teaching, learning and assessment in traditional curriculum areas by supporting good teaching practice.*

The project as stated in Section V stated the curriculum areas of interest were:

- Design Technology
- Information and Communication Technologies
- Writing
- Music

Specific objectives were not stated, except the notion of using pseudo code in the writing of the robot dramas. The outcome of design technology work was impressive, although a little more Art than Design Technology. This is not untypical of Primary school approaches to that subject.

However, the work did not involve resistant materials or tool skills associated with these materials. This is not untypical of Primary schools in general.

The ICT effort was dominated by learning to program the Roamer. Some high quality writing had been done earlier in the week, but the standard dropped when it came to using the Roamer. This was understandable given the general circumstances.

The discovery of the robot music facility and the volume feature did attract experimentation with musical notes as the students tried to make the robots sound like R2D2.

There was little formal assessment done, except, of course, the student movies. Though as an example of writing they were simple, as an expression of ICT they represented good work.

There was a plenary session and the students universally cited the development of their programming skills as their main achievement during the day. Clearly their curiosity was provoked because they eagerly asked questions about other keyboard functions that they had not had tome to explore.

6) Engagement *Through engagement Educational Robots can foster affirmative emotional states and social relationships that promote the creation of positive learning attitudes and environments, which improves the quality and depth of a student's learning experience.*

In the preliminary work students worked well and stayed on task. However, when the robots came out, engagement, enthusiasm and energy noticeably increased (Fig 4).

Figure 4 All eyes on the Roamer. The concentration dramatically increased when the robots were used.

The teams took it in turns to do the filming. When they had completed the task they returned to the classroom and voluntarily continued to explore and play with the robots.

Interestingly, in the plenary session several of them enquired whether they could buy them and take them home. This indicated an aspect of engagement - enjoyment.

The Roamer day ended the school closed for the two week Easter holidays. On returning from this break students were set a writing assignment about the project. It is clear from the student work that the enthusiastic element of engagement was still evident [10]. One student, who was not there on the day, but still had to do the writing assignment reported:

Everyone says that they had loads of fun and that everyone was drawn in from what the people said. From the sounds of it I think everyone had a really good time. I wish I could have been there.

This was the "word in the playground"; inhibited by the desire to provide the answer the teacher wants to hear.

Nick Flint reported:

Roamer provided motivation and a focus. A couple of students went on the school blog during the Easter break and wrote about the session. This is the first time they have ever done anything like that, and to do it in the holiday was even more remarkable.

This certainly supports the engagement claim that work with educational robots improve the attitude of learners about learning.

7) Personalisation: *Educational robots personalise the learning experience to suit the individual needs of students across a range of subjects.*

The class was reasonably homogenous and special changes to the robot, or the activity were not necessary. Consequently, the Personalisation Principle was not featured.

8) Equity: *Educational robots support principles of equity of age, gender, ability, race, ethnicity, culture, social class, life style and political status.*

The class of 20 consisted of 15 girls and 5 boys. They worked together well. The ICT Coordinator/class teacher, pointed out that the group who did the romance (Eastenders) story line were all girls. It is an aspect of equity that the robots are tools which allow students to express their interests and concerns. It was also noted that particularly engaged in the tasks was a pupil from a traveler family/background.

9) Sustainable Learning: *Educational Robots can enhance learning in the longer term through the development of meta-cognition, life skills and learner self-knowledge.*

Sustainable Learning skills are also referred to Life Long Learning skills. The essence of this idea is engagement in this activity offers the student the opportunity practice and develop skills that can be applied to radically different situations.

The activity engaged students in practice of sustainable learning skills. Table VI lists the skills identified by ERA and highlights those involved in this particular activity.

The highlighted skills were observed. This does not mean that the other skills were not involved or being developed. Nor does it mean that all students were developing the skills in the same way or to the same level. In fact an incident occurred where a student did lose emotional control. This is a recognised problem with this particular student and was dealt with according to school policy. Robot develop the sustainable skills by putting students into situations where they have the opportunity to practice those skills. We all fall over when we learn to walk. But generally, this is done in a safe environment. There is always the chance that problems will occur in the dynamic situations Roamer activities create. While it safe environment, it also is an area deserving of more detailed research and development to find positive ways of dealing with such breakdowns.

10) Practical: *Educational robots must meet the practical issues involved in organising and delivering education in both formal and informal learning situations.*

The way the activity was conducted and the circumstances that prevailed infringed this important Principle. This does not reflect the professionalism of the teaching staff, nor the particular activity.

It was a general view of the teaching staff and the observers that too much was being attempted in such a short time period. The day was a success, but it was possible to improve the quality of the planning and execution in a way that would allow the students to get more out of the activity.

TABLE VI. SUSTAINABLE SKILLS ANALYSIS

Category	Sub Category	Sustainable Skill
Cognitive	Managing	Research (Investigation)
		Resource Use
		Planning/Organisation
		Goal Setting
	Thinking	Types of thinking
		Metacognition
		Critical Thinking
		Problem Solving
		Decision Making
		Learning to Learn
		Creativity
Emotional	Relating	Accepting Difference
		Conflict Management
		Social Skills
		Communications
		Cooperation
	Caring	Concern for Others
		Sharing
		Empathy
		Supportive Attitudes
Social	Giving	Helpfulness
		Responsibility
		Leadership
		Group Contributions
	Working	Debate
		Presentation
		Teamwork
		Self Motivation
Personal		Self Esteem
		Accountability
		Character
		Confidence
		Emotional Control
		Self Discipline
		Concentration
		Memory
		Observation Skills
		Motivation
		Determination
		Reslience

The prominence of the robot overpowered the project objectives. That is, the notion of the Star Wars theme faded into the background. In doing this, the potency of the overall project to deliver the original learning objectives and the contribution Roamer made to it was changed. This was acceptable and is always a possibility in constructionist projects. However, in general:

- Big projects like this should not be used to teach students how to program the robot, they should know before hand - this allows them to concentrate on the other curriculum issues - in this case writing the dramas
- While the school understood and practiced Assessment for Learning (AfL), preparation for using these techniques was not sufficient. Consequently learning intentions and success criteria were not clearly established [11].
- The robots should have been used on some smaller activities earlier in the week

These are not criticisms but practical lessons to be learnt.

VIII. E-ROBOT

When the No Child Left Behind Act (NCLB) was launched in the USA it made a statement that only "scientifically based educational research" was of value [12]. This essentially values positivistic research and dismisses qualitative methods. Catlin and Blamires [2] refute the stringency of the NCLB approach, but accept that the research basis supporting educational robotics has its problems. For example, if we focused on this specific activity as a case study, how do we relate its findings to similar or even radically different educational robotic activities?

The ERA Principles provide the framework for evaluating an educational activity or method. It is flexible in that it can accept data from observational studies like this, or large scale quantitative methods described under the positivistic philosophy.

Even though the positivistic methods often claim to be longitudinal studies, in real educational terms they are not. The e-Robot project aims to gather data from the educational robotic community through online social networking on an ongoing basis. Because it is community based it has no time limit. A recent paper published by the UK Government takes a similar approach to the NCLB approach [13]. Like the NCLB approach this paper is recommending the medical model of research as its guide. It is an indication of trend that will become prevalent in education. e-Robot aims to gather data in response to this future demand.

This report provides a form of qualitative analysis against the ERA Framework. This is an interesting instantiation, in ways typical of educational robotic research. It is clear that so much more could be done to gather better data, structure the analysis and synthesise the data with results from other projects.

CONCLUSIONS

This project gathered positive evidence confirming some of the ideas of ERA. It showed the desperate need to develop research tools and metadata structures that can better shape the design of qualitative research, its analysis and synthesis with data from other projects. This is subject to ongoing development.

ACKNOWLEDGMENT

Thanks to the staff and students at Maple Cross School, particularly teacher Nick Flint and Deputy Head . Thanks also goes to photographer Alan Coode and to Kate Hudson of Valiant Technology. Thanks also to Mike Blamires of Canterbury Christ Church University for his proof reading and advice.

REFERENCES

[1] Catlin, D. and Blamires, M. (2010). The Principles of Educational Robotic Applications (ERA): A framework for understanding and developing educational robots and their activities. Proceedings of Constructionism 2010, Paris.

[2] Catlin, D. and Blamires, M. (2010). The e-Robot Project: A Longitudinal On-Line Research Collaboration to Investigate ERA Principles. TRTWR 2010 Conference, Darmstadt, Germany, part of SIMPAR 2010.

[3] Catlin, D. Smith, J.L. and Morrison, K. (2012). Using Educational Robots as Tools of Cultural Expression: A Report on a Project with Indigenous Communities. Proceedings of the 3rd International Conference on Robotics in Education.

[4] Anon, Background information on Maple Cross http://en.wikipedia.org/wiki/Maple_Cross

[5] Cohen, R. (2011) Maple Cross Junior Mixed Infant and Nursery School Inspection Report Number 379383 Dated 15th to 16th November 2011, Ofsted, Accessed 10.04.13 http://www.ofsted.gov.uk/provider/files/1985835/urn/117266.pdf

[6] Department for Education: Free School Meals FAQs. - But isn't FSM an inaccurate measure of disadvantage? Accessed 10.04.13 http://www.education.gov.uk/schools/pupilsupport/premium/ppfaqs/a0076064/free-school-meals-faqs#faq7

[7] Bransford, J.D., Brown, A.L., Cocking, R.R. (Editors) (2000) How People Learn: Brain, Mind, Experience and School, p213 to 217. National Academy Press.

[8] Papert, S. (1980a) *Mindstorms: Children, Computers and Powerful Ideas,* (p63, 68 and 205) Basic Books.

[9] Ibid. p23.

[10] Students of Maple Cross School (2013) Star Wars in Rickmansworth, GO Magazine, Published by Valiant Technology May 10th 2003 http://go.valiant-roamer.com/2013/05/10/star-wars-in-rickmansworth/ accessed 1st June 2013.

[11] Catlin, D. (2012) Maximising the Effectiveness of Educational Robotics through the Use of Assessment for Learning Methodologies. Proceedings 3rd International Workshop of Teaching Robotics, Teaching with Robotics Conference 2012, Riva del Garda, Italy

[12] Reyna, V: What is Scientifically Based Evidence. Presentation given at The Use of Scientifically Based Research in Education: Working Group Conference Department of Education, Washington, D.C (February 6, 2002) http://www2.ed.gov/nclb/methods/whatworks/research/transcript.pdf

[13] Goldacre, B. (2013) Building Evidence into Education. Accessed 01.07.13 http://media.education.gov.uk/assets/files/pdf/b/ben%20goldacre%20paper.pdf

Robotics when I want my two front teeth:

Issues around the core curriculum of primary school.

Mícheál Ó Dúill (Mike Doyle)
Logios.Org (& Euro Lyceum, Sofia, Bulgaria)
logios.org@googlemail.com

***Abstract*– The teaching of robotics is viewed as an introduction to that particular branch of engineering. Primary school children do not see it as such. For them, robots are products of the fantasy industry. So, the question arises: What is the place of robotics in primary school and how is it best introduced? Building on earlier contributions, three streams of thought are brought together to offer an answer. The first stream is a refinement of the technicity proposition, using a new object to think with and modified model of the adaptation. This leads to a view of the nature of primary education and its relationship with the tertiary phase. The core of primary education, conceptual and pedagogic, is conceived as an undifferentiated whole, innocent of subjects. The foundation of all to follow, it is largely focussed on the skill set required to master pencil and paper media. The second stream is educational robotics. This entails different skills: the capability to construct in three dimensions and to use the computer as the base medium. This takes robotics out of the realm of traditional teaching and into Turing teaching. Two approaches to robotics in primary education are considered. The short course kit-based method is contrasted with the Ilieva curriculum. The third stream is a little more subterranean. Observations from Ilieve-style teaching, not possible with the kit-method, lead to questions about traditional primary teaching method, notably in mathematics. These turn raise further interesting questions about traditional teaching and quite what children are learning. A question about robotics for six year olds has catalysed more fundamental lines of inquiry.**

***Keywords*– Robotics, primary school, curriculum, technicity, learning medium, information, teaching methodology, practice**

I. INTRODUCTION

Ó Dúill [1] claimed that robotics in school had regressed over the past thirty years. In that paper he also sought to make a distinction between the classroom use of robots by children and the discipline that is robotics. The former [2, 3] engaged a childôs social brain more than their emerging scientific one. Informing his analysis was a development of the proposition, previously reported [4], that the human species has a unique evolutionary adaptation, technicity, which confers the capacity for technology and scientific thought. The distinction between naïve and scientific thinking [5] was formalised as between environmentally based complex perceptual-verbal v-concepts and technicity-based low-entropy t-concepts. The theoretical discussion included a description of Turing teaching, teaching that took the computer rather than pencil and paper as the base medium of education [6]. He used this perspective to question the efficacy of traditional teaching method in mathematics and literacy in primary school.

The objective of this paper was to provide, firstly, a more developed view on technicity; secondly, in the light of a shift in emphasis to pedagogy [7] from research interventions [8], a comparison of a short-course kit-based introduction with the whole school curriculum approach developed by Ilieva [9]. The former will focus on work associated with MIT and Tufts University and the products of the LEGO Company. Study of the latter has been facilitated by Doyle, who introduced her method in the first three grades of a primary school this year. This exercise was valuable in that the children were all at the normal levels for their age in school subjects but had little experience of LEGO-based teaching (as opposed to playing with LEGO sets at home). In both school settings observations were made that call into question traditional teaching method in mathematics, and which raise issues for further research.

The author felt that clarity on the nature of primary school was necessary, there being tension between an academic focus on subjects and the developmental viewpoint. This is nicely illustrated by recent shifts in emphasis in the English national curriculum [10, 11] and learned society intervention therein [12]. Invisible in these debates is the nature of the medium of traditional instruction. Alexander emphasised oral methods [13] but, as revealed by the illustrations in the (English) National Curriculum Primary Handbook [10], school-children inhabit a two dimensional world ï a paper and pencil flatland. Notable in the context of this paper is the finessing of the primary mathematics curriculum by Alexander [13]. As an aid to thinking, a visual metaphor for education is introduced.

II TECHNICITY AND TEACHING

Over the past decade, Ó Dúill has developed the idea that the human has an evolved adaptation, technicity, which gives the species a technological capability. The view from a window, figure 1, contains both natural and technological elements.

Figure 1. View from a window.

The technological, including the poor quality graffiti, are more simple in form and content that the natural. Therefore they are of lower entropy. The second law of thermodynamics requires that there be a source of low entropy information from which these entities are conceived. This source can only be internal to the brain.

All brains incorporate information on the environment from which the organism constructs a representation of the world. This representation is not a true picture of reality; it is a gene-determined and of great evolutionary depth. Colour vision, for example, evolved in fishes some 500 million years ago. In the human, this information is expressed in a thin layer of cortex, figure 2. There are two classes of information: that about the organism itself; and that from which representations of the external world are constructed. This latter, first explored by Hubel [14] and colleagues and therefore referred to as the Hubel zone, is the repository of elemental information from which perceptions are constructed.

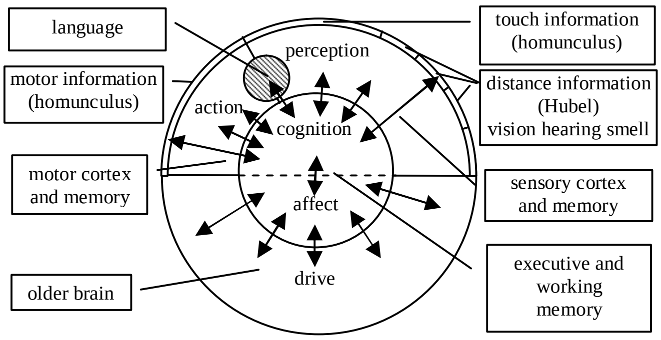


Figure 2. Schematic of the information relationships in the human brain. Note that language is shown as a module within the brain but outside the executive and working memory.

In normal mental operation in mammals and primates, Hubel zone information is unavailable to cognition. The technicity proposition states that in the human, and only in the human, the ongoing process of prefrontal invasion of other brain areas reached the Hubel zone and made low-entropy information available to cognition. It is also proposed that the sensory and motor homunculi were similarly accessed. No specific neural connection is proposed, merely that the information became available to executive cognitive and affective processes. This proposition is consistent with principles of brain evolution and prefrontal function [15, 16] and hominine brain expansion.

Hubel zone information includes the length and angle of lines, direction of motion and a colour space defined by the opponent pairs: yellow/blue, red/green, and black/white. Here is the ‘false colour’ of vision, which includes purple (reddish blue) for which no spectral photon exists. The colours in the scene of figure 1 are a mental construct that the human shares with the fishes of the Coral Sea and all other vertebrates that retain colour vision.

The consequences of this evolutionary adaptation may be illustrated by the game of snooker, figure 3. That such games exist at all is the result of what is known as ‘social brain’ expansion [17]: increases in information processing capacity that, lubricated by language, enabled reciprocal altruism to become an evolutionarily stable like-style. The complexity of thought, shown by depth of linguistic intentionality and theory of mind that resulted is sufficient to devise and play this game of skill and strategy. Technicity is the source of its physical substrate. Spherical balls, flat rectangular table and truncated cone cue are all constructed from Hubel information. Colours include the reddish white (pink) and blackish red (brown); constructs of colour vision. Game-play is based on Hubel motion information, for player and spectator. Play additionally uses homunculus information to construct complex motions for cue and object ball in order to position them for future play. Snooker pits two social minds against another in an aesthetic and low-entropy environment made possible by the technicity adaptation, to the delight of audiences.

Figure 3. The snooker table and the mind of the player (after DmitryYakunin).

Language is a function of perception, and therefore unreliable. In figure 1 the diamond shape motif is actually square, which may be seen by rotating the page. Visualisation is cognitively more powerful than verbalisation. Papert was right [18]. But we also know this from the history of science. We also know it from child development and school. Children pick up a pencil and draw after language is well developed. The focus of primary school is on visualisations: reading and writing of language and number supersede oral methods; and drawings illustrate everything. The primary school phase is the critical period when the technicity adaptation comes on stream and cognitively powerful neural connections are created.

A. The Child and the School

Language is an evolutionary adaptation of great depth. It is intimately tied to our false-but-adaptive perceptions. Children arrive in school with this adaptation well developed. It is the cognitive tool they bring to education. Technicity is only just beginning to develop. The problem for educationalists is that technicity has no direct access to language. For a technicity concept to access language it first needs to be a perception. So infants in the crib are offered shapes like cubes and cylinders in primary colours to handle and post through holes. Linkage to language is a feature of primary education that has led some academics [13] to mistake it for the purpose of this phase.

Discourses on school curricula tend to be subject oriented. Influenced by fractured academe, the primary curriculum in England was so for many years, with a recent (but soon to be reversed) switch to collective themes. Superimposed on this has been the enduring 3Rs: skill in literacy and numeracy. The model has been the secondary and tertiary curricular divisions. Thus, demands are made for, e.g. computing or robotics to be part of primary education. That this is misconceived may be illustrated with a visualisation, figure 4.

Figure 4. Education as a tree with an annual life-cycle (after lôOccitane).

School life, like that of a tree, has an annual cycle. Children begin in kindergarten and progress through the system until they leap off into the world of work. How to conceptualise each phase is the question.

Before we consider the tree, we had better look at is source of sustenance. There is the earth of the culture within which it is embedded. For the young child, the physical culture is far more immediate than for the older. No new technology exists for them, all is extant. It is only as children grow up that the connotation of technology with artefacts not found in a scrap-yard or museum develops. This cultural compost changes year on year, as leaves fall. The sun, air and rain may be likened to the human culture that surrounds the growing child. Formal education, school, may be likened to the fertilisers and other treatment that a gardener undertakes.

Taking the tree as a representation of school, the primary phase may be likened to the trunk. Secondary school is the larger limbs, leading to the smaller branches: the multiplicity of disciplines found at tertiary level. Robotics is one of these.

The largely undifferentiated primary school trunk provides the cognitive and affective support for the later phases. At a neurological level, by the end of primary school affective prefrontal (executive) connections are fully developed and cognitive ones largely mature. Here the foundations are laid. Subject divisions are blurred within topic teaching: indeed the divisions have little meaning. Notwithstanding that boys, in particular, are culturally aware of (fantasy) robots, robotics has no reference. Inserting robotics is analogous to grafting an outer branch onto the tree trunk: isolated, it is unlikely to take.

B. Flatland

Subject-level descriptions of primary schooling finesse the medium of instruction. Literacy and numeracy are categorised as language and mathematics. However, when the activity of children is examined, they are seen to be mastering the skills of two-dimensional representation in a frozen flatland. The medium of instruction and representation is pencil and paper. Success is measured in terms of capability to decode, encode and animate marks on paper. In traditional teaching method, pencil and paper based, the word and world is mapped to two dimensions.

Robotics neither fits this pencil and paper mapping nor the cereal-packet, toilet-roll tube and glue construction that passes for technology in traditional primary schooling. It is seriously three-dimensional and computational. It does not fit traditional teaching method; it makes different cognitive demands. Some of these, it will be seen, throw into question the efficacy of traditional teaching.

IV. Robotics in the Classroom

The isolation of robotics from tradition means that there is no curriculum for robotics, or any of its component parts. It is fortunate, however, that one product, little over fifty years old, has proved popular both in primary education and at home. LEGO has been associated with construction and robotics for thirty years, so comparison between approaches is possible.

The LEGO Company approach is modelled on its toy system: a kit-based short-course methodology to introduce robotics. For primary school, this is characterised by WeDo.

The long-term presence of LEGO in the classroom has made feasible curriculum development for all primary grades. The work of Ilieva [18] in developing a curriculum attained national acceptance in Bulgaria. This is now being applied by Doyle.

It is important to recognise that the objectives of the two approaches differ. WeDo is a short course aimed at older primary school children. The Ilieva curriculum extends over all four years of primary school. In her case the issue is not how to introduce robotics but what contribution ‛roboticsô might make to childrenôs development in the trunk of the learning tree.

B. Flatpackery

WeDo is a self-contained kit for a twelve-week course that enables children to construct and control a variety of models.

A limited range of LEGO components makes possible twelve different models. This assortment includes a range of bricks, gears, axles, a motor, and distance and tilt sensors. A USB hub connects to a computer. The computer is used to display both the construction instructions, which are the classic LEGO pictograms, and the software, also pictographic (figure 5).

Figure 5. The LEGO WeDo programming environment.

It is obvious that this is consistent with the LEGO good-play ethos. Children have all the elements and all the help they need successfully to construct a model and make it work. The models all have a child-friendly, out of the toy box, appeal. This is a self-contained experiential package that introduces some aspects of robotics. In the context of traditional primary school teaching, it has the advantage that the teacher needs little or no understanding of construction or computing.

The analogy with self-assembly furniture in the sub-title of this section is not inappropriate. For each individual model comprehensive LEGO-style visual step-by-step instructions are provided. The programming environment is similarly visual: programs are constructed by linking pictograms of the controllable elements to represent a sequence of actions. For each model a program is provided for children to copy. For each model links to the STEM agenda are suggested. For each model an invitation to children to ñmake their ownò model or program is extended.

B. The Ilieva approach

As a part of the first Bulgarian national curriculum directed at teaching children mastery of the computer as a medium, Ilieva in 1999 devised a module for the control of external devices. Taught for over fifteen years [8], this method has recently been used by Doyle in the context of design and technology in a school working to the English national curriculum [9, 10]. It has thus been possible to compare observations of childrenõs learning in two different environments. Both teachers have also used WeDo. In introducing the Ilieva approach, Doyle was in the fairly unique position of starting from scratch with the three first grades. Thus, childrenõs LEGO capability and learning were capable of being compared with their level of attainment in the core subjects, notably mathematics.

For completeness it is useful to revisit Ilievaõs curriculum grid (table 1) and to outline her method. The material base is LEGO, with a good range of bricks to construct situations that could incorporate a number of models. Control Lab elements: lamps, motors, and sensors (touch, light, temperature, and rotation) and Logo programming providing for robotics.

TABLE 3
FACSIMILE OF ILIEVAÕS CURRICULUM GRID FOR ICT AND LEGO

	ICT	LEGO	TEAMWORK
1^{st} grade	Introducing the computer. Learning to use the mouse. Working with graphics and sound. Using ToolKid, specially designed software	Getting to know the construction material. Make the first simple models. Using LEGO bricks	Individual work to a teacher example. Individual work on their own idea. First steps to learn to work in a team of two. Outcome: individual projects
2^{nd} grade	Learning to use the keyboard. Working with graphics, text, animation and sound and combinations of these. Using ToolKid.	Make a variety of more difficult constructions. Make more realistic models with many details. Learn to recreate a first simple situation. Introducing controllable models and programming.	Working individually or in twos or threes on one common theme discussed with and agreed by all children. Every construction part of the common project. No isolated models allowed. Outcome: class project.
3^{rd} grade	Individual and class projects that combine different types of information. Using ToolKid, MS Word and Paint and the Internet. Product: movies, stories, comics, slide-show.	First robotics projects using sensors. Programming with procedures. More complex situations.	Freedom to choose and change team membership in the context of each new class project. Learn to cooperate with the work of younger children on school projects.
4^{th} grade	Use of the Internet, Paint, Word and PowerPoint to make individual and class projects based on the curriculum for other school subjects.	Larger and more complex projects. Programming with super-procedures and conditions.	Coordinate the work of all classes on whole school projects. Organise a presentation, introducing the work of all children, to the school and parents.

A continuous curriculum through all four grades of primary school, with a double lesson each week for the whole school year, enables children understanding the material. This gives them the capability creatively to construct and creatively to write programs to control the behaviour of their creations. The emphasis on teamwork provides for purposeful discourse. The situations that these creations contribute to are rooted in reality. They include villages with streetlights, a crossing from home to school and shops with traffic lights, wind and water mills in a landscape, a Christmas town and a summer fun-fair (luna park) with swings, roundabouts, rides and side-shows.

When children are first learning, example models are made by the children to teacher demonstration with accompanying verbal commentary. When errors are made in construction, the model is dismantled and the child helped to make corrections. Visual construction and spoken description are combined at all stages. This is particularly so for dimensions: the length and width, in buttons, of a brick or plate; and for the aesthetic aspects: choice of colour and attractiveness of form. A teacher-built model is often available for inspection, providing a visual deconstruction approach to construction.

Programming uses the venerable computer language Logo. Text-based, it links directly to literacy. They are not restricted to the controllable elements. It is possible to relate the words used in Logo procedures to the language that the children use to describe the behaviour of their models. Children may write a procedure that incorporates their own name, like writing their name on their class-work. As with the LEGO elements, the teacher has the possibility of providing example, or more complex programs, as a part of the teaching process.

Figure 6 illustrates the situation method: a house with a burglar alarm and police in action.

Figure 6. Police Action

In the past school-year opportunity to apply Ilievaôs method in a new setting arose. LEGO was to be the foundation of design and technology teaching in a school that followed the English national curriculum. The LEGO resource was identical to that used by Ilieva, including the Control Lab elements. None of the children having been introduced to LEGO, except as a toy at home, the three lower grades all started at the same level of experience. They differed only in their prior learning in other subjects, notably literacy and mathematics. The language of instruction was English, a second language for most children in the school. Thus, a similar approach was in use by Ilieva in her school and Doyle in his, and it was possible to compare notes. Both are teachers who work with the computer as their base medium and have expertise in LEGO construction and primary minds.

A good selection of LEGO elements as a teaching resource meant that neither teacher was constrained by the limitations imposed by a kit like WeDo. Not only could LEGO learning begin from understanding the material, removing the need for step-by-step instructions, but the teachers were able to create constructions that complemented the situation built by the children. This enabled the children to observe more advanced possibilities of the medium (figure 7).

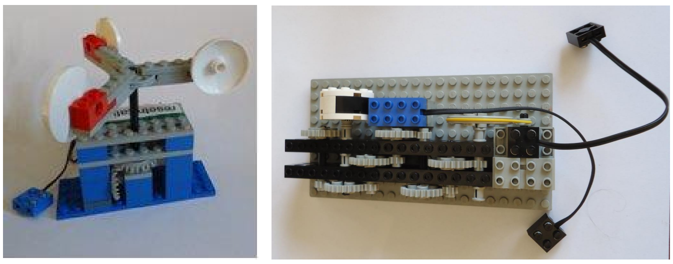


Figure 7. Teacher constructions to enhance situations.

To enhance the Windmills an anemometer using the rotation sensor was built. This was programmed to set the power of the windmill motors and to display a bar-chart of ówind-speedô ï how hard the children could blow! As with real wind turbines, too hard a puff would shut them off. The second is a day/night clock used with the Streetlights. This used the light sensor and gears to slow the motion.

V. Two issues

The Ilieva approach, by building robotics into the curriculum, brought to light two issues that could not have been observed in the kit-based short course. The first illustrates the pervasive nature of naïve thought, the second raises interesting questions about traditional mathematics teaching and learning.

A. Reverse direction

Pictographic step-by-step instructions make use of the power of technicity to simplify the construction process, be it of the model or program. But, the core objective of primary school is not ease of construction. It is literacy, numeracy and science. It is clear that the pictographic construction instructions and programming do not contribute to the development of the first two, but what of science? Does this approach to construction develop scientific as opposed to naïve thinking?

Consider motors. WeDo (NI LabVIEW) has pictograms for ómotor-this-wayô and ómotor-that-wayô with a curly arrow illustrating the direction of rotation (figure 5). At primary school it is usual to talk about clockwise and anticlockwise rotation; but this is a detail. The subterranean question is: Do we want children to work at a perceptual level or to help them to understand the underlying but invisible science?

From a scientific viewpoint, direction of rotation of a DC motor is changed by switching the polarity of the power supply. The science curriculum for primary school includes simple electric circuits. Physically reversing the connections to a battery is one way of changing the direction of rotation of a motor. For many years LEGO have produced switches to reversal polarity; currently used with Power Functions toys.

They also produced a lamp that was sensitive to polarity (#4767). Though never included in an educational package; when used in teaching, this component immediately opens an opportunity for talking about direction of current.

Notably, WeDo and Control Lab can use polarity-sensitive LED lamps from the Power Functions range. However, these are wired so as not to be polarity sensitive: a problem avoided in making a toy work; an opportunity lost in education.

The previous programming environment, text-based Logo, seemed similarly inflicted, with an órdô command that the documentation described in terms of motor direction. But the linguistic órdô command may also connote reversal of current direction: a property of language not open to pictography. It appears that the perceptual pictographic approach, unmodified by language, may lead to naïve rather than scientific thinking. This runs counter to the science curriculum, which seeks to help children understand the physical foundation of the world: replacing naïve v-concepts by scientific t-concepts.

B. The 5-brick

An ongoing issue with the children of all grades was sound construction and choice of brick. Bricks are stored in bins according to size. LEGO bricks come in sizes of 1, 2, 3 & 4, after which they increment by 2. The exercise described below (figures 8, 9 & 10), undertaken at the end of April, illustrates the difficulties third grade children encountered when working in three dimensions.

The objective was to assess the childrenôs capacity to apply the understanding they had developed since the beginning of the school year. All the processes had been used on a number of occasions and principles of sound construction, including brick-bonding and use of the roof plate as a template for the base area emphasised.

A similar exercise was undertaken with the second grade; and later with fourth and sixth grades.

Can you build a house with a door and window?
This house has no back!

Choose a partner to work with.

Collect all these bricks before you begin to build:

Number	Description	Dimension	Comment
1	Building plate	16 x 32	To build on
1	Plate	6 x 16	For the roof
1	Window with shutters	4 x 3	
1	Door	4 x 5	
4	Brick	8 x 1	Try to choose bricks of the same colour or that go together well
8	Brick	6 x 1	
3	Brick	4 x 1	
5	Brick	3 x 1	
7	Brick	2 x 1	
8	Slope	4 x 3	Roof
3	Slope	4 x 2	Roof
1	Slope	2 x 2	Roof
1	Brick	2 x 2	Chimney
2	Round brick	1 x 1	Chimney

The house looks like this:

The brick colours are to help you find the correct bricks.

1 .You have exactly the bricks you need.
2. Use the roof plate to help you with the first bricks.
3. Look carefully at the picture to see which bricks go where.
4. Look at the built house if you are not sure.
5. Remember ï we have no 5 bricks!
8. Ask me if you are not sure.

Figure 8. Worksheet for backless house exercise.

The mathematical understanding required was well within the range expected of children in these grades [9]. Their oral and pencil and paper exercises had encompassed all the number relations required. They had used physical structural material. Geometry had progressed into angle and area. And they knew their multiplication tables. No child from the first grade would agree that two and two make five. Their capacity to apply this knowledge to a construction was, however, very limited. The beginning of a house construction is shown in figure 9.

No group used the roof plate as a template for the base area of the house, though some began to build on it.

It is clear that a ófiveô brick is required to complete the end wall. The bricks provided included twos and threes. The only group that solved this problem without help was the one that examined the ready-built house very carefully, i.e. used visual decomposition. Attempted solutions included a six brick that stuck out at the back and two twos, leaving a gap.

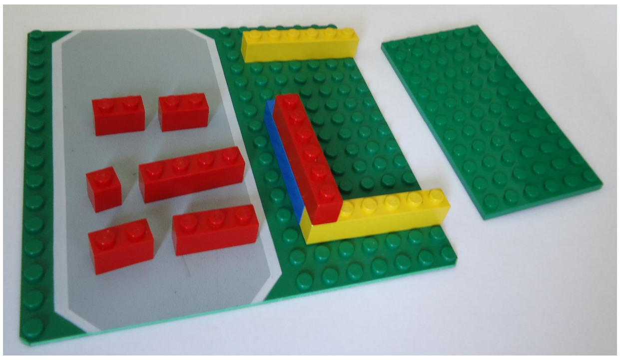


Figure 9. The need for a 5-brick.

This difficulty in organising bricks in three dimensions was not unique to this exercise; on the contrary, it is illustrative of characteristic behaviour.

Figure 10. Stacking vs. bonding bricks.

Once the house was built, the children added features, such as a surrounding wall (figure 10). Note the stacked bricks in the wall ï bricks of the same dimension piled atop each other.

It is worth emphasising the problems of three dimensional thinking by reference to the experiment with the fourth and sixth grades, because these children are of the age (11 and 13) where robotics is often introduced. Though not involved with the LEGO teaching, they knew the material from home. Like the younger children, they did not use the roof as a foundation template and therefore had trouble with the base area. Pictorial representation was insufficient for them to construct the house and they needed a physical model to scrutinise. The older children were notable more organised in collecting the bricks. One boy was notable for building up and out from one corner. So, how difficult is the construction aspect of robotics?

VI Discussion

An inability to build bonded walls with small plastic bricks at first appears well removed from robotics. However, capacity to visualise in three dimensions is fundamental to engineering. The inability of the children to apply their oral and pencil and paper mathematical skills to the task of construction resonates with complaints made by engineers, employers and academe, about the mathematical skills of school leavers.

A. Working on the flat

The technicity proposition suggests that human children do not apply innate three dimensional capabilities inherited from a simian past, rather they construct three-dimensionally from genetic data that is primarily two-dimensional. Children find it easier to work on the flat, or so it appears in the flat-world of schoolbooks. The notion that developmentally children move from a primarily oral world to the flatland of drawing and only later into space finds some support in studies of drawing in children [19] (orthogonal chimneys) and the work of Piaget. Without a doubt, traditional school method both follows this sequence and is book-bound. From the beginning of primary school to university graduation, the learner works on the flat. Is this efficacious?

B. Mathematics

Is it possible to say that children understand number when they cannot apply mental arithmetic skills to a simple problem where number does not exceed the digits of a hand? O Duill, [6] in discussing Turing teaching, where a computer rather than pencil and paper is the base educational medium, noted the disjunction between perceptual counting and the language of number in written computation. The 5-brick phenomenon observed by both Ilieva and Doyle raises yet another question. Does traditional mathematics teaching method lock childrenö understanding of number within the language domain?

C. Pictograph, language and science

Does the recent regression to pictography in computing, cf. WeDo software, exacerbate the isolation of domains: language in one box, construction in another? Have we, as Ó Dúill has suggested [20], by exchanging Logo for pictographs, lost the opportunity to extend childrenö understanding of language? The editor of LogoWriter beautifully introduced the difference between language that humans can understand and that which a computer can act upon. But, perhaps the loss has been even greater.

The issue of direction-of-rotation vs. direction-of-current is important. Science progresses by displacing concepts derived from false but feasible perceptions and replacing them with imperceptible ones that are congruent with physical reality. The classic is the heliocentric/geocentric view of the motion of the Sun and Earth. Electricity and magnetism is similarly not open to perception and must be revealed to consciousness through technology. The power that technicity provides may help direct thought to deeper understanding or may, equally, merely reflect appearances. The question, for teaching method in robotics is whether the pictographic approach leads to naïve rather than scientific thought. Or is it possible that naïve rather than scientific thought leads to misleading pictography?

The classic LEGO pictographic instructions have made the assembly of models relatively easy. More powerful and more universal than words, they are a congruent projection of three dimensions onto two. The operative word here is óassemblyö. Children who follow the step-by-step instructions to assemble the WeDo drumming monkey neither design nor create. In the English national curriculum, robotics is a part of Design and Technology. We teach children to understand their materials and thoughtfully creatively to construct: We want them to develop design capability. This does involve visualisation, but of a different quality from pictography. Indeed most advances in science and engineering have been based on visualisations rather than verbalisations: the Papertian object cognitively the more powerful. But this also applies to the words and letters of written language that primary school children are learning. Contribution to and congruence with the core curriculum is a requirement for primary school robotics.

D. Catch 22

Whilst agreeing, in the absence of an alternative, that kit-based courses enable children to experience robotics through construction, we must question what learning takes place. For children who await their adult front teeth, this is important.

But the problem is deeper. The schizophrenic authorship of this paper is not capricious. It signifies the conflict between a curriculum that has existed since writing was invented and the entrée of the computer over thirty years ago. Professionally, as a primary school teacher, the author must work to a syllabus [21] that seeks to inculcate ôICT literacyö through office-style applications from the earliest stage. Personally, he sees not a set of applications but a new medium, a Turing medium that challenges many of the assumptions that underlie traditional teaching and its method. Unfortunately, the current approach to robotics sets it apart from traditional academic curricula in a classic vocational niche, where the cognitive dissonance that thoughtful change requires is inhibited.

Perhaps the greatest inhibiting factor, with which the author has had personal experience as a small-time politician, is a lack of political will. Whilst affirming the STEM agenda, the political class is trapped in language-land. The membership of the British House of Commons does not include any engineers [22], and the nearest thing to a scientist is a medical doctor. At present, the education minister is a former journalist. In this milieu the thought necessary to begin break away from tried and tested curricula, syllabi and method in the presence of a new medium is inhibited by an agenda focussed on standards of attainment that are largely a function of traditional media.

VI Conclusion

The objective of this exploration of robotics in primary school was initially aimed at comparing the kit-based approach that tends to be the norm with a curricular model. It was hoped that technicity and the tree metaphor would help in thinking about this. It was not intended to consider maths or science teaching. However, classroom observations in the two contexts raised these matters. Science emerged first, raising questions about the power of pictography and its relation to language. Maths crystallised recently, with the question of shape and number, reviving concern about the efficacy of traditional mathematics method. Both highlight the importance of integrated curricula.

A. Theory

Technicity, which asserts that we construct technology from low-entropy genetic information, confirms Seymour Papertôs intuition that consciously constructing an object open to public is cognitively more powerful than a verbal formulation based on perception. Such an object may now be termed Papertian.

The primary school sees the emergence of technicity, which succeeds the language development of infancy, and is largely complete by its end. Hence, primary school is best viewed as a time of trunk-building. A transition from traditional to Turing teaching in these years, to reduce mental imbalance imposed by the arduous apprenticeship in literacy and numeracy, would be beneficial. It would facilitate the linking of technicity and language that is essential to communication; viz. this paper.

Change requires a necessity, at which the classroom results hint. One issue is the question of the pace of development of the capacity to think, to visualise, in three dimensions. Sound theory, though not prerequisite, is helpful. Technicity needs to be proven by other workers for it to provide the foundation of thoughtful constructionism.

B. Practice

The curriculum approach has been little applied. Most use of LEGO, in construction and robotics, remains experiential and pictographic: kit-based short-courses. The curriculum method requires either obsolete or scratch built materiel, including the software; both largely inaccessible to primary school. General change implies a market for course material. A TERECoP for the primary years specifically would be attractive, but course material would be required.

The author and his colleague will be further developing their approach in the next year, with the aim of documenting it. This will be facilitated by the availability of new facilities in the authorôs school. Hopefully, it will be possible to report on developments in theory and practice at a future RiE event.

Acknowledgement

The author wishes to acknowledge the contribution of Vessela Ilieva of NSOU ñSofiaò to the thought and practice reported in this paper. Given that it would not have been possible without her, she is its catalyst.

References

[1] O Dúill, M. (2012), "Reversing Robotic Regression: Why our Culture Rejects Robotics in School", In Proceedings of the 3rd international conference on Robotics in Education, RiE2012. September 2012., pp. 95-103. MatfyzPress, Czech Republic.

[2] Catlin, D., Smith, J.L. & Morrison, K. (2012), "Using Educational Robots as Tools of Cultural Expression: A Report on Projects with Indigenous Communities", In Proceedings of the 3rd international conference on Robotics in Education, RiE2012. September 2012., pp. 73-79. MatfyzPress, Czech Republic.

[3] Petrovic, P. (2012), "Having Fun with Learning Robots", In Proceedings of the 3rd international conference on Robotics in Education, RiE2012. September 2012., pp. 105-112. MatfyzPress, Czech Republic.

[4] Ó Dúill, M. (2012). ñThe Technicity Thesis: a constructionist proposition.ò Proceedings of Constructionism 2012. University of Athens, Athens.

[5] Bransford, J. D., Brown, A. L., Cocking, R. R. (eds.) (2000) ñHow People Learn, Expanded Edition.ò Washington DC: National Academy Press.

[5] Ó Dúill, M (2012) ñTowards Turing Teaching.ò Proceedings of Constructionism 2012, University of Athens, Athens.

[6] Alimisis, D. (2012), "Robotics in Education & Education in Robotics: Shifting Focus from Technology to Pedagogy", In Proceedings of the 3rd international conference on Robotics in Education, RiE2012. September 2012., pp. 7-14. MatfyzPress, Czech Republic.

[7] Arlegui, J., Pina, A. & Moro, M. (2011) "A Paradox in the Constructive design of Robotic projects in School", In Proceedings of 2nd International Conference on Robotics in Education (RiE 2011). Vienna, Austria, September, 2011. , pp. 29-34. INNOC - Austrian Society for Innovative Computer Sciences.

[8] Ilieva, V. (2010) ñLEGO and LOGO in the Primary School ï a simple way to learn through creation.ò In: Clayson, E.J., Kalaġ I, (eds) Proceedings of Constructionism 2010 ï 12th European Logo Conference. Faculty of Mathematics, Physics and Informatics Comenius University, Bratislava..

[9] QCA (2010). ñThe National Curriculum: Primary handbook.ò London: Qualifications and Curriculum Agency

[10] DfE (2013).ò Draft National Curriculum programmes of study.ò London: DfE. Online at: https://www.education.gov.uk/schools/teachingandlearning/curriculum/nationalcurriculum2014/b00220600/draft-national-curriculum-programmes-of-study [20/03/2013]

[11] Furber, S. (2012) ñShut Down or Restart: Report of the Royal Society into Computing in Schoolsò. The Royal Society, London.

[12] Alexander, R. (ed.) (2010) ñChildren, their World, their Education: Final report and recommendations of the Cambridge Primary Reviewò Routledge, London.

[13] Hubel, D.H. (1995) ñEye, Brain, and Vision.ò http://hubel.med.harvard.edu/index.html [12.05.2014]

[14] Striedter, G. Ó Dúill, M. (2012). ñThe Technicity Thesis: a constructionist proposition.òProceedings of Constructionism 2012. University of Athens, Athens [15] Fuster, J.M. (2008) ñThe Prefrontal Cortex.ò Fourth Edition, Academic Press, London.

[16] Dunbar, R. (2004) ñThe Human Story.ò Faber & Faber, London.

[17] Harel, I, & Papert, S (1991) ñConstructionism.ò Ablex, Norwood, NJ.

[18] Ileva, V.& Doyle, M. P. (2013) ñICT in the Bulgarian Primary Schoolò Proceedings of WCCE 2013, Torun (in press).

[19] Cox, M. (1992) ñChildrenôs Drawings.ò Penguin, London.

[20] Ó Dúill, M. (2012). ñBack to the Future: Can we reverse a quarter-century of regressionò Proceedings of Constructionism 2012. University of Athens, Athens

[21] Cambridge International Examinations, ñCambridge ICT Startersò http://www.cie.org.uk/qualifications/academic/primary/ictstarters/overview. [24/06/2013]

[22] McGuinness, F (2010) ñSocial background of MPsò, London. *www.parliament.uk/briefing-papers/SN01528.pdf* [24/06/2013]

Will robots take over Polish gymnasiums?

an attempt at implementing robotics into the gymnasium curriculum

Jarosław Kotliński
Wydział Nauk Pedagogicznych
Dolnośląska Szkoła Wyższa we Wrocławiu
Wrocław, Poland
jarkotlinski@interia.pl

Grzegorz Troszyński
robotyedukacyjne.pl
grzegorz@trobot.pl

***Abstract*—Due to its interdisciplinary nature, mechatronics, and especially robotics, its most spectacular and educationally attractive area, has been used as an educational tool at various levels of education for more than a decade. The article outlines an attempt at implementing the subject of mechatronics as regular compulsory classes in gymnasiums (students between the ages 14-16). Next to a description of the tooling, special emphasis has been put on the necessary shifting of the paradigm of the teaching model and the changing role of the teacher. A departure from a teaching model based on a transmission of knowledge in favor of a constructivist model with an extensive use of project work is suggested, together with a shift in assessment towards formative assessment.**

Keywords—educational robotics, robotics curriculum, constructivism, constructionism, secondary education, project based learning, educational technology

I. MOTIVATION FOR INTRODUCING ROBOTICS TO GYMNIASIUMS

As shown by the results of exams taken by the graduates of gymnasiums in Mathematics and the Natural Sciences, and technical subjects, the competences demonstrated by young Poles in these areas are insufficient. Incidently, this problem is not new, and not unique to Poland [1]. Students' humanistic competencies are on average several percentage points higher than their technical ones. This tendency has been growing for a few years now. When analyzing the situation one must relate it to the way the school teaching system is organized, especially when it comes to teaching technical competencies. The Teaching and Learning International Survey [2] has found that Polish teachers favor teaching methods based on a direct transmission of knowledge, which does not facilitate students' pursuing their interests or active learning. In the classroom, the teachers seldom use active learning, student-oriented methods, especially for teaching Maths and Natural Sciences. The number of teachers using active learning methods is almost four times lower than the one for humanistic subjects. The survey has also discovered the lack of both teaching programs for Maths and Natural Sciences and teachers that fully support students' active learning aided by modern teaching aids, including ITC. The content of Crafts and Technology is often out of touch with the latest developments in modern technology and the students' experiences, making the subject unattractive for the students.

Fig. 1. Student working with LEGO Mindstorms NXT kit.

It was the findings of the survey that inspired the authors to develop a comprehensive teaching program for mechatronics in gymnasiums. The program aims to address the questions of how and what to teach, how to assess outcomes, and what teaching philosophy to adopt in order to change the negative tendencies in acquiring technical knowledge and technical skills – so essential for life today.

II. TROBOT M-SYSTEM MECHATRONICS EDUCATIONAL KIT

To ensure the classes we propose offer the students a chance to act using modern teaching aids, we have decided to introduce elements of mechatronics as the leading theme of the innovative teaching programs for Crafts and Technology in gymnasiums and an extra subject Mechatronics. It has long been recognized that experiential, hands-on education provides superior motivation for learning new material, by providing real-world meaning to the otherwise abstract knowledge [3].

At the heart of realizing the educational content is the belief that "activity is the core quality of living organisms, and their way of existing" [4]. Human activity has a clearly defined direction set by its purpose, or goal, which also dictates how it is performed. Thus, the more interesting and more attractive a goal is, the more motivating and stimulating students' interests it becomes. Therefore, if the teachers who undertake to realize the mechatronics program want to achieve the set educational aims, they must understand that it is crucial to explain to the students the purpose of their activities and be able to motivate them towards achieving it. The aims can be achieved thanks to activity aid kits for constructing and programming robots. There are several types of construction kits available on the market (e.g. Lego Mindstorms NXT, VEX Robotics, etc.) however in order to fully accomplish the set aims, the authors have seen the need to create a kit dedicated to the specific target group made up of gymnasium students and teachers.

Fig. 2. Elements of the Trobot M-system school mechtronics kit

The key features of the Trobot M-system school mechatronics kit are:

- the elements of the kit are designed to look like real professional components (as opposed to toy-like products, which students of gymnasiums – 14 to 16 year olds - are reluctant to use),
- the modular, easy-to-extend design uses readily available and inexpensive parts end elements,
- wide range of programming languages and environments available
- the metal (aluminium) construction elements ensure durability, reliability, and versatility of use (they can be combined in many original ways depending on students' invention) and allow to build large size structures,
- the microcontroller board has been placed in a transparent polycarbonate casing that protects it from mechanical damage,
- visual (LED) indicators of main microcontroller board function (communication, motors, power),
- motor drivers protected against over current, short circuit, under voltage lockout and overtemperature,
- wiring of sensors and actuators is baseed on error-proof RJ-type connectors,
- power supply protection against over- and under voltage, reverse polarity,
- battery monitor preventing deep battery discharge,
- the most common size AA rechargeable batternies are used, promoting economical and ecological soution.

The controller of the school mechatronics Trobot M-system kit with the ATmega32U4 microcontroller is equipped with a 4 channel motor controller. The implemented PID controller allows controlling both the speed and the position of the motors with incremental encoders. I/O capability of M-system controller:

- four DC brushed motors @6V, max 2A
- four standard servos with 7.2V power supply
- four analog output sensors with (0-5V output range)
- up to four sensors on I2C (TWI) bus
- extension modules are also available
- Arduino 1.0 pinout

Wireless communication with external devices, including PC hosting Prophio application, is done by 2,4GHz Bluetooth module working in SPP mode. A USB connection allows easy interfacing with PC, including programming the board without the use of an external hardware programmer (the board comes with preburned Arduino bootloader). The M-system microcontroller board is fully compatible with Arduino system, allowing the use of hundreds Arduino shields (extension boards) widely available [5].

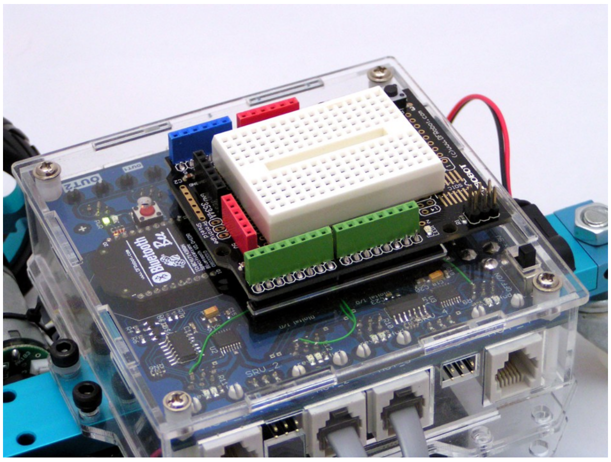


Fig. 3. Controller of the Trobot M-system school mechatronics kit, Arduino-compatible, with an additional prototype module

III. PROGRAMMING ENVIRONMENT

To complement the mechanical and electronic elements, the third core component of the school mechatronic kit is the programming language and environment. The authors propose the use of the Prophio graphic programming, based on the popular and widely used educational Scratch and BYOB programs [6] [7].

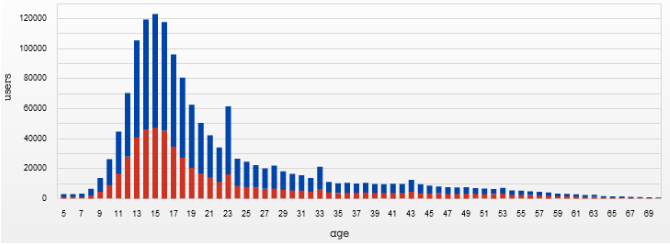


Fig. 4. Scratch registered users by age and gender [8]

Scratch is a project of the Lifelong Kindergarten Group. It was developed in 2003 at the Massachusetts Institute of Technology to address the needs and interests of the young people attending extra-curricular classes run by Intel Club Houses. Since its launch, 3 million programs created by 1.4 million users from nearly all parts of the world have been posted on the project's website [9]. The site is growing at over 40 thousand new users and 200 thousand new programs each month. The core membership on the site is between the ages of 13 and 16 – which is exactly the age of the students in Polish gymnasiums.

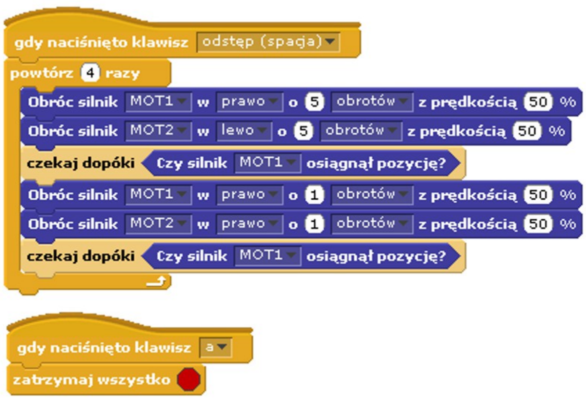


Fig. 5. Sample program using advanced location and motor speed functions of a mobile robot.

What makes Scratch and its modifications so popular? Scratch allows creating programs without the need to remember and enter commands following the strict rules of the traditional text languages. What is more, Scratch makes it possible to add multimedia content, such as photos, graphics, and sounds, to the created programs. The embedded editor of raster graphics enables editing existing objects and creating new ones; it is easy to add your own recordings to the sound library that comes with the program. What is more, creating the program code is like putting together pieces of a puzzle, which produces a formally error-free code, making it easier for first-time programmers. Scratch is an open-source project, which results in its many modifications and extensions, for example BYOB (Build Your Own Blocks), which allows to create your own functions and procedures, or Prophio (Programming of Physical Objects), which provides extensive opportunities to control external devices based on the Trobot M-system kit.

The M-system controller and the PC with Prophio running communicate by radio, which allows changing a program even with the device in operation, and observing immediate result of the changes introduced. Radio communication also makes it possible for a physical device and the objects on the computer screen to interact, which lends the created programs new possibilities, inaccessible for compiled languages. For example, incoming data from sensors attached to the controller can be used to control the movements of an animated object on the screen or conversely, to control a mobile robot with the keyboard or a mouse. Blocks have been designed to allow controlling motor speed, its revolutions, and the position of the servo, or reading the controller input. Special blocks have been prepared for selected sensors, which return the processed value of a measurement, e.g. the distance in centimeters, for the Sharp GP2D12 distance sensor.

It needs to be highlighted that Prophio, a development on BYOB, can also be used to teach more complex programming issues, such as recurrence, types of data, procedures and functions, or conditional instructions, which makes it valuable for teaching in high schools. There are numerous instances of university courses using this environment [] On the other hand, the modular design of the M-system kit enables the use of text programming languages and programming environments, open-source Arduino or the C/C++ language being of special interest here.

IV. An Attempt at Implementing in School Practice

For more than a decade numerous academic centers, schools, societies, and vendors have been trying to introduce robotics in education at practically every level, from pre-school to university. The advantages of robotics as a tool that enhances the quality of education in schools, develops technical and social skills, and engages young people's interests in science and technology, are widely appreciated. On the other hand, there is no consistent, uniform, coordinated approach to the subject, or a thorough assessment of outcomes [10]. This is true for both national and European initiatives. There is a wealth of courses in robotics run as extra-curricular classes, interactive labs organized at science centers, or robotics competitions, some of which worldwide. However, implementing robotics into the school curriculum remains limited to a few individual forms in a handful of schools. The barriers to wider implementation mainly concern [3]:

- Lack of teacher time
- Lack of teacher training
- Lack of age-suitable academic materials
- Lack of ready-for-use lesson materials
- Lack of a range of affordable robotic platforms

The authors have developed the "Mechatronics in Gymnasiums" teaching programs and a two-module "Crafts and Technology classes in implementing mechatronics "program as a project realized under Sub –measure 3.3.4 “Modernization of education's content and methods" of Priority III of the Human Capital Operational Program. The programs aim to meet the need to prepare didactic materials for students as well as teachers. The Center for Education Development in Warsaw is the Intermediate Body and Implementing Authority.

The programs have drawn on the experience gained by Trobot instructors between 2011 and 2013 when working with students of 18 different gymnasiums in 4 voivodeships in central Poland. The classes were of various characters, from trial lessons to regular, 48-hour courses, and were run with teachers of Maths and Natural Sciences, ICT, And Crafts and Technology. Their contributions, together with extensive consultations with heads of schools and educational authorities, have led to developing a model for introducing mechatronics into the school curriculum.

The project sets to introduce the teaching programs in 32 gymnasiums as a pilot study, in a form that may be readily adaptable to an individual school's needs:

- Crafts and Technology classes in implementing mechatronics (min.65 hours per course)
- Mechatronics – an additional, optional subject integrating mathematical, physical, IT, and crafts and technology content (min.65 hours per course)
- extra-curricular classes "Youth technology clubs" – targeting students with special educational needs

40-hour trainings for 64 teachers from all the 32 gymnasiums taking part in the project will be organized. Additionally, for 2 years the teachers will be supported by experienced instructors. At the same time, a platform for exchanging experiences of teachers and heads of schools, along with a range of active promotional activities, have also been planned.

V. Projects in the mechatronics classroom

To realize the subject of mechatronics right, the classes rely on a consistent use of individual and team projects. The final component of the program – the Engineer's Work – designed to review and put to creative use the knowledge and skills acquired during the classes, is a team project that takes several hours to complete. We believe that this method develops students' autonomy, is highly motivating, and enables to combine the different learning styles and personal characteristics of the individual students. Moreover, it helps to prepare the students to take and own responsibility for their learning now and for life. It outfits the students with the skills necessary to manage projects independently both in their future jobs and lives, and fosters entrepreneurship, which is a crucial skill in today's world.

If a project is carried out well, it teaches discipline, organization, realistic goal-setting, measuring, testing and presenting results. The teaching by projects method is also very flexible to use – it works as well with small-size, short-term projects concerning a fragment of a class, as with grand-scale, team projects which run for several hours and involve the cooperation of several students [11].

VI. Assessment in the mechatronics classroom

Every considerate teacher asks themselves a lot of questions about assessing students' performance. Many of these questions are how to asses to help a student learn. The authors believe that the expectations of an ambitious and creative teacher are addressed by Formative Assessment. The method goes with the "philosophy" of the subject, which is why it is favored by the designers of the program. We would therefore suggest that teachers employ formative assessment techniques along with the summative ones (grading). Formative assessment provides students and parents with more comprehensive and more detailed feedback on the students' grasp, performance and outcomes, the progress made and possible difficulties, pointing out the potential sources of the problems. Formative assessment undoubtedly supports the learning process, as it makes it easier for the student to understand the purpose of the classes, what he or she is expected to accomplish, and the criteria for evaluating their own and their schoolmates' work. Consequently, the student's independence and sense of ownership of their own learning increase.

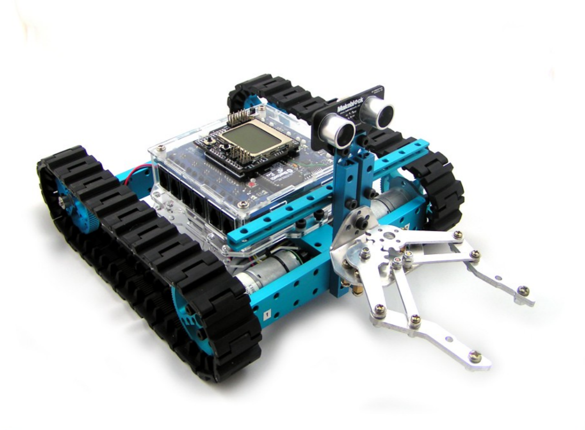


Fig. 6. Mobile tracked platform built with the Trobot M-system kit

As mentioned before, formative assessment is an element of humanistic education, whose main aim is to help the students learn – it motivates and engages, allows monitoring their own progress, and facilitates taking responsibility for their own learning. Formative assessment is also a kind of a toolbox that helps the teacher to organize the students' learning space in an attractive way.

According to professor Kwieciński, the school is for many students a place to spend time in without the effort and "distress" of work, and attending classes has become "…a process of acquiring by the student, class after class, day by day, the ethos and habit of avoiding work" [12]. To counter this process, and make the school a place of intensive daily work for teachers and students the authors highly encourage the use of formative assessment, whose basic tools are presented below.

The assessment applied by schools and individual teachers in their everyday teaching experience (continuous, end of term, and final assessment) reflects, to a greater or lesser extent, their psychological and philosophical views. The teachers who advocate the "positivist" approach to knowledge and the behavioral school of human development tend to concentrate on assessing the outcome of the learning process. The humanistic teachers, or "reflective practitioners," on the other hand, prefer to monitor the learning process on a continuous basis, and discuss it with the students and their parents. The different approaches determine the choice of the assessment methods and the degree to which the student has an influence on his or her assessment process [13]. By assessment method we mean the type of activity performed by the teacher in order to gather empirical evidence which will then allow to estimate the various aspects of the students' learning, and consequently to asses student's achievements. The authors of the program recommend the teacher to employ a humanistic, reflective, and motivating approach when assessing students' achievements in mechatronics, and choose methods complementary to formative assessment. Among others:

- construction tasks (e.g. building robots)
- experiment (e.g. choosing a gear ratio in order to achieve the expected speed of a vehicle)
- observation of students at work (e.g. activity, interest, originality of ideas, peer help, self control)
- presentation of outcomes (e.g. multimedia presentations)
- drawing tasks in SketchUp,
- educational project
- coupled with:
- tests (e.g. quizzes)
- measuring quantities (e.g. distance, time etc.)
- written assignments (tables, crosswords, gap filling)

When realizing tasks, the teacher should pay attention to:

- respecting Health and Safety procedures,
- proper organization of the workstations,
- using equipment, tools, and devices,
- the ability to notice problems, make and verify hypotheses, draw conclusions, and reason, the ability to work with instructions (diagrams), being active and creative when working.

VII. Conclusions and Future Actions

The value of robotics, or more generally mechatronics, as a tool for enhancing the quality of education in teaching Maths and Natural Sciences and ICT has been tested in numerous limited-scale activities. The project outlined in this paper is an attempt at implementing Mechatronics into the school practice in an organized way, on a scale that would enable conducting systematic research on the true effectiveness of the tool. Two teaching programs based on elements of mechatronics, complete with teaching resources– a student's book and a teacher's book have been developed. Regular classes are scheduled to begin in September 2013, with almost 3 thousand students between the ages of 13 and 16, in 32 gymnasiums. Monitoring and evaluation of the project will run for two years, until 2015, when the final results will be presented.

[1] Miller, David, and Cathryne Stein. "Oh THAT's What You Use π For!." Autonomous Robots 1 (1995): 3.

[2] Teaching and Learning International Survey (TALIS), Nauczanie – Wyniki Badań 2008, POlska na tle miedzynarodowym, Ministerstwo Edukacji Narodowej, Instytut Badań Edukacyjnych, Warszawa 2009.

[3] Mataric, Maja J., Nathan Koenig, and David Feil-Seifer. "Materials for enabling hands-on robotics and STEM education." AAAI spring symposium on robots and robot venues: resources for AI education. 2007

[4] TYSZKOWA M., Aktywność i działalność dzieci i młodzieży, WSiP, Warszawa 1990, s. 6

[5] Arduino. (2013, June 26). [Online]. Available: http://www.arduino.cc 2013

[6] Resnick, Mitchel, et al. "Scratch: programming for all." Communications of the ACM 52.11 (2009): 60-67.

[7] Harvey, Brian, and Jens Mönig. "Bringing "No ceiling" to scratch: can one language serve kids and computer scientists." Proc. Constructionism (2010).

[8] MIT Media Lab. (2013, June 26). [Online]. Available: http://stats.scratch.mit.edu/community/usersbyage.html 2013

[9] MIT Media Lab. (2013, June 26). [Online]. Available: http://wiki.scratch.mit.edu/wiki/scratchers 2013

[10] Bredenfeld, Ansgar, Alexander Hofmann, and Gerald Steinbauer. "Robotics in Education Initiatives in Europe-Status, Shortcomings and Open Questions." Workshop Proceedings of Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010). 2010.

[11] Miko – Giedyk J., „Udział szkoły i nauczycieli w procesie wyrównywania szans edukacyjnych młodzieży wiejskiej", [w:] (red.) Męczkowska – Christiansen A., Mikiewicz P., Idee – diagnozy – nadzieje. Szkoła polska a idee równości, Wydawnictwo Naukowe Dolnośląskiej Szkoły Wyższej, Wrocław, 2009

[12] Kwieciński Z., „Pedagogie postu". Preteksty – konteksty – podteksty, Wyd. Impuls, Kraków 2012 r. s. 233

[13] Kwieciński Z., Śliwerski B. (red.), Pedagogika. Podręcznik akademicki, Wydawnictwo Naukowe PWN, Warszawa 2003

Building a Rube Goldberg Machine using Robotic Toolkits. A Creative, Hands-on Exercise for Secondary School Students.

Dorit Assaf
Artificial Intelligence Laboratory
Department of Informatics
University of Zurich
Switzerland
Email: assaf@ifi.uzh.ch

***Abstract*—We describe in this paper a Rube Goldberg Machine exercise which we conducted with secondary school students in a real classroom environment. Our aim was to train a number of important skills young people need such as team work, creative thinking, problem solving, and time management. Our goal was further to contribute to engineering education by using robotic hardware to solve the task. A Rube Goldberg Machine is a deliberately over-engineered machine that performs a very simple task in an unnecessary complex way, often by mechanical chain reactions. The students had to come up with creative ideas for each part of the machine's chain reaction, design, build and program them using robotic toolkits. We describe the exercise setup and its results. We show that it is very appealing and suitable for the classroom. Opposed to typical educational robotics activities, where teams compete against each other, here the students have to collaborate to create one common artwork.**

I. Introduction

Since the 1980's universities notice a constant enrollments decline in STEM (science, technology, engineering, mathematics) related disciplines [11], [13]. The consequent lack of workforce in a society that on the other hand demands more and more on skilled people in these disciplines caused stakeholders to take measures. Institutions and projects to promote science and technology were founded (e.g. Roberta [5], FIRST[1], IPRE[2], CEEO[3], KIPR[4]) and a variety of attractive out-of-school activities were created. The use of robotic platforms as a learning tool in the classroom became increasingly popular at all levels of education [17], [7], [4]. Robotics seems to unify a large number of required skills such as problem solving, logic reasoning, computer science, engineering as well as team work. Another important reason that speaks for the use of robots in class is its hands-on approach, which facilitates the application of pedagogical principles such as constructionism, problem-based learning as well as collaborative learning [14], [1]. Popular educational robotics platforms are car-like robots (e.g. Bee-bot[5], Asuro[6], Thymio[7], Boe-bot[8], ePuck[9]), humanoids (e.g. NAO[10], Bioloid[11]), or the most popular, the flexible platform LEGO Mindstorms[12]. Further, a number of international robot competitions for students emerged within the last decade, such as RoboCupJunior[13] [16], [15], EUROBOT[14], FIRST LEGO League[15], Botball[16], Robolympics[17], World Robot Olympiad[18], and Robotour[19]. The properties of a robotic platform influences the exercise design and consequently the skills that can be trained. Typical robot projects in both classroom activities and competitions are: robot soccer, robot rescue (navigation through unknown terrain to find and transport an object), robot sumo, robot dance, maze solving, fire fighting, pick and place of objects, line following, etc.

In the theory of play and learning two opposite forms of play can be distinguished: games and free play [6]. Games are structured, predefined, rule-bound and goal-directed, whereas free play is unstructured, spontaneous and improvisational. Games train special skills, lead to master a certain action or product, or the discovery of a satisfactory solution. Free play is chaotic, infinite (has no logical ending point), spontaneous, improvisational, and creates an own construction of meaning. In free play, no complex learning should be involved, experimentation should rather be encouraged. Open-ended play is positioned between games and free play. Initially, it might have free play characteristics, but it resembles games as the players come up with rules and goals. Opposed to free play, open-ended play restricts players in their free play as it offers objects with design intentions. Designs for open-ended play should be on one hand specific and easy to understand on the

[1] http://www.usfirst.org/roboticsprograms
[2] http://www.roboteducation.org
[3] http://www.ceeo.tufts.edu
[4] http://www.kipr.org/

[5] http://www.terrapinlogo.com/bee-botmain.php
[6] http://www.arexx.com/arexx.php?cmd=goto&cparam=p_asuro
[7] https://aseba.wikidot.com/en:thymio
[8] http://www.parallax.com/
[9] http://www.e-puck.org
[10] http://www.aldebaran-robotics.com
[11] http://www.robotis.com/xe/BIOLOID_main_en
[12] http://mindstorms.lego.com
[13] http://www.robocup.org/robocup-junior/
[14] http://www.eurobot.org
[15] http://www.firstlegoleague.org/
[16] http://www.botball.org/
[17] http://www.robolympics.ch
[18] http://www.wroboto.org
[19] http://robotika.cz/competitions/robotour/en

other general enough to foster imagination and creativity.

Our aim was to create an educational robotics classroom activity which enables open-ended play and free experimentation. Due to their game characteristics, traditional educational robotics activities do not emphasize open-ended play. We believe that both games and open-ended play must have their place in a curriculum, the former to train specific skills, the latter to foster creativity. We describe in this paper the "Rube Goldberg Machine exercise" which was part of a series of different exercises conducted in the context of the EmbedIT project. The aim of this project was to contribute to science and technology education by developing a novel open-source robotic kit (EmbedIT[20]). The toolkit should enable students to explore in a motivating and fun way interdisciplinary topics such as (biologically inspired) robotics, embodied artificial intelligence, electronics, computer science, neuroscience, psychology, arts, etc. We designed a number of example exercises using this toolkit which train a variety of skills young people need for a successful professional life. The example exercises further tested the versatility and usability of the toolkit. All exercises were conducted with students (secondary school, undergraduates) in real classroom environments. The "Rube Goldberg Machine" was one of the example exercises, which aimed to foster open-ended play, creative and artistic skills, where the focus was not on building or controlling a robot. For more details about the toolkit, the other example exercises, and results we refer to the thesis "EmbedIT – an Open Robotic Kit for Education" [3]. We describe as follows the Rube Goldberg exercise idea, its setup, and discuss the results.

A *Rube Goldberg Machine*, named after the American cartoonist and inventor Reuben Goldberg (1883-1970), is a deliberately over-engineered machine that performs a very simple task in an unnecessary complex way, often through chain reactions. In his cartoons he drew humorous, unrealistically complicated machines that solve trivial tasks such as turning off the lights, opening the garage door and watering the plant (Figure 1). He never built the machines he drew, but his cartoons have become an inspiration to engineers, students and artists across the world that actually constructed similar kind of complicated machines. A number of art installations exist, inpired by Rube Goldberg (e.g. "Der Lauf der Dinge" by Fischli & Weiss in 1987, "Cog" TV commercial by Honda in 2003), as well as a board game ("mouse trap", 1963). The machine found further its way into classrooms, for instance the yearly Rube Goldberg machine competitions at secondary school level[21], or for teaching topics such as simulations and physics [10], [8], [9], [12]. Traditionally, Rube Goldberg Machines consist of purely mechanical chain reactions. In our exercise we encouraged the students to use in addition robotic components such as sensors and actuators.

The idea to build a Rube Goldberg machine is very appealing and suitable for class. The construction of this machine requires a variety of important skills. The teacher or the students basically just predefine a simple task which has to be executed by the machine. The students have to be very creative to come up with fun ideas for the chain reactions. The students should not be constrained by a fixed mechanical construction kit, moreover they should be encouraged to use any object they can find around them. The more a daily-life object is used in an unusual way to release the next step in the machine's chain reaction, the more interesting and fun it is. Further, the chain reactions are not more than simple sensing and actuation actions. The students learn about different ways how a chain reaction can be implemented (either through purely mechanical "sensing" and actuation such as for instance a ball rolls and pushes another object, or through sensors that are coupled with actuators using a predefined control program). As a next step the students have to find mechanical construction solutions to incorporate the sensors and actuators in the machine, sensors have to be calibrated, control routines have to be programmed. Additionally, the very important skills needed in this exercise are team work, problem solving, and time management. Opposed to typical robot education activities, where teams compete against each other, here the students have to work together to create one common artwork. Every team is responsible for an equally important subpart of the machine (i.e. one or several chain reactions). The students have to collaborate with their neighboring teams on how to interface their individual chain reactions together. Each group's work is equally important and crucial for the success of the machine. Time management plays an important role as well if the machine has to be finished on time. Especially because the students usually underestimate the consumed time for testing, since no Rube Goldberg machine runs neatly in the first try. The final result is an artwork with equal contribution by every team. The machine can further be filmed and put on-line for friends and family.

This exercise concept might be especially appealing for students that normally don't like competitions. Further, since many robotic competitions often value specific properties or behaviors of an agent (speed, navigation), students may stick to established effective solutions (e.g. building a wheeled robot to achieve fast locomotion) rather than exploring more creative and unusual ways to solve a problem. With this Rube Goldberg machine exercise out of the box thinking is appreciated and open-ended play is fostered.

II. Exercise setup

We conducted the Rube Goldberg Machine exercise in two occasions, at the Zürcher Hochschule der Künste (ZHdk) and Kantonsschule Trogen with total 24 students between 12 and 19 years. The students formed teams and were provided with a number of different sensors for exploration: force sensitive resistors, accelerometers, gyroscopes, switches, flex sensors, liquid level sensors, tilt sensors, piezo vibration sensors, different kinds of potentiometers, compass, temperature sensors (digital and infrared), pressure sensors, different distance sensors, light sensors. The provided actuators were vibration motors, RC servo and DC motors. After the exploration phase the students chose the sensors and actuators for their part of the chain reaction. We further provided mechanical construction parts from LEGO, Meccano[22] and Stokys[23] as well as a other material such as cardboard, duct tape etc. (Figure 5c and d). In addition, we encouraged the students to consider using any object that can be found in the room.

[20]http://www.embed-it.ch
[21]http://rubegoldberg.com/?page=contest
[22]http://www.meccano.com
[23]http://www.stokys.ch

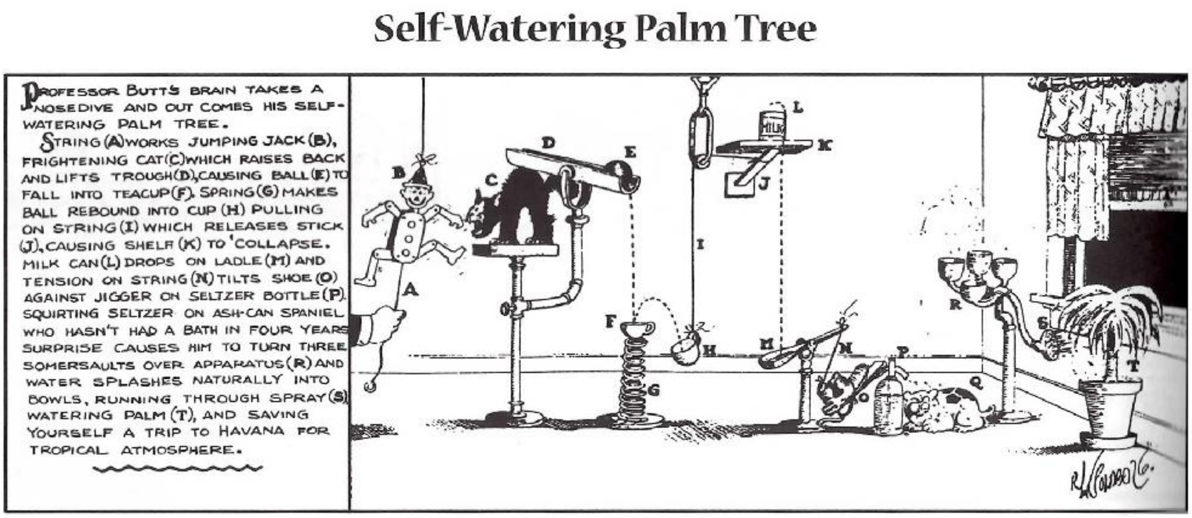


Fig. 1. A cartoon of Reuben Goldberg. An unnecessarily complex machine to water a palm tree. Picture courtesy of http://www.rubegoldberg.com

The EmbedIT kit provided the robotic hardware and the software user interface to the machine. Due to its generic module design, the toolkit enables the attachment of a variety of sensors and actuators to a system without having to deal with electronic circuits (Figure 2). Each EmbedIT module is connected to a bus and detected automatically by the software. With the graphical programming environment the students could easily define the sensory-motor control (Figure 3). For further information about this toolkit, refer to [3], [2].

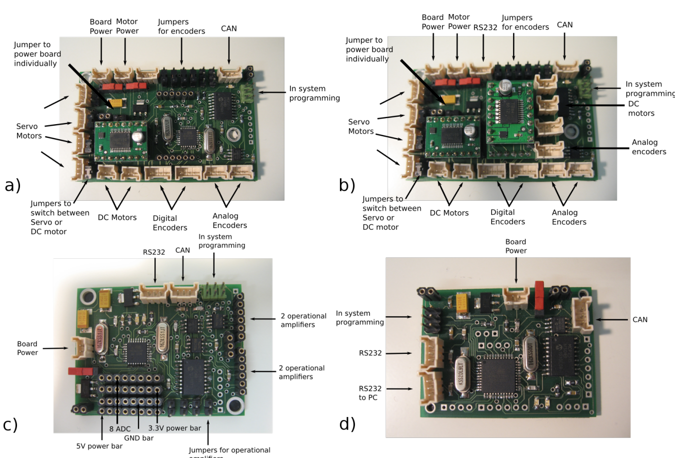


Fig. 2. A detailed description of each EmbedIT module. a) an actuator module supporting either two DC motors or four servo motors, b) an actuator module with DC motor extension attached. This allows to attach two additional DC motors, c) a sensor module, d) a master module.

A. *Junior design department at ZHdk, July 17th 2012*

The ZHdk interaction design department gives kids the opportunity to learn about design in a summer school week called "Junior Design Department". The goal was to build a Rube Goldberg machine in one day (6 hours) using the EmbedIT toolkit. Twelve students between 12 and 16 years old (average age 14, two female) attended this workshop, most of which have never worked with robotic systems before.

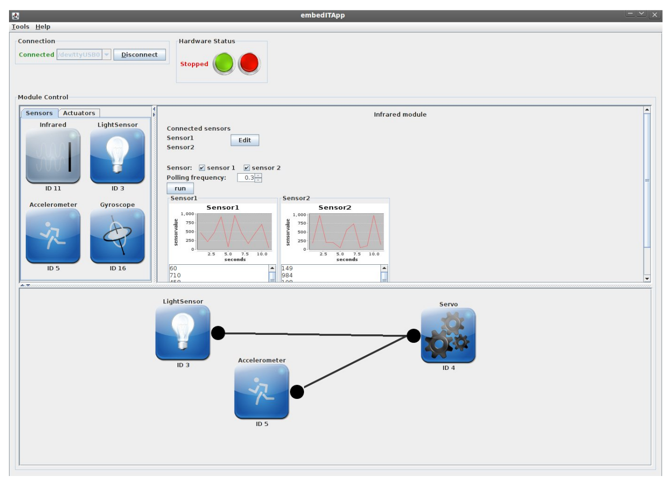


Fig. 3. An example of the EmbedITApp where a servo module and four sensor modules (infrared, light, accelerometer, gyroscope) are listed. Each available module in the application is represented with its ID, icon and a button (upper left column of the application). To access and control a module directly, the user has to click on a module button to display its control panel (upper right column). In the graphical programming area (lower panel) sensor motor relationships can be defined.

The Rube Goldberg machine consisted of five subparts. Part one: A small ball is released which pushes down a flap over a light sensor. The decreasing light level is detected and releases the LEGO car (Figure 4a). Part two: The bumper of the LEGO car bends a flex sensor located at the edge of the table which turns the white wheel (Figure 4b). The white wheel pushes the small car wheel which knocks over the lit candle. Part three: the temperature sensor detects the candle and turns a servo motor. Part four: The servo motor rolls a horizontal bottle which pours water into a container. The liquid level sensor detects the increased liquid level and moves a servo motor (Figure 4c). Part five: the servo motor pushes a ping-pong ball which rolls down a ramp. At the end of the ramp the passing by ball is detected by a distance sensor located at the rear of the waiting car. The car is released and pushes a pointed stick to a balloon filled with confetti. The balloon bursts and the confetti is released (Figure 4d).

Of course the whole chain reaction was somewhat ambitious and did not work initially. We needed several tries until it finally worked. The execution duration was seven seconds at a length of six meters (see video[24]).

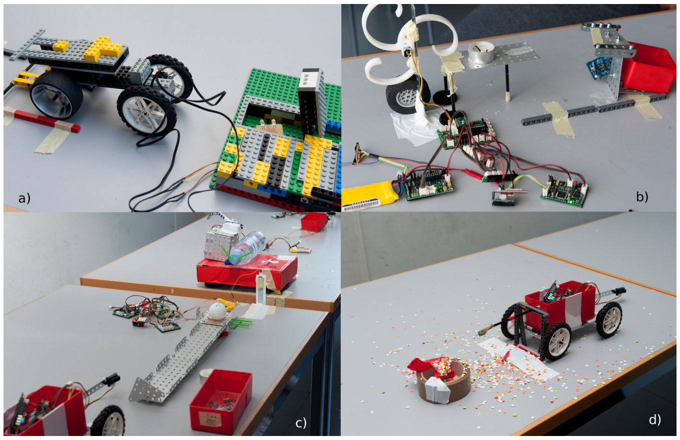


Fig. 4. The constructed Rube Goldberg machine. a) A small ball is released which pushes down a flap over a light sensor. The decreasing light level is detected and releases the LEGO car. The bumper of the LEGO car bends a flex sensor which turns the white wheel. b) The white wheel pushes the small car wheel which knocks over the lit candle. The temperature sensor detects the approached candle and turns a servo motor. c) The servo motor rolls a horizontal bottle which pours water into a container. The liquid level sensor detects the increased liquid level and moves a servo motor. The servo motor pushes a ping-pong ball which rolls down a ramp. At the end of the ramp the ball is detected by a distance sensor located at the rear of the waiting car. d) The car is released and pushes a pointed stick to a balloon filled with confetti. The balloon bursts and the confetti is released.

Fig. 5. Students working on a Rube Goldberg machine. a) and b) setting up chain reaction number two (flex sensor turns wheel which releases a small wheel), three (small wheel knocks over a candle which is detected by the temperature sensor and moves servo) and four (servo pushes a bottle which pours water into a container. A liquid level sensor detects the increases liquid level and moves another servo motor). c) and d) Students combined mechanical construction kits such as LEGO, Meccano and Stokys with other material such as cardboard and duct tape.

B. Kantonsschule Trogen, September 26th 2012

Trogen is a rural village in eastern Switzerland. 12 students between 17-19 years (average 18, two female students) built a Rube Goldberg Machine (Figure 6). We asked about the students' backgrounds. 83% major in natural sciences, the rest in economics and arts. As future study interests 67% mentioned technical disciplines, followed by other disciplines such as economics, medical doctor, or pedagogy. The full day workshop duration was still a little too short for building the planned machine. Only half of the it was ready and worked by the end of the day (though some parts were also quite ambitious such as a marble that was supposed to be tossed by a catapult into a cardboard box).

Fig. 6. Students of the Kantonsschule Trogen are building a Rube Goldberg machine.

III. RESULTS AND DISCUSSION

We used questionnaires to collected feedback from the students about their background, the class itself, and the toolkit. Due to the low number of students in each class, we mainly used the collected data to get a qualitative idea, which always has to be looked at in combination with our observations and personal conversations. In the questionnaires we asked questions such as "did you enjoy the class?", "have you learned a lot?", or "would you recommend the class to other people?". The questionnaire contained further questions about the toolkit used, we show here just the relevant data for the exercise itself. Figure 7 shows the questionnaire results of both the ZHdK students (left three bars) and the Trogen secondary school students (right three bars). Overall, the students liked the exercise and would recommend it to other people. With the question "have you learned a lot?", we wanted to get a feeling about how much they think they benefit from this exercise, knowing this is very hard to measure and highly subjective. The older students had the impression they haven't learned so much. This might be explained by the fact that they had prior experiences in building robots with LEGO opposed to the younger students at ZHdK. Further, this workshop was only held as a unique event. Probably additional theory about sensors and actuators, design concepts of mechanical engineering would have increased the feeling of having learned a lot. More theory should be considered in the future, however, in the context of our experiments this was not possible due to time constraints. For future instances of this exercise we would ask this question more precisely (e.g.

[24] http://youtu.be/_CaaPSq5Ohk

which skills have been trained, such as team work, mechanical engineering, electronics etc.). The toolkit proved to be suitable for this kind of class activity. Even those students who never worked with robots before could build a system using a variety of sensors and actuators within a short time. It seems to provide the design required for open-ended play which is on one hand specific and easy to understand on the other hand general enough to foster imagination and creativity.

The students explicitly mentioned in the free comment section of the questionnaire the appreciation of the creative work by building the parts of the Rube Goldberg machine. This open and unconstrained exercise was especially demanding for the EmbedIT hardware since a large number of different sensors and actuators had to be integrated within only a few modules. We observed generally a high engagement of the students. However, they tended to aim high and underestimate the effort of building the chain reactions. We further observed that they neglected the testing and tuning part of the machine. It might lead to frustration if the students don't finish on time and cannot run the machine as planned. Like in real work life the keys to success when developing a system are time management, early testing, and incremental increase of complexity. This Rube Goldberg Machine trains all that in a creative and fun way.

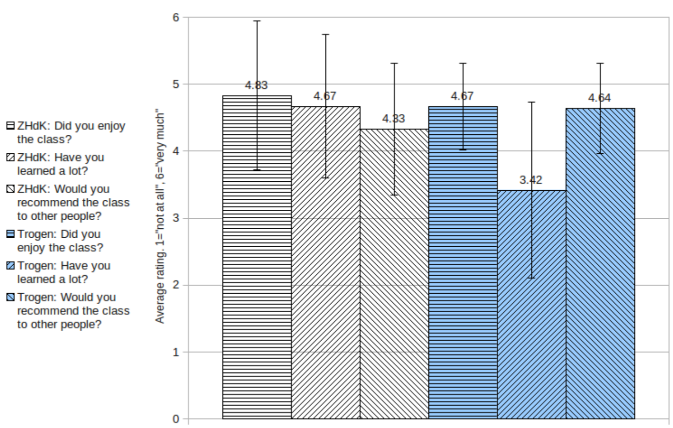


Fig. 7. Survey results of the Rube Goldberg machine exercise at ZHdK (12 students) and Kantonsschule Trogen (12 students). The bar chart shows the average rating to different questions concerning the class. The scale ranges between one and six, where one stands for "not at all" and six represents "very much".

IV. Conclusion

We presented in this paper the Rube Goldberg Machine exercise which enables open-ended play and aims at training a number of important skills including time management, team work, problem solving, prototyping, mechanical design, and working with robotic hardware and software. The exercise was conducted with 24 students between 12 and 19 years in real classroom environments. Educational robotics initiatives often focus on building robots and/or programming them to solve "traditional" engineering tasks such as navigation through space and pick and place of objects. Further, encouragement is usually given in form of a competition. We showed, that engineering education can also occur in a less competitive way. By building a Rube Goldberg machine each group is equally responsible for the success of this machine. The teams have to collaborate with each other, e.g. by defining interfaces. The result is an art project which can be recorded and shared with friends and family.

Acknowledgment

We would like to thank all students attended the Rube Goldberg Machine workshops. A special thanks further to Christian Fitze, Florian Wille, and Moritz Kemper that provided class time to conduct these exercises with their students.

References

[1] Alimisis, D., Moro, M., Arlegui, J., Frangou, S., Papanikolaou, K. (2007). *Robotics & Constructivism in Education: the TERECoP project.* In Proceedings of the 11th European Logo Conference, pp. 111. Comenius University Bratislava

[2] Assaf, D., Pfeifer, R. (2011). *EmbedIT – an Open Robotic Kit for Education.* In Proceedings of Research and Education in Robotics – EUROBOT 2011 (pp.29–39) CCIS 161, Springer-Verlag Berlin Heidelberg, ISBN 978-3-642-21974-0

[3] Assaf, D. (2013). *EmbedIT – An Open Robotic Kit for Education.* Ph.D. thesis. Artificial Intelligence Laboratory, Department of Informatics, University of Zurich

[4] Barker, B.S., Nugent, G., Grandgenett, N., Adamchuk, V.I. (2012). *Robots in K-12 Education: A New Technology for Learning.* Premier reference source. IGI Global. ISBN 9781466601826

[5] Bredenfeld, A. and Leimbach, T. (2010). *The roberta initiative.* In Workshop Proceedings of SIMPAR 2010 Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, pp. 558–567

[6] De Valk, L., Bekker, T., Eggen, B. (2013). *Leaving Room for Improvisation: Towards a Design Approach for Open-ended Play.* Proceedings of IDC 2013 The 12th International Conference on Interaction Design and Children. ACM

[7] Gerecke, U., Wagner, B. (2007). *The Challenges and Benefits of Using Robots in Higher Education.* Intelligent Automation & Soft Computing Volume 13, 1. pp. 29–43

[8] Howard, W., Williams,R., Yao, J. (2010). *Simulations of Carnival Rides and Rube Goldberg Machines for the Visualization of Concepts of Statics and Dynamics.* Engineering Design Graphics Journal Volume 74, 2 p1-11

[9] Jordan, S., Adams, R. (2008). *..A good imagination and a pile of junk.* American Society Engineering Education Annual Conference, ASEE, AC 2008-2031

[10] O'Connor, D. (2003). *Application sharing in K-12 education: Teaching and learning with Rube Goldberg.* TechTrends, Volume 47,5, Springer US, pp. 6–13

[11] Patterson, D. A. (2005). *Restoring the popularity of computer science.* Communications of the ACM, 48(9):25–28.

[12] Pinkerton, K. D. (1993). *A Rube Goldberg machine project for seniors.* Phys. Teach. 31, 468

[13] Rocard, M., Csermely, P., Jorde, D., Lenzen, D., Walberg-Henriksson, H., and Hemmo, V. (2007). *Science education now: a renewed pedagogy for the future of europe.* RTD info, EUR22845

[14] Savery, J. and Duffy, T. (1996). *Problem based learning: An instructional model and its constructivist framework.* Constructivist Learning Environments: Case Studies in Instructional Design. pp. 135–148. Educational Technology

[15] Sklar, E., Eguchi, A., Johnson, J. (2003). *RoboCupJunior: Learning with Educational Robotics.*, volume 2752 of Lecture Notes in Computer Science, pp. 238–253. Springer Berlin Heidelberg

[16] Visser, U. and Burkhard, H. (2007). *Robocup: 10 years of achievements and future challenges.* AAAI AI Magazine, 28, 2

[17] Weinberg, J., Xudong, Y. (2003). *Robotics in education: Low-cost platforms for teaching integrated systems.* Robotics & Automation Magazine, IEEE Volume:10,2. pp. 4–6

Developing a Planetary Rover with Students: Space Education at TU Berlin

Cem Avsar, Walter Frese, Thomas Meschede and Klaus Brieß
Department of Aeronautics and Astronautics
Chair of Space Technology
Technische Universität Berlin
Berlin, Germany

***Abstract*—Practical engineering education in space technology is a major challenge. TU Berlin has collected many years of experience in educating prospective space system engineers. With its history of developing and operating real space missions and its many ongoing research activities in the field, students are provided the opportunity to experience space technology. The Chair of Space Technology adresses the issue of innovative education constantly. Novel methods in hands-on education that arise through new technologies and trends are embedded into the curricula. A current project envisages the development of the autonomous planetary rover SEAR (Small Exploration Assistant Rover) with students. With SEAR, TU Berlin will participate in the SpaceBot Cup, a planetary exploration competition aimed to push forward key technologies in space robotics. To comply with the strict development timeline of nine months, educational methods for efficient utilization of student's ressources and provision of high-level education are applied.**

I. Introduction

Globalization, rapidly evolving technologies and shifts in demographics are changing the nature of engineering practice. Broader skills and a priority for application–driven engineering is demanded by students nowadays [1]. This creates new requirements for curricula at universities. With its origins in the 1980's, educational change in engineering is a relatively new field of research and many innovative changes are stand-alone and do not impact departmental and national practice [2]. Project-based courses have proven to be a suitable approach for engineering training and further help students in improving their teamwork and communication skills [3].

Since spacecraft design is a mature endeavor and the academic world is almost devoid with experience in the space industry, engineering in space technology is a major challenge for universities [4]. However, with miniaturization of technology and new trends in space engineering education, new opportunities arise for students and universities to become involved with practical space education.

The TU Berlin, Chair of Space Technology is experienced in spacecraft missions and devoted itself to offer high-class hands-on education for its students. By involving undergraduate students in real practical research activities during their curriculum, they are offered the best possible methods to graduate as aerospace engineers. An example of such a project is SEAR (Small Exploration Assistant Rover). It is the attempt to develop an autonomous planetary rover with mostly undergraduate students. Planetary exploration is an inherent part of astronautics and there are many parallels to terrestrial applications in robotics [5]. The SpaceBot Cup is an initiative launched by the German Aerospace Center (DLR) to push forward key technologies in space robotics. TU Berlin, as participant, uses the opportunity to utilize the experiences in methodological education in yet another lecture course.

II. Space Technology at TU Berlin

The TU Berlin, Chair of Space Technology is affiliated to the Department of Aeronautics and Astronautics. The focal point of the facility is on educating space system engineers to meet the demands of the future space market. By housing all facilities that are characteristic for a space technology site, reaching from laboratories, ground station and mission control center to testing facilities, students can work and learn in a professional environment.

The research focus of the Chair is set on miniaturization of technologies for small satellites. The research activities are spread over all segments of a spacecraft mission. TU Berlin has a long heritage in development and operation of satellites. Its complete satellite fleet with specification of masses and main payloads is displayed in figure 1. In a study of the statistical history of university–class small satellites, conducted by Swartwout [6] in 2011, TU Berlin was leading the list with seven launches into orbit.

The launch and therefore the mass of a spacecraft is one of the major cost drivers of a satellite mission [7]. More frequent launches of low priced satellites allow a faster return of science [8]. Small satellites open up potentials for universities and small businesses to conduct formation and constellation missions. TU Berlin is highly involved in developing miniaturized components for small satellites based on commercial off-the-shelf hardware. Further research activities encompass the development of launch vehicles, conduction of high-level technological studies and robotics for planetary exploration.

III. Space Education Practices at TU Berlin

The Chair of Space Technology offers a variety of space related lectures for Bachelor's and Master's students. The curriculum covers basic and advanced topics of spacecraft technology, mission design, operations and further topics in several lecture courses. The philosophy of the Chair is to bring space technology close to students by collecting practical experience. With small satellites emerging, it has become feasible to conduct a complete satellite mission with students. Practical lecture courses at TU Berlin are composed of several

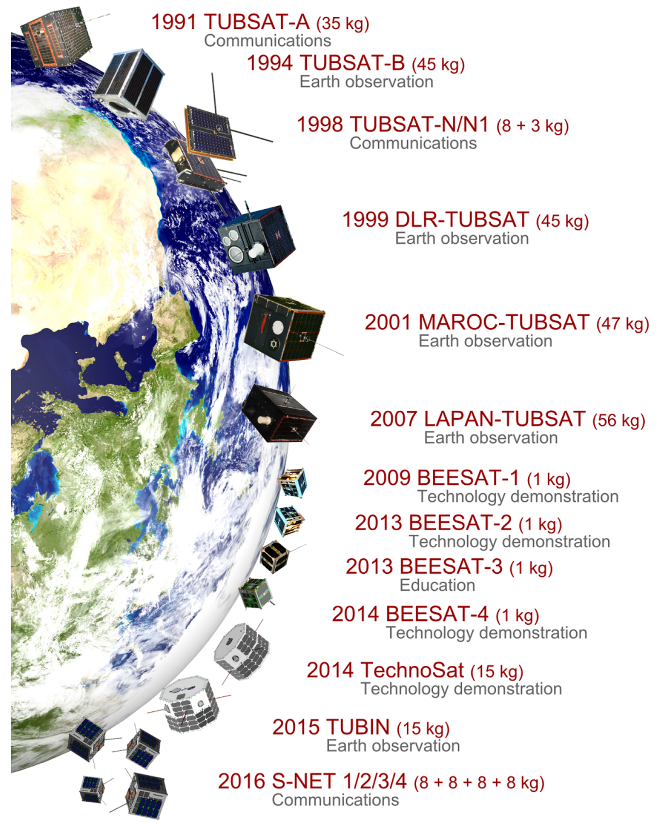


Fig. 1. TU Berlin's satellite fleet (Including past and future missions).

approaches for project based education. These emphasized to be mission design courses, space-related experiments and hands-on courses. In following, the education approaches shall be explained in detail with highlighting selected practical projects in context.

A. *Mission design courses*

Mission design courses have been widely conducted for many years at TU Berlin. The topic of a project is mostly a high-level technology spacecraft mission. The variety reaches from complex satellites and planetary rovers to Mars sample return missions.

The mission design courses are organized in a way that an almost real economy-like environment is created. This includes work processes according to ECSS (European Cooperation for Space Standardization) standards. The main mission goals and the high-level mission requirements for a project are given due to specific circumstances or by the course supervisor. The mission is broken down into several work packages that cover all aspects of the mission. These include e.g. the design of a satellite attitude control system, conduction of thermal analysis for a spacecraft or preparation of a mission operations schedule. Each work package is worked on by one or multiple students, depending on the estimated work load of the work packages.

The precise defintion of interfaces is of great importance. Therefore, the assignment of a systems engineering authority is indispensable. The main parameters are updated and discussed during several group meetings to initiate the next design iteration. Interim reports and presentations help to push the design process forward. All results are documented properly until end of semester. The documentation is distributed to several reviewers composed of researchers and experts of the facility. As closing of the course, a review according to ECSS standards is conducted, during which the students shortly present their results and have to respond to critical questions from the review board. The project is evaluated for each student by the performance in midterm and end presentation, as well as by the documentation.

With this course concept, a large group of students can be reached who, in the end of the course, will have a good understanding of the mission design process, a broad understanding of all mission aspects and a deepened knowledge in a specific discipline. Over 20 such mission design courses have been conducted at the Chair of Space Technology so far. Most of them showed promising results. It was found that students are more motivated when the course topic is conducted in context of a real mission. The students are stimulated by the huge responsibilities assigned to them.

B. *Space-related experiments*

Many components in TU Berlin's satellite fleet are developed in-house with the help of students. Space components development involves extensive functional and environmental verification processes. In the TUPEX (TU Berlin Picosatellite Experiments) series, students have the chance to participate in a sounding rocket campaign to test novel miniaturized components. The REXUS/BEXUS program allows students from Europe to fly an experimental payload on a high altitude balloon or a sounding rocket. It is sponsored by the German Aerospace Center (DLR) and the Swedish National Space Board (SNB) in collaboration with the European Space Agency (ESA). TU Berlin participated three times in a sounding rocket campaign with diverse experiments. Other space-related experiments involve flights in reduced gravity aircraft.

These kinds of activities address fewer students than mission design courses, but the intense education provided to a handful of students in conducting a real experiment, reflects in a very deep practical knowledge and experience at the end. In most cases, the students write their theses about the related project, since a sounding rocket campaign runs over a period of approximately one and a half years.

C. *Hands-on courses*

A high challenge lies in conducting an aerospace lecture, in which the goal is to develop real hardware in short time. The Chair of Space Technology is largely involved in this area and has collected a broad experience over the years. The lectures often desire a minimum set of requirements, e.g. basic knowledge of electronics, mechanics and satellite subsystems, which are all covered by previous courses at the Chair. Following, some examples for hands-on courses shall be given.

1) CanSats and sounding rockets: For space education, some particularly new concepts arose, for example with the CanSat programs. A CanSat is a satellite in shape of a standard soda can. An international class CanSat has to follow certain specifications. It shall have a maximum mass of 350 g and a cylinder shaped body with 66 mm in diameter and 115 mm height [9]. With this body, it fits into a rocket that usually carries a CanSat to an altitude of a few kilometers. It is then ejected in apogee and drops to ground safely with a parachute. A CanSat is not a real satellite, but it contains many downscaled subsystems that are pretty similar to those of an ordinay satellite. With costs of a few hundred dollars, a CanSat project provides an affordable opportunity for educators and students to acquire basic knowledge of space engineering and to experience engineering challenges in building a satellite [10]. Students can develop interesting missions like atmospheric measurement experiments with telemetry downlink or conduct more advanced missions like a GPS-based flyback.

Also, sounding rockets are developed at TU Berlin with strong involvement of stundents. Some examples are the hot water driven AQUARIUS and the rocket DECAN [11] that is designed to launch up to 15 km altitude and can carry one CanSat. Impressions of CanSat and sounding rocket projects at TU Berlin are shown in figure 2.

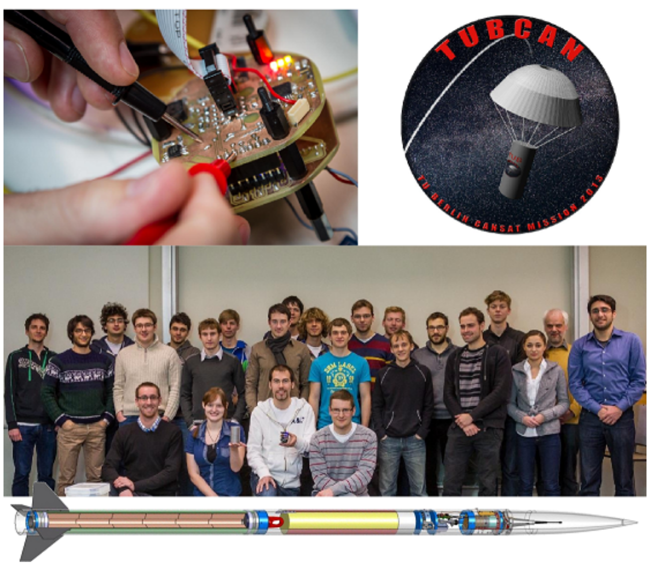


Fig. 2. Impressions of TU Berlin CanSat and sounding rocket projects.

2) CubeSats: Another, more sophisticated, approach lies in designing real satellite missions. With the CubeSat specifications, it has become feasible to build a satellite for comparably low costs. The specification states that the satellite shall be a cube with the size of 10 cm $\times$ 10 cm $\times$ 10 cm and a maximum mass of 1,33 kg [12]. There are small components, ejection mechanisms and launch providers available on the market for this standard. More and more universities worldwide are building CubeSats. Since TU Berlin was very successfull with the Berlin Experimental and Educational Satellite (BEESAT-1), which was launched in 2009 and is still operating, it had launched the initiative to build a completely new CubeSat, BEESAT-3, in lecture courses. The payload of BEESAT-3 is a new S-band transmitter for picosatellites, with downlink rates of 1 Mbit/s [13]. The phases of development were spread over four semesters and the students designed the subsystems and all other mission aspects. The works were supplemented by a group of researchers, volunteers, and students writing their theses.

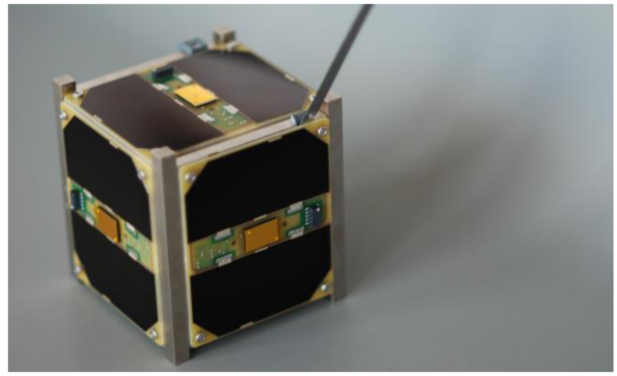


Fig. 3. BEESAT-3: CubeSat developed by students.

3) Rovers: Robots are part of space technology and thus moving in the focus of the Chair of Space Technology. They offer a great platform for developing hardware, electronics and software with results that can be observed almost immidiately. With the TU Berlin Robot (TUBROB) program, a small rover was built by students during several semester theses. TUBROB-2, a less complex rover was built during a lecture course on aerospace electronics. SEAR is a high-technology rover that will take part in the DLR SpaceBot Cup. Major parts of the rover are developed in a lecture course titled ”Planetary Exploration and Space Robotics”.

With these kinds of projects, the students get a very deep knowledge in a specific subsystem. Due to the high motivation enforced by taking part in a real mission, students are very commited to the project. Nonetheless, it was found that a test on general subjects related to the project is a good way to maintain that students pay high attention when all topics are discussed in the group.

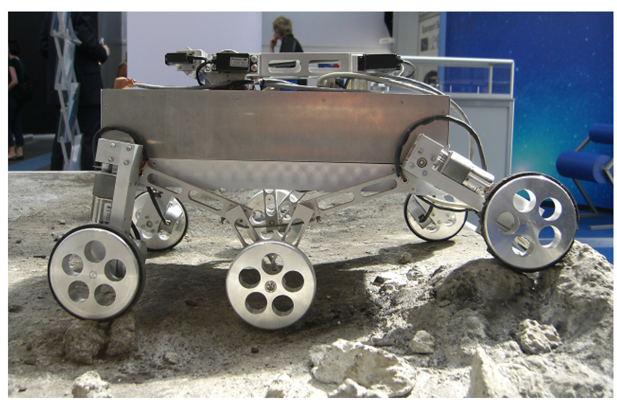


Fig. 4. TUBROB: TU Berlin educational rover.

IV. The DLR SpaceBot Cup

Space robotics has gained more and more attention in Germany since 2009 [14]. It is now in the focus of the national programme for space and innovation. It combines technologies from different disciplines and is a driver for cutting edge technologies that can be utilized in space and on Earth.

The DLR SpaceBot Cup is a space robotics contest organized by the German Aerospace Center (DLR) and funded by the Federal Ministry of Economics and Technology (BMWi).

With this contest, the advancement of key technologies in space robotics shall be pushed forward. The setting for the contest is a planetary exploration scenario. Competitors from universities and small companies in Germany have been called to take part in the contest. Out of all applicants, ten candidates have been selected to build one or more robots that shall achieve all predefined tasks. On an area of 36 m × 28 m that represents a planetary surface including rocks, sand and acclivities, the robots have to fulfill several tasks that are typical for a future planetary exploration scenario. There are several constraints for the contestants and the overall design time of nine months is very short. A special focus in this contest is set on aspects of autonomy. The robots have to perform all tasks autonomously. Therefore, e.g. no crew member in the control room can have visual contact to the robot(s) during the contest course.

Constraints

- The time on the planetary surface is limited to 1 h per contestant.
- The number of robots per contestant is not limited.
- The total mass that can be dropped onto the planetary surface cannot exceed 100 kg.
- The use of GPS is prohibited.
- A rough map of the environment with elevation profile will be supplied four weeks prior to the contest.
- Each team can have three contacts for five minutes with the rover, for telemetry download and software update. Steering of the rover is prohibited.
- Contact with the rover can only be established from inside of a control room from which the planetary surface is not visible.
- There will be latencies of a couple of seconds when communicating with the robot(s).
- Inclinations of 15 degrees have to be overcome.

Tasks

In general, there are three categories of tasks to be absolved. The main success criteria for winning the contest is time. For any more contacts needed, interventions and not achieving tasks, this will result in addition of penalty minutes.

A. *Locating and identifying objects*

The robot has to allocate itself in an unknown area. Three predefined objects that will be distributed randomly over the area have to be found, identifyied and mapped. The objects, as seen in figure 5, consist of a battery pack, a glass filled with water and a base station. All objects will have a distinctive color. The battery pack has a size of 10 cm × 20 cm × 4 cm and a mass of 1 kg. The glass has a diameter of 8 cm, a height of 12 cm and a mass of 1 kg. The base station has a size of 20 cm × 40 cm × 20 cm.

B. *Planning and Transport*

Two objects, the battery pack and the glass filled with water, have to be grasped and transported to the base station. Therfore, an optimized route through the terrain has to be calculated.

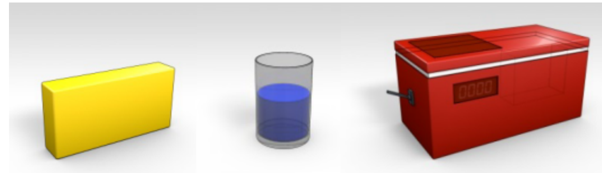


Fig. 5. Objects to be located and identified (Source: DLR).

C. *Obstacle avoidance and assembly*

On the drive back to the base station with the objects, an additional obstacle has to avoided that will be put in the way of the robot(s) nominal path. With arrival at the base station, the battery pack has to be mounted to it. A contact will verify successfull mounting. The glass filled with water has to be placed on top of the base station. The mass of water will be measured automatically to figure out how much was spilled on the way. To complete the course, a switch on the base station has to be triggered.

V. Design of SEAR

The requirements for SEAR are derived from the tasks it has to achieve. SEAR is a conventional rover. The concept was prefered over crawlers and other types of robots, due to its simplicity and stability in basic state. Figure 6 shows a drawing of SEAR with its body, undercarriage and manipulator.

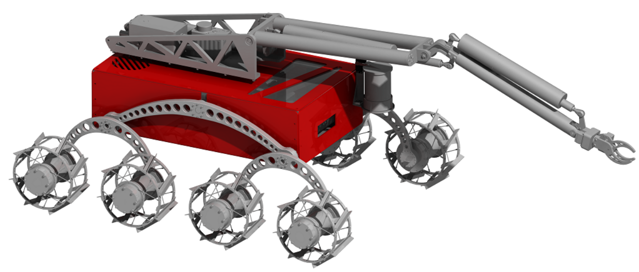


Fig. 6. SEAR assembly.

SEAR is still under development. Following are some current design parameters. SEAR has a width of 860 mm. When the manipulator is driven to its full extent, SEAR can reach a length of 1950 mm with a height of 670 mm. The overall mass is estimated with 60 kg. With a battery capacity of 20 Ah SEAR is powered for at least two hours in full operational mode.

A. *Core components*

The core of SEAR consists of on-board electronics for power distribution and computer processing. The power system makes use of a 24 V 20 Ah $LiFePO_4$ battery block and DC/DC-converters for generating the 12 V and 5 V bus voltages. A set of eight motors for locomotion are connected to the battery. A manipulator on top of the rover is used to grab, store and mount objects. Due to the short development timeline, the hardware concept is kept relatively simple. Most of the components are plugged and merged to a central processing unit. The main sensors are three Kinect cameras for 3D vision. One is mounted onto the manipulator, one at the front of the chassis and one at the back. An intertial measurement unit (IMU) is used supplementary for several purposes in location, navigation and manipulation tasks. Data from the wheel motion controllers is additionally used for odometry. For connecting small devices like infrared range sensors for measuring distance to closely located objects or servos for storing mechanisms, a supplementary microcontroller board is being used. Simple commands can be sent to the board by the main central processing unit. Several electronics components can easily be attached to the board if necessary during late phases of development.

B. Locomotion

SEAR belongs to the class of wheeled robots for planetary exploration. The drivetrain consists of eight wheels passively suspended on a Rocker-bogie system as shown in figure 7. Both rockers are connected to each other and to the chassis through an averaging mechanism.

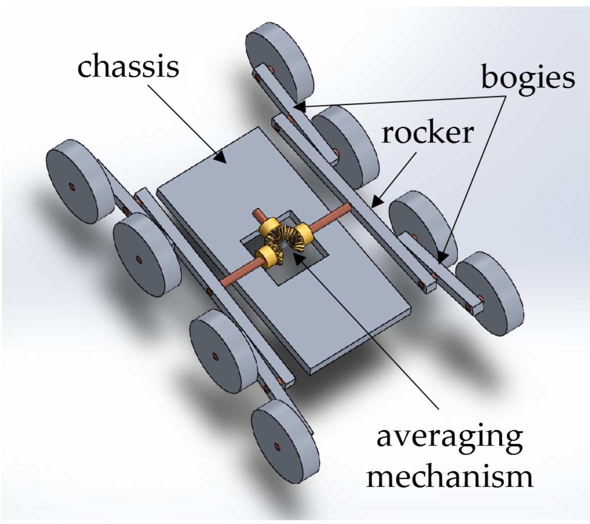
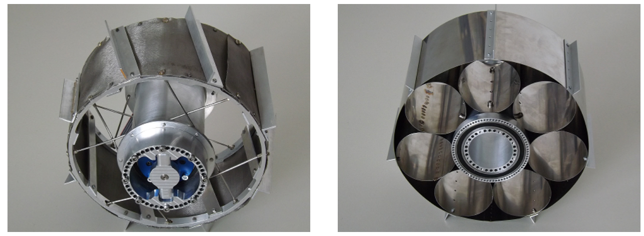


Fig. 7. SEAR's eight-wheeled Rocker-bogie locomotion system concept.

By adding bogies to each end of the rocker arm, two wheels on one end of the rocker can be suspended from two on the other end. The Rocker-bogie suspension system shown in figure 7 produces no asymmetrical loads on the wheels. The Rocker-bogie's big advantage is that it can negotiate obstacles that are twice the wheel height [15]. Each wheel is passively loaded by the Rocker-bogie suspension increasing traction greatly compared to a layout with eight wheels simply attached to the chassis.

The Rocker-bogie suspension is skid steered, which reduces the average system complexity. Compared to the most common six-wheeled concept with four steering motors for the front and the rear wheels and six wheel motors (as used in four Mars rovers since Sojourner developed by NASA), which is proposed to be the system with the highest mobility, the SEAR concept needs eight motors only. The rockers and the bogies are designed to withstand the side moments produced during skid steering.

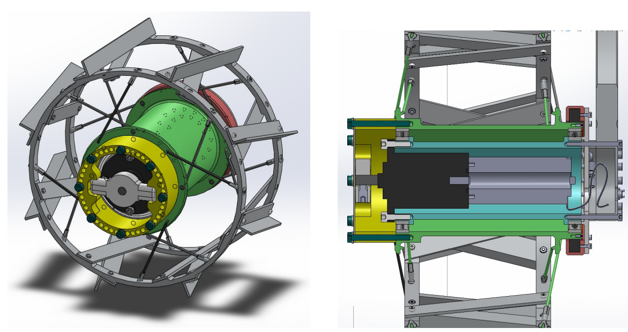
As mentioned above, a wheel based chassis was selected and constructed according to the formula 8 × 8. Two different types of wheels had been investigated, each with an



(a) Rigid wheel. (b) Flexible wheel (concept rejected).

Fig. 8. Two wheel prototypes for SEAR manufactured at the TU Berlin, Aerospace Department's workshop.

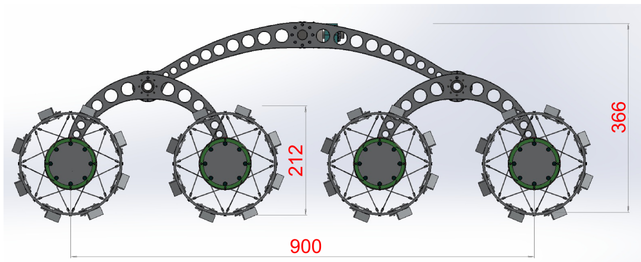
outer diameter of approx. 20 cm and the width of the total supporting surface of 100 mm (see Figure 8). A metal-elastic wheel showed a very good soil traction performance in a theoretical investigation. In the end, the flexible wheel design was rejected due to several difficulties in manufacturing it at the department's workshop. After a redesign, a rigid wheel with spikes was chosen as shown in figure 8(a). The supporting surface is formed by a steel mesh. The mesh is spanned between two rings machined from an aluminum alloy using outer and inner hoops. For the case of the rover driving across soft soil, the supporting surface was designed larger. Each of the aluminum rings is connected via eight steel spokes to a hub made from aluminum alloy. Inside the hub a commercial-off-the-shelf brushless motor with a gear box is used. The wave of the gear box is decoupled from axial and radial forces via a coupling shaft (see Figure 9(b)). The main axial and radial forces between the rotor and the stator are contained by two dedicated ball bearings and not only by the ball bearings of the gear box. Each of the motor-wheels is controlled individually by a wheel drive electronics implementing amongst others wheel velocity and momentum-of-rotation control loops. The dissipated heat is transferred from the motor to the bogie via a thermal interface composed of several heat conductors made of copper.



(a) Isometric view. (b) Cut through the motor-wheel.

Fig. 9. CAD design of SEAR's spiked wheel.

The rockers and the bogies are machined from an aluminum alloy. Each of the bogies with the length of approx. 30 cm carries two wheels and is connected to the rocker (approx. 70 cm) via ball bearings exactly in its middle as shown in figure 10. An averaging mechanism consisting of three ball-beared gear-wheels is attached to the chassis of the rover. The rocker and the bogies are curved in order to achieve a certain relation of the center of gravity and the point where the forces from the rocker are coupled into the main body of the rover. The suspension mechanism, the motor-wheels and the averaging mechanism together achieve a total mass of approx. 17 kg wich can be reduced by decreasing safety factors using advanced analysis methods and by utilising materials with better mechanical properties.



(a) Side view.

Fig. 10. Side view on SEAR's drivetrain (Dimensions in mm).

C. Software architecture

Selecting a robotics software framework is a mandatory task, if one wants to use as much preexisting knowledge and technology as possible. Among several options [16], [17], [18], [19] the Robot Operating System (ROS) [20] as main framework was selected, because of its built-in publish-subscribe mechanism, a thriving community [21] and a detailed documentation together with an open source license. ROS' use in several well-known research projects [22], [23] is just a small hint to the leading-edge software paradigms and application programmin interface (API) of the framework. This also makes it an ideal choice for education purposes as students get a deeper insight on the inner workings of a robot. The publish-subscribe nature of ROS on the other hand enables it to easily distribute key development tasks among the students. For this purpose at first key tasks have been identified which could easily be put into separate so-called ROS packages. For many of these tasks there are already standard packages existing in the ROS framework which are adapted to work with SEAR.

D. Task planner

Task planning is organized into a hierarchy which resembles the corresponding high-level and low-level routines of the rover. While low-level routines are beeing carried out by individual ROS packages, a finite state automata is used to generate the high-level commands and propagate them down the hierarchy.

E. Simultanous Location And Mapping (SLAM) and path planning

Mapping of the rover's environment and simultanous location is beeing provided by the navigation stack of ROS which is well integrated with a commercial Kinect sensor. This sensor is a widely used sensor among robotics projects and generates point cloud data from the environment. The navigation nodes then use this data to calculate the position of the sensor within the surroundings by correlating geometric features from the map data with those in the point cloud. While the original map generated from the point cloud is a 3D map, the path planning algorithm provided by ROS requires a 2D pixelmap (figure 11). The 3D data is used and converted into the required format by representing regions with slopes too steep to be handled by SEAR as obstacles. The path planning algorithm first tries to find a way through obstacles without taking details into account. This is called the global path. In a next step the local planner takes care of the details like steering and turning radius. For this purpose a detailed model of the rover provides the necessary parameters.

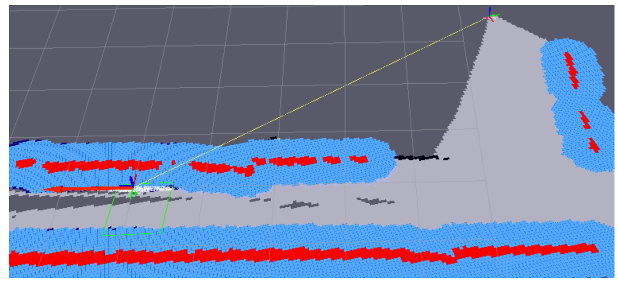


Fig. 11. 2D pixelmap of the global cost map calculated from the point cloud data of the Kinect sensor.

F. Object recognition

Searching for the required objects, identifying them and subsequently measuring range and orientation is beeing done with the widely used OpenCV library [24] coupled with the Kinect sensor. The algorithm primarily uses color information from a video image, as the target objects have a specified color in contrast to a rather monotonous environment.

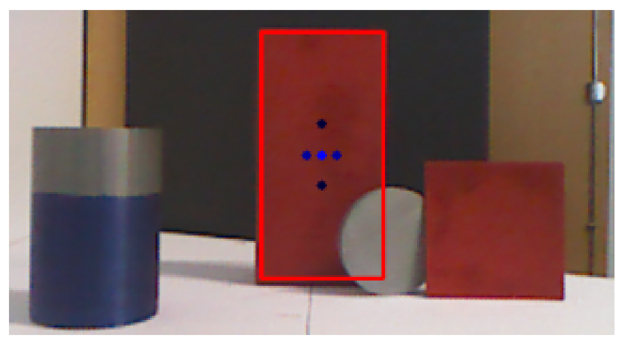


Fig. 12. Recognition of the battery pack prototype by using Kinect sensor data.

G. Manipulation

In total, the manipulator has six degrees of freedom. It is able to reach for objects around the rover. A stereoscopic camera on the manipulator allows a 360 degrees observation without having to rotate the rover itself. A special grasping mechanism has to be developed to grab the specified objects. An additional challenge is the transportation of the glass of water. Control algorithms are also provided by the ROS manipulation planning stack. It relies on a generic gradient-based numeric algorithm for inverse kinematics. Trajectory planning for collision avoidance works by taking the point cloud into account, which is provided by one of the Kinect sensors (figure 13).

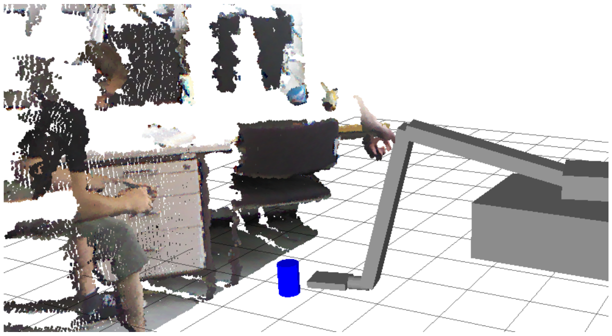
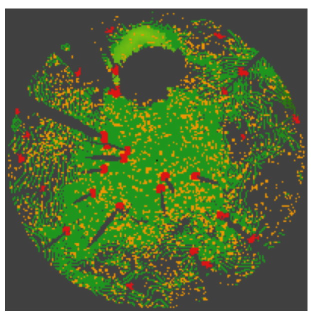


Fig. 13. Simulation of manipulator trajectories utilizing the point cloud for collision avoidance.

H. Navigation

The navigation is based on input from a COTS stereoscopic camera. The Kinect camera has proven to be a good solution in many robotics projects around the globe. A colored scatter-plot allows visual navigation like shown in figure 14. It is made use of the open-source framework ROS in combination with image processing libraries like OpenCV and for SLAM algorithms. Besides visual odometry with the stereoscopic camera, data from wheels and an inertial measurement unit in fusion shall allow a more accurate position determination.

(a) Terrain view.



(b) Local environment view.

Fig. 14. Principles of SEAR's exploration strategy. (a) Position of rover within terrain. (b) Local environment based on simulated sensor data.

VI. SEAR IN EDUCATION

SEAR is a project that is mainly conducted with undergraduate students under supervision of staff members. SEAR is put together of a core team consisting of three permanent staff members who are mainly involved in other projects and three undergraduate student assistants solely dedicated to the project. The staff members are each responsible for supervising aspects of project management, mechatronics and software. The student assistants are provided with a work environment and facilities. Having a core team has proven to be a critical success factor for the good advancement of the project. Much groundwork had to be layed by the core team in terms of setting up a programming environment and defining interfaces in order to make it possible for over 20 students to write software in parallel. For example, an Ubuntu image was created that encovered the ROS environment and all important packages for a kick-start in a uniform working environment. A two-layered hierarchical structure was also found out to be helpful for increasing decision makings inside the project.

During summer term 2013, aspects of the rover development are covered by students in project work in the lecture ”Planetary Exploration and Space Robotics”. Students receive 6 ECTS credits for the course which consists of a weekly theoretical lecture and project work. With a total number of 20 students there is huge manpower that can efficiently be utilized within three months. The members of the student group have a different background. Most are Master's and Bachelor's students from the aerospace engineering study course between the ages of 20 to 25. One fourth are computer engineering students. The rest is coming from related engineering study courses. Most of the students have a basic knowledge in programming. The interdisciplinary group emerged to be a great basis for distributing tasks in the areas of mechanics, electronics and software. By using the programming language Python, even students with little understanding of programming managed to write sophisticated code. A strong support by group members, good introductary tutorials and the well-documented robotics framework made a fast and efficient software development possible.

High attention was payed in designing the course structure. During the first week, the students were introduced to the contest and the rough rover concept. Like in industry, a job catalogue provided an overview of available positions within the project. The job description file was developed in brainstorming and mind mapping sessions with the SEAR core team. In general, it is a challenge to find self-containted work packages with an appropriate and equally spreaded work load. Around 15 jobs for up to two persons each were to be found in the document. Following, a typical job description for each a mechanical, electrical and software engineer is listed.

JOB DESCRIPTIONS

Mechanical engineer for conception of a grasping mechanism
In the contest, two specific objects shall be grasped by the rover. Therefore, a concept for the grasping procedure has to be developed. A gripper, sensors for touch feedback and prototypes of the objects shall be developed. Tests shall be conducted.
Tasks until course date 4: - *Evaluation of available grippers for specific objects.*

Electrical engineer for design of on-board electronics
All components on the rover including on-board computer, motors, sensors, actuators etc. have to be supplied with electrical energy. An appropriate energy storage has to be alloted that can provide all systems with energy for at least one hour. For all components, appropriate voltages have to be conditioned. For some electronics, adapter circuits have to be designed.
Tasks until course date 4: - *Presentation of a block diagram that shows all electrical components, their properties and interfaces.*

Software engineer for image-based object identification

During the contest, three objects on the terrain have to be located and identified. Therefore, image-processing algorithms have to be assembled and adapted. With test settings and object prototypes, the algorithms shall be proven.

Tasks until course date 4:
- *Learning basics of robot operating system by doing the tutorial.*

For a rapid start of the project, in each work package a specific initial task was presented. The task had to be done withing four weeks after the students had been allocated to their work packages. The allocation process was done as follows. After students received the jobs description document, they had to fill out a one page application form within one week. Desired position, application writing, curriculum vitae and three alternative positions were necessary information to be delivered by the students. By carefully evaluating the application forms, the jobs were assigned fairly.

The course structure is very promising for the students to make a good engineering education experience. The course is evaluated by project work and a test on general knowledge about planetary exploration and space robotics. Withing 10 weeks the students finished the mechanical design of the rover and had written functioning software parts covering all aspects of the mission. They won deep practical insight in at least one specific part of developing mechanics and software for a robot. Through the strong collaboration, weekly meetings and laboratory work hours they also gained an insight into all other aspects of a robot. During a period of approx. two months for manufacturing, the software parts will be extended and tested seperately. After assembly, complete system tests will be conducted on a 5 m $\times$ 7 m planetary surface testbed.

VII. Conclusion

Although, practical space engineering education is a major challenge, several approaches for a successfull implementation of a practical curriculum have been presented. TU Berlin successfully excercised several approaches like mission design courses, space-related experiments and hands-on courses. Over the years, it has been found that students are highly motivated when they are assigned responsibilities within a project with real impact. Not only students can benefit from a real project experience in education, but also the Chair can benefit from a well-organized manpower in lecture courses to push forward important projects.

Acknowledgment

The SEAR project is funded by the German Aerospace Center (DLR) with funds from the Federal Ministry of Economics and Technology (BMWi) on the basis of a decision of the German Bundestag (Grant No.: 50RA1314).

Special thanks go to all students that allowed the implementation and successfull completion of practical lecture courses at TU Berlin.

References

[1] J. J. Duderstadt, "Engineering for a chaning world," The University of Michigan, Ann Arbor, MI 48109-2094, Tech. Rep., 2008.

[2] R. Graham, "Achieving excellence in engineering education: the ingredients of succesful change," The Royal Academy of Engineering, 3 Carlton House Terrace, London SW1Y 5DG, Tech. Rep., March 2012.

[3] C. L. D. et al., "Engineering design thinking, teaching, and learning," *Journal of Engineering Education*, vol. 94, pp. 103–120, 2005.

[4] M. Gruntman, "The time for academic departments in astronautical engineering," in *Proceedings of the AIAA SPACE 2007 Conference & Exposition*, Long Beach, California, USA, September 2007.

[5] T. Ltd., "Space exploration and innovation," Brighton, United Kingdom, Tech. Rep., October 2010.

[6] M. Swartwout, "Attack of the cubesat: A statistical look," in *Proceedings of the 25th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, USA, August 2011.

[7] J. V. B. et al., "Affordable launcher for small spacecraft missions," in *Proceedings of the 19th AIAA International Communications Satellites Systems Conference*, Toulouse, France, April 2001.

[8] M. B. et al., "A survey of small satellites domain: Challenges, applications and communications key issues," *ICST'S Global Community Magazine*, 2010.

[9] *French CanSat Competition Rules, 5th edition*, CNES, 2011.

[10] H. S. et al., "Cansat leader training pogram (cltp) report," Tokyo Metropolitan University, Tokyo, Japan, Tech. Rep., July 2010.

[11] K. L. et al., "Stern - raketenprogramm für studenten," in *Deutscher Luft- und Raumfahrtkongress 2012*, Berlin, Germany, September 2012.

[12] *CubeSat Design Specification Rev. 12*, California Polytech State University, 2009.

[13] M. F. Barschke, "Beesat-3: A picosatellite developed by students," in *Proceedings of the 61st Annual Deutscher Luft- und Raumfahrtkongress*, Berlin, Germany, September 2012.

[14] B. für Wirtschaft und Technologie, "Schlaglichter der wirtschaftspolitik, monatsbericht mai 2012," Tech. Rep., May 2012.

[15] P. E. Sandin, *Robot Mechanisms and Mechanical Devices Illustrated*. The McGraw-Hill Companies, Inc., 2003.

[16] J. Jackson, "Microsoft robotics studio: A technical introduction," *Robotics & Automation Magazine, IEEE*, vol. 14, pp. 82–87, December 2007.

[17] A. Makarenko, "Orca: Components for robotics," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006.

[18] H. Bruyninckx, "Open robot control software: the orocos project," in *Proceedings of the 2001 ICRA. IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001.

[19] R. D. et al., "Openrave: A planning architecture for autonomous robotics," no. CMU-RI-TR-08-34, July 2008.

[20] M. Q. et al., "Ros: an open-source robot operating system," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[21] S. Cousins, "Exponential growth of ros," *Robotics & Automation Magazine, IEEE*, vol. 18, pp. 19–20, March 2011.

[22] T. Krüger, "Exoskeleton control of the robonaut through rapid and ros," in *Proceedings of the 12th Symposium on Advanced Space Technologies in Robotics and Automation*, Noordwijik, Netherlands, May 2013.

[23] S. Cousins, "Ros on the pr2," *Robotics & Automation Magazine, IEEE*, vol. 17, pp. 23–25, September 2010.

[24] G. Bradski, "The opencv library," *Doctor Dobbs Journal*, 2000. [Online]. Available: http://elibrary.ru/item.asp?id=4934581

Simulation and Assessment Educational Framework for Mobile Robot Algorithms

Nelson Munoz Ceballos[1,2], Jaime Valencia[3], Alessa Alvarez Giraldo[1,3]
[1] *Colombian Politechnic Jaime Isaza Cadavid,*
[2]*Metropolitan Technological Institute*
[3]*University of Antioquia*
Medellín-Colombia
E-mail: nmunoz345@gmail.com

***Abstract*--A mobile robot simulator useful in research and education was implemented in Matlab, it models the differential kinematics as well as proximity sensors of the robot. It allows the performance assessment of navigation algorithms through various quality metrics that are useful for comparing and analyzing navigation algorithms of mobile robots. An example simulating and comparing two autonomous navigation algorithms is presented.**

Keywords-- Educational robotics, Mobile robots, Navigation algorithms, Performance metrics.

INTRODUCTION

The development of a robust autonomous navigation system for mobile robots is a broadly studied topic, and an open field for research. Those systems are continuously evolving and new approaches and applications are constantly emerging.

Different ways to address the robot navigation arise frequently, each of them with new valuable contributions. As time goes on, the problem and its possible solutions are better understood according to the specific application.

Many methods have been used just to solve specific problems while their strengths or weaknesses are not completely understood. The comparison of algorithms with reference frames or standard procedures is usually a relegated task [1], [2].

This paper presents a framework for simulation and assessment of mobile robotics navigation algorithms, useful for teaching and research in robotics. In section 1, the simulator is described, in section 2, various performance metrics used in the navigation of mobile robots are defined, in section 3, example of two navigation algorithms are presented, in section 4, the process to be followed for assessment of algorithms is showed. Finally, in section 5, some conclusions are presented.

1. MOBILE ROBOT SIMULATOR

A graphic 2D simulator has been developed, it is useful for teaching and researching on navigation algorithms for mobile robots, it offers the possibility to evaluate the performance of the implemented navigation technique. The framework was carried out using Matlab, chosen for its potency to create mathematic programs, its easy use and the possibility of adding toolboxes as neural networks, fuzzy logic, etc.

The mobile robot simulator allows creating an environment, a robot, the environment obstacles as well as algorithms that process information of the robot state and act upon it. Furthermore, the framework includes a set of performance metrics [3], [14].

All the tasks in the simulator are carried out using graphical user interface and commands as displayed in figure 1. The user can generate the graphic environment by programming and design the own functions using the basic commands.

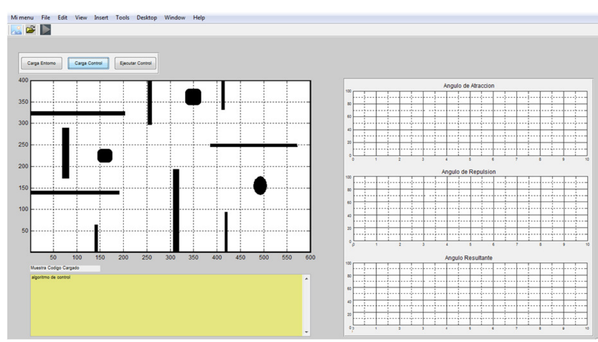


Fig. 1. Simulator interface.

1.1 Robot environment

The environment is created in graphic edition software and uploaded in the simulator as an image. The white areas are understood as empty while the black areas are taken in as objects in the environment. The image boundaries are taken as walls.

1.2 Robot movement

In order to simulate the robot movement, its kinematics should be taken into account which is subject to the next equations for a differential locomotive robot:

$$\dot{x} = vCos\theta$$
$$\dot{y} = vSin\theta$$
$$\dot{\theta} = w$$

Where $\dot{x}$ and $\dot{y}$ are the speed in the axes x and y; θ is the angle of the robot with the axis x; v, w are the linear and angular speed of the robot (movement and spin speed respectively). The linear and angular speeds are yielded by the navigation algorithm.

It is necessary to break down the previous expressions in differential equations to allow the computational estimate. In each sampling period T, the new x and y position regarding the center of the robot is calculated as well as its orientation, then the robot is drawn in that position.

1.3 Robot sensors

Proximity sensors should be defined by the user and the amount can be set according to the user needs. These can be set up according to the type of sensor the user needs to simulate, indicating their position in the robot periphery, the opening angle and the scope of the sensor (figure 2).

The distance measure is given in pixels and it is estimated taking the length between the point where the sensor is located and the closest point of any object within its detection scope.

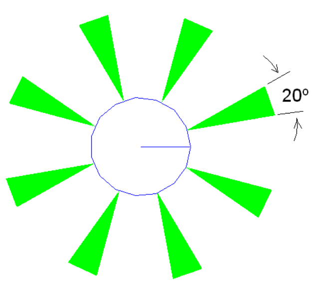


Fig.2. Proximity sensors (opening angle: 20º, detection scope: 25 cm)

2. PERFORMANCE METRICS ON NAVIGATION

There are various metrics that can be used to evaluate the performance of a navigation system, but none of them is able to indicate the quality of the whole system. Therefore it is necessary to use a combination of different indexes quantifying different aspects of the system. Having a good range of performance measurements is useful for: Optimizing algorithm parameters, testing navigation performance within a variety of work environments, making a quantitative comparison between algorithms, supporting algorithm development and helping with decisions about the adjustments required for a variety of aspects involved in system performance [13].

Navigation performance metrics can be classified in the following order of importance: Security in the trajectory indexes or proximity to obstacles, metrics that consider the trajectory towards the goal and metrics that evaluate the smoothness of the trajectory.

2.1 Security metrics

These metrics express the robot security while it travels through a trajectory, taking into account the distance between the vehicle and the obstacles in its path [5].

Security Metric-1 (SM1): Mean distance between the vehicle and the obstacles through the entire mission measured by all the sensors; the maximum value will be produced in an obstacle free environment. If the deviation of the index from its maximum value is low, it means that the chosen route had fewer obstacles.

Security Metric-2 (SM2): Mean minimum distance to obstacles. This is taken from the average of the lowest value of the n sensors. This index gives an idea of the risk taken through the entire mission, in terms of the proximity to an obstacle. In an obstacles free environment SM1 = SM2 is satisfied.

Minimum Distance (Min): Minimum distance between any sensor and any obstacle through the entire trajectory. This index measures the maximum risk taken throughout the entire mission.

2.2 Dimension metrics

The trajectory towards the goal is considered in its time and space dimensions. In general, it is assumed that an optimal trajectory towards the goal is, whenever possible, a line with minimum length and zero curvature between the initial point (x_i,y_i) and the finishing point (x_n,y_n), covered in the minimum time.

Length of the Covered Trajectory (P_L) is the length of the entire covered path by the vehicle from the initial point to the goal. For a trajectory in the x-y plane, composed of n points, and assuming the initial point as $(x_1, f(x_1))$ and the goal as $(x_n, f(x_n))$, P_L can be calculated as:

$$P_L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (f(x_{i+1}) - f(x_i))^2} \quad (1)$$

Where $(x_i, f(x_i))$, i = 1, 2, . . . , n are the n points of the trajectory in cartesian coordinates [6].

The length of a trajectory given by y = f(x), in the x-y plane between the points (a, f(a)) and (b, f(b)), can also be calculated as [10]

$$P_{Laprox} \cong \int_a^b \sqrt{1+(f'(x_i))^2}\,dx \quad (2)$$

Mean distance to the goal (Mgd): This metric can be applied to robots capable of following reference trajectories. An important aspect when determining the quality of the robot navigation system is the ability to follow a trajectory that aims to reach a goal, so, to evaluate the quality of the execution of the trajectory, the mean distance between the vehicle and goal is analyzed. The difference is more significant if the covered distance is shorter [9]. The mean distance to the goal is defined by the square of the proximity to the goal distance l_n, integrated across the length of the trajectory and normalized by the total number of points n:

$$l_n = \min\left(\forall n\left(\sqrt{(x_i - x_n)^2 + (f(x_i) - f(x_n))^2}\right)\right) \quad (3)$$

$$Mgd = \frac{\int_0^l l_n^2\,ds}{n} \quad (4)$$

Control Periods (LeM): It is the amount of control periods. This metric relates to the number of decisions taken by the planner to reach the goal, if the robot moves with lineal and constant speed (v). This gives an idea of the time needed to complete the mission [5].

2.3 Smoothness metrics

The smoothness of a trajectory shows the consistency between the decision-action relationship taken by the navigation system, as well as the ability to anticipate and to respond to events with enough speed [9]. The smoothness of the generated trajectory is a measure of the energy and time requirements for the movement; a smooth trajectory translates into energy and time savings [4]. Additionally, a smooth trajectory is also beneficial to the mechanical structure of the vehicle.

Bending Energy (B_E): This is a function of the curvature, k, used to evaluate the smoothness of the robot's movement. For curves in the x-y plane, the curvature, k, at any point $(x_i, f(x_i))$ across a trajectory is given by:

$$k(x_i, f(x_i)) = \frac{f''(x_i)}{(1+(f'(x_i))^2)^{\frac{3}{2}}} \quad (5)$$

The bending energy can be understood as the energy needed to bend a rod to the desired shape [11]. B_E can be calculated as the sum of the squares of the curvature at each point of the line $k(x_i, f(x_i))$, along the length of the line L. So, the bending energy of the trajectory of a robot is given by:

$$B_E = \frac{1}{n}\sum_{i=1}^{n} k^2(x_i, f(x_i)) \quad (6)$$

Where $k(x_i, f(x_i))$ is the curvature at each point of the trajectory of the robot and *n* is the number of points in the trajectory.

The value of B_E is an average and does not show with clarity enough that some trajectories are longer than others. Therefore, TB_E can be used instead; this metric takes into account the smoothness and length of the trajectory simultaneously.

TB_E is defined by $TB_E = \int_a^b k^2(x)dx$ (7)

and numerically, $TB_E = \sum_{i=1}^{n} k^2(x_i, f(x_i))$ (8)

In a straighter trajectory, the values B_E and TB_E will be lower, which is desirable since the energy requirement is increased according to the increase in the curvature of the trajectory.

Smoothness of Curvature (Smoo) is defined by the square of the change in the curvature k of the trajectory of a vehicle with respect to the time, integrating along the length of the trajectory and normalized by the total time t [9].

$$Smoo = \frac{\int_0^l \left(\frac{dk}{dt}\right)^2 ds}{t} \quad (9)$$

3. NAVIGATION ALGORITHMS

The navigation algorithms provide basic capabilities for the mobile robot, such as the ability to evade obstacles and to generate a trajectory towards a goal. (goal-seeking obstacle-avoidance)

3.1 Algorithm 1

This is a reactive algorithm based on a potential field method, which produces two different behaviors: first, goal attraction, and second, obstacles repulsion (keeping away from objects). The planning of the movement consists in the proper combination of both behaviors in such a way that the robot reaches the goal without collisions. This combination is achieved using a vector sum [7].

3.2 Algorithm 2

This algorithm is based on reactive behaviors, denominated AFREB "adaptive fusion of reactive behaviors" [12]. By using a neural network, an appropriate combination of the behaviors can be achieved, so that the system is able to perform complex tasks, such as navigation towards a goal, while evading obstacles in its path. The AFREB basically consists of the following modules: behavioral fusion, fusion supervisor, behavior primitives (1, 2, … n), and executor.

4. SIMULATIONS AND RESULTS

The simulation framework for comparing the performance of the algorithm 1 and algorithm 2 was used. This software enables teaching and researching in mobile robot navigation.

The robot simulated is Giraa02 [8], it has a differential locomotion system, 8 proximity sensors and odometry sensors; its diameter is 30cm.

As didactic example two different scenarios are used to test algorithms. The environment is similar to offices, it means, A 6m x 4m frame, structured environment with static obstacles, some obstacle borders are sharp, also, there are straight lines obstacles, and narrow zones, figure 3.

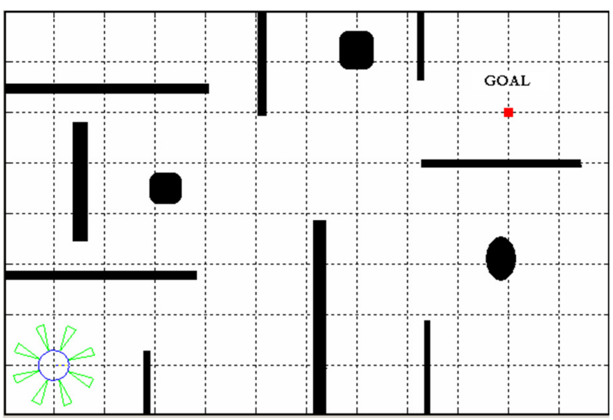


Fig. 3.Test Scenario for Mobile Robot Navigation

The metrics for autonomous navigation considers the security of the trajectory and measures the risk taken by the robot in its movement towards the goal, similarly measure aspects related to the planning of the trajectory and the quality of the trajectory according to the energy and time required for the movement.

For general purposes, only one metric is required for each one of the 3 categories described in section 2, but the use of various metrics helps to improve the analysis.

4.1 Simulations

The paths generated by the algorithms, in all scenarios are shown in figure 4 and 5. Table 1 summarizes the results obtained from the simulation using both navigations algorithms according to the quality metrics described.

4.2 Analysis of results

In scenario 1, the algorithm 1 uses less control periods, and consequently takes less time to complete the mission, and covers a safer and shorter path, the figure 4 shows that algorithm 1 produces a great orientation change for each control period. Algorithm 2 covers a smoother path, there is a smaller change in the orientation during each control period, resulting in energy saving and less structural stress on the robot.

From table 1, it can be deduced that the difference between both algorithms in the trajectory and time taken is approximately 3.3% and 3.1% respectively. The robot programmed with algorithm 2 passed at minimum 7 cm from any obstacle, it showed approximately 65% less bending energy than algorithm 1.

In scenario 2, the algorithm 1 uses more control periods, and consequently takes more time to complete the mission. Its covers a safer and longer path, the figure 5 shows that algorithm 2 covers a smoother path, there is a smaller change in the orientation during each control period, with consequent energy saving and less structural stress on the robot. Algorithm 2, makes the robot able to transit through narrow zones like corridors, keeping a safe distance from the obstacles and also generating smooth trajectories.

These results are an example to demonstrate this is a useful way to test robots navigation algorithms, but more test scenarios are necessary.

4.3 Other features of the simulation framework

The mobile robot simulator is useful both to quantitatively compare navigation algorithms for robots and to observe the performance of the algorithms at different cases of study e.g. the problem of local minimums.

Both of the studied algorithms have movement planning strategies based in sensors, with local obstacles dodge. These features imply the local minimums problem, which occur when the robot is navigating to a goal but gets trapped oscillating in the presence of obstacles in an area with the exit direction opposite to the goal, or when the movement results in a direction getting away to the goal.

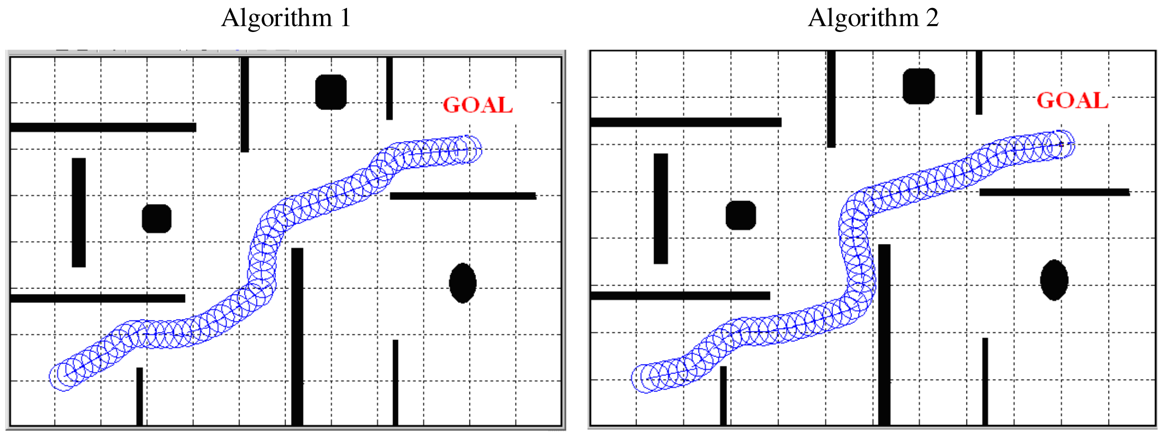


Fig. 4: Paths generated by the control algorithms: Start point (50,50), Goal (500,300)

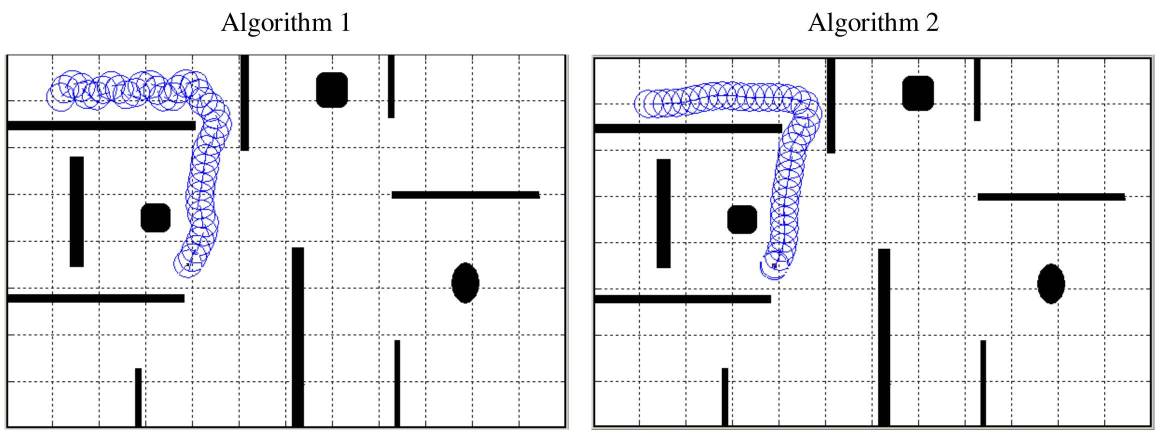


Fig. 5: Paths generated by the control algorithms: Start point (50,350), Goal (195,175)

Table 1: Robot performance

Metric	SM1 [cm]		SM2 [cm]		Min [cm]		PL [cm]		LeM		TB_E	
Scenario	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2	Alg. 1	Alg. 2
1	26.1	25.6	18.3	17.3	11	7	562.7	581.9	283	292	0.2463	0.0846
2	25.0	24.4	13.0	12.4	7	3	395.7	359.9	199	181	0.4007	0.0140

SM1 maximum = 26.5cm

The described situations create a conflict in the reactive behavior commanding the robot navigation, figure 6. The simulator evidences that the problem is more noticeable in algorithm 1, because the navigation direction is a result only of the vector sum of the attraction potential to the goal, and the repulsion potential, may enter in a local minimum when the robot navigated in a direction getting away of the goal.
In figures 7 and 8, the navigation mission is similar to that in scenery 2, it implies the movement from the point (50,350) to the point (160,175), and the goal is marked with a red point, which is 45 cm away from the original goal in figure 5. This slight modification causes that the robot with algorithm 1 stays trapped and the attraction and repulsion potentials are in conflict. Algorithm 2 achieves a satisfactory performance because the goal is located in a direction not totally opposed to the movement direction and the behaviors as searching of free areas and line following, sum in the movement direction allowing the robot exit this area and arrive to the goal.

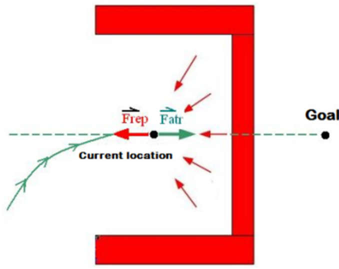


Fig. 6. Local mínimum example [7]

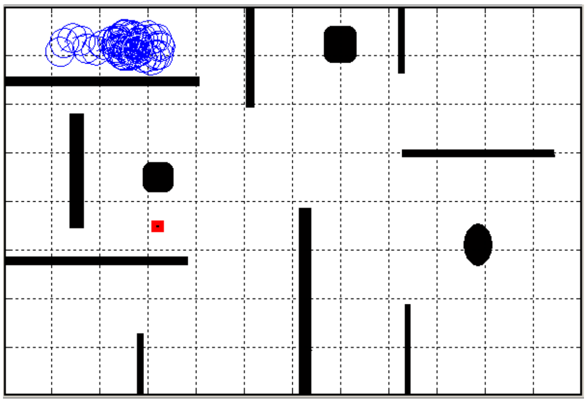


Fig. 7.Path generated by algorithm 1

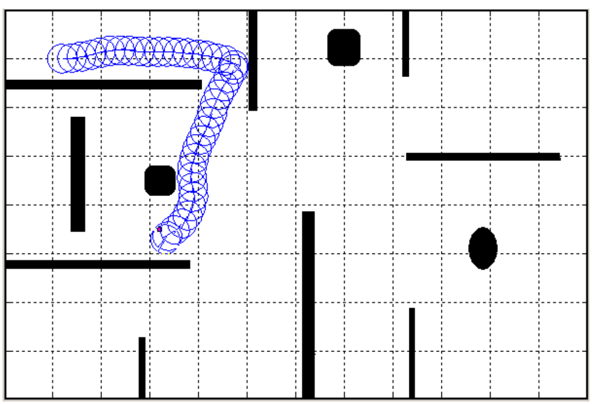


Fig. 8. Path generated by algorithm 2

In [15] and [16] there are some research work about local minimum and solving alternatives.

5. CONCLUSIONS

This paper describes a framework to provide analysis with several performance metrics. It is useful to compare mobile robots navigation algorithms including safety, dimension and smoothness of the trajectory. The suggested metrics are quite straight forward. However, it was shown that they can be used together to systematize simulated or experimental studies on control algorithms for mobile robot navigation.

A very simple didactic example was presented. The obtained results demonstrate the need to establish a procedure that can be used to analyze and compare navigation algorithms for mobile robots using several performance metrics. This is an open topic of research. It has become necessary to establish proper approaches and benchmarking procedures, for example, using a benchmarking standard framework for navigation algorithm assays and performance evaluation.

This metrics can be applied in simulated environments, but the performance metrics evaluation is more important in real environments. Many of the challenges in robot navigation come from the challenges of real environments, such an uncertainty in the sensors and the errors as odometry, which are generally not considered in simulation.

ACKNOWLEDGEMENT

This work was partial result of the 2061080241 project.The authors would like to thank to ITM, UdeA, and Polytechnic Jaime Isaza Cadavid.

REFERENCES

[1] J. Evans and E. Messina, "Performance Metrics for Intelligent Systems", Proceeding of the Performance Metrics for intelligent Systems Workshop, Gaithersburg, MD, August 14-16 of 2000.

[2] S. Wong, L. Middleton and B. MacDonald, "Performance metrics for robot coverage task", Proceedings Australasian Conference on Robotics and Automation ACRA, Auckland, New Zealand, 2002, pp. 7-12.

[3]Muñoz, N.; Valencia, J. & Londoño, N. (2007). Evaluation of Navigation of an Autonomous Mobile Robot, Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, Washington, DC, EEUU. http://www.nist.gov/el/isd/ks/upload/PerMIS_2007_Full_Proceedings.pdf

[4] S. Dongqing, "Aerial robot navigation in cluttered urban environments", PhD Thesis, The Florida State University, Florida, USA, 2006, pp. 87.

[5] J. Álvarez, Planificación del movimiento de vehículos autónomos basada en sensores. Tesis doctoral, Universidad de Oviedo, Oviedo, España, 1998, pp. 178.

[6] Y. Guo and J. Wang, "A new performance based motion planner for nonholonomic mobile robots", Proceedings of the 3rd performance metrics for the Intelligent Systems Workshop (PerMIS'03) NIST, Gaithersburg, MD, September 2003.

[7] J.C Latombe, "Robot Motion Planning", Kluwer Academic Publishers, 4th Edition, Boston, 1996.

[8] N. Muñoz, C. Andrade, N. Londoño. "Diseño y construcción de un robot móvil orientado a la enseñanza e investigación", Ingenieria & Desarrollo Ed. 9, 2006.
[9] J. Rosenblatt, "DAMN: Distributed Algorithm for Mobile Navigation. PhD. Thesis", Carnegie Mellon University Robotics Institute, Pittsburg, PA, 1997.
[10] M. Selekwa, E. Collins and J. Combey, "Multivalued Verus univalued Reactive Fuzzy Behavior Systems for Navigation Control of Autonomous Ground Vehicles", Proceedings from the 17th annual Florida Conference on the Recent Advances in Robotics FCRAR2004, May 20, 2004.
[11] E. Aguirre and A. Gonzalez, "Fuzzy Behaviors for mobile robot navigation, design, coordination and fusion", International Journal of Approximate Reasoning, vol. 25, 2000, pp. 255-289.
[12] A. Zalzala and A. Morris, Neural Networks for Robotic Control, Ellis Horwood, 1996, pp.278
[13] G. Cielniak, A. Treptow and T. Duckett, "Quantitative Performance Evaluation of a People Tracking System on a Mobile Robot", Proceedings of the European Conference on Mobile Robots (ECMR), Ancona, Italy, 2005.
[14]Nelson David Munoz Ceballos, Jaime Alejandro Valencia and Nelson Londono Ospina (2010). Quantitative Performance Metrics for Mobile Robots Navigation, Mobile Robots Navigation, Alejandra Barrera (Ed.), ISBN:978-953-307-076-6, InTech, Available from:
http://www.intechopen.com/books/mobile-robots-navigation/quantitative-performance-metrics-for-mobile-robots-navigation
[15] Ordóñez, C. (2006) Design and implementation of a limit cycle negotiation strategy for robot navigation. MSc Thesis, The Florida State University, 62pp, Florida, EEUU.
[16] Qing L.; Lijun W. ; Bo C. ; Zhou Z. An improved artificial potential field method for solving local minimum problem. 2nd International Conference on Intelligent Control and Information Processing (ICICIP), 2011 (Volume: 1) July 2011 ISBN: 978-1-4577-0813-8

Suggestions for the assessment of navigation algorithms in educational robotics

Nelson Munoz Ceballos[1,2], Jaime Valencia[3], Alessa Alvarez Giraldo[1,3]
[1] *Colombian Politechnic Jaime Isaza Cadavid,*
[2]*Metropolitan Technological Institute*
[3]*University of Antioquia*
Medellín-Colombia
E-mail: ndmunoz@elpoli.edu.co, nmunoz345@gmail.com

Abstract
A simple way of teaching the topics performance metrics and benchmarks in educational robotics is presented. Aspects related with the performance assessment of navigation algorithms in mobile robotics are described. The results obtained with a mobile robot in simulation and in a real environment are presented.

Keywords:
Educational robotics, Mobile robots, Navigation algorithms, Performance metrics, Benchmark

1. INTRODUCTION

Performance metrics, benchmarks and other widely accepted methods of comparison, are important tools for academic, scientific and industrial developments, as well as for commercial product manufacturing. The tests are an opportunity to show the strengths and limitations of current technology and provide data for future test [1]. However, research in robotics is weak in these tools. The complexity of robotics and intelligent systems is growing every day, it is necessary to define experimental approaches and procedures or quantitative comparative evaluation methodologies, which offer certain advantages, firstly, determine reliable reference methods in order to allow comparison of research results in mobile robotics, to make industrial application possible. On the other hand, the experimental part requires systematically repeated experiments, also it is needed to check whether to some extent the new procedures and algorithms proposed in research constitute a real breakthrough in certain topics or issues such as navigation with obstacle avoidance, SLAM, etc. [2], [3].

Performance metrics and benchmarks are important concepts which are gaining importance in international research, therefore it is transcendental formally include them in educational robotics.

2. OVERVIEW

Over the last years some steps have been taken to find a way in which the research results in robotics could be evaluated and compared in a framework of globally accepted procedures. This is an open topic of current research given the large number of new applications with mobile robotics and the interest of the international scientific community, in assessing their progress. In this context, initiatives as EURON[1] or BRICS[2], partnerships as IEEE[3] and institutions such as NITS[4] promote the development and application of methodologies for evaluation and comparison to improve the quality of research results, increase the likelihood of solid research publication and increase international visibility of these results, allowing rapid adoption and development of industrial applications [1], [4], [5].

Traditionally, international scientific community has employed mainly two methods for comparison of research: working teams on large challenges and robotics competitions. The working teams on large challenges are usually dealing with the identification of landmarks, important points they want to reach in certain field of robotics, for example define the best methodologies for experimentation in robotics, or how future human-robot interaction is going to be, among others. In this way a working standard for comparing progress in certain area is determined. For example, several years ago DARPA[5] agency proposed to universities and research institutes to develop

---

[1] European Robotics Research Network, http://www.euron.org/
[2] Best Practice In Robotics Project, http://www.best-of-robotics.org/
[3] Institute of Electrical and Electronics Engineers, http://www.ieee.org/
[4] National Institute of Standards and Technology, http://www.nist.gov/
[5] Defense Advanced Research Projects Agency, http://www.darpa.mil/

autonomous navigation systems for vehicles, in order to move through a path, without human intervention, only using a list of intermediate points[6]. This challenge has boosted research in this field with great success.

The robot competitions are another way to compare the performance of systems developed to perform a specific task with well-defined rules and metrics. Organizing these scientific competences has been a way to catch the attention of researchers and produce high quality solutions, but has the disadvantage of not being used as a way of evaluating and comparing continuously, due to high costs and the fact that are usually performed once a year [6], [7]. Figure 1 shows an area or test scenario of urban search and rescue for mobile robots used in the international competition Robocup.

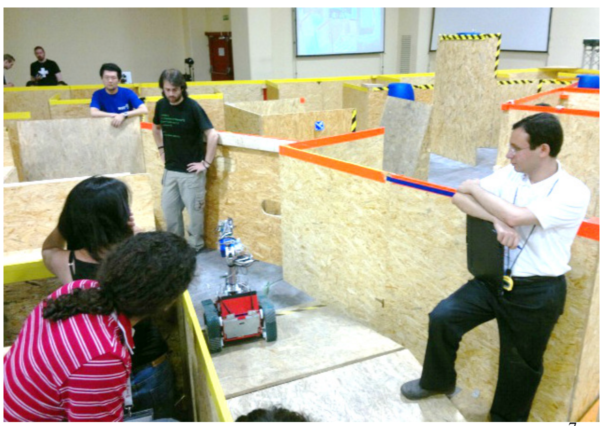


Fig. 1. Test scenario for autonomous mobile robots[7]

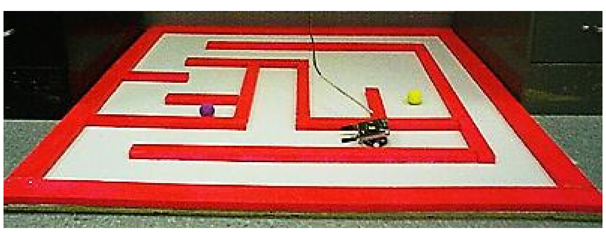


Fig. 2. Test scenario for a khepera mobile robot.

Figure 2 shows a test area for a Khepera robot, mobile robot widely used for research and education.

When the real scenarios are not available, an alternative is simulation software. Simulation is one of the most important tools in robotics research and development. On one hand, enables the evaluation of alternatives in the design phase of a robotic system. On the other hand, simulators allow continuing tasks associated with software development when the real robot is not available (e.g. because it is damaged or because someone else is using it). Although the simulators do not cover all the possibilities that arise in the real world, it is easier to build scenarios in a simulator and are generally helpful to reveal critical design points that need more attention and analysis in the real world. Furthermore, executing the navigation algorithms in a simulator offers the possibility of quickly and easily debug before bringing the robot to actual experimentation.

Figure 3 shows the main panel of Robotsim simulation software that represents the real environment shown in Figure 2, and allows simulating the execution of control algorithms as well as interacting with the Khepera robot sensors system.

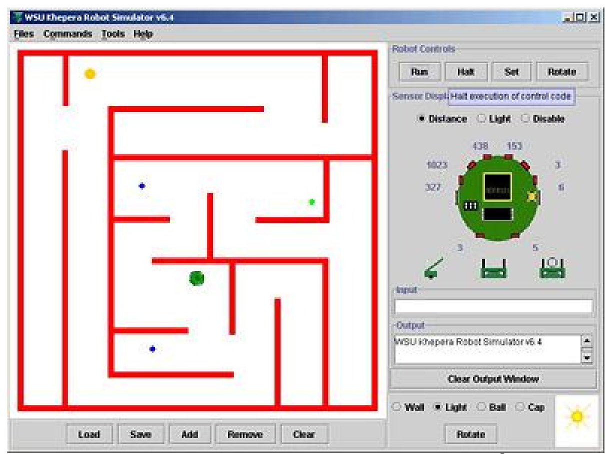


Fig. 3. Robotsim simulation software[8]

Despite the help offered by simulators, generally do not include the performance evaluation of the algorithms nor the comparison between them, most of them only allow timing and measuring the length of the path of a navigation mission.

The incorporation of a wide spectrum of performance metrics in an orderly and systematic manner in robotics research is relatively recent, the first event related to performance metrics for intelligent systems (including robots) was performed in August/2000[9], the first European event dedicated to performance metrics in robotics research was conducted in May/2006, it served as a prelude to the first global workshop on the topic, conducted in the framework of the international

[6] http://archive.darpa.mil/grandchallenge/
[7] http://www.robocup2012.org/

[8] Robotsim simulator, http://imej.wfu.edu/articles/2002/2/03/demo/robotsim/index.asp
[9] Performance Metrics for Intelligent Systems (PerMIS), http://www.nist.gov/el/isd/ks/permis.cfm

conference on robotics and intelligent systems October/2006[10].

Only some developments are oriented to simulate and evaluate the performance of algorithms for mobile robot navigation, combining metrics and procedures for comparison of results.

In [8] features of a framework for automatic evaluation of obstacle avoidance methods in scenarios with different features (density, complexity, congestion, etc.) are described. Performance is measured in terms of the robot parameters as robustness, optimization, security, etc.

In [9], a set of metrics are presented. These allow evaluation of the navigation quality of mobile robots algorithms and quantitative comparisons between different techniques for robot motion. This set of metrics go beyond the typical shortest-time or shortest-distance metrics, usually used as the unique performance measure for motion planning techniques: they include, e.g. measures regarding trajectory and smoothness that are related to real robot dynamic constraints. Simulation results are also presented.

In [6] different aspects of MoVeMA are presented, it is a reference framework for evaluating and comparing motion algorithms for mobile robots and autonomous vehicles, taking into account real-world problems such as uncertainty and non-holonomic constraints. This software is integrated as benchmarks database to the simulator Player / Stage.

3. WHAT SHOULD BE ASSESSED IN MOBILE ROBOTS NAVIGATION

A navigation algorithm provides basic capabilities to the mobile robot, such as the ability to evade obstacles and to generate a trajectory towards a goal. In [10] a detailed report on several important aspects that can be evaluated on the performance of a navigation algorithm it is also suggested how to publish the results, it represents a guidelines to apply a good experimental methodology in robotics.

This article is focused in the assessment of some performance features of navigation algorithms in mobile robots in a simple manner. This can be useful to teach the topics performance metrics and benchmarks in educational robotics.
There are several metrics that can be used to evaluate the performance of a navigation system, but none of them are able to indicate the quality of the whole system. Therefore it is necessary to use a combination of different indexes that quantify different aspects of the system. Having a good range of performance measurements is useful for: optimizing algorithm parameters, testing navigation performance within a variety of work environments, making a quantitative comparison between algorithms, supporting algorithm development and helping with decisions about the adjustments required for a variety of aspects involved in system performance [11],[12].

In navigation and obstacle avoidance, typical performance criteria are: [9],[13].
1. Mission success: number of successful missions.
2. Path length: distance traveled to accomplish the task.
3. Time: time taken to accomplish the task.
4. Collisions: number of collisions per mission, per distance and per time.
5. Obstacle clearance: minimum and mean distance to the obstacles.
6. Robustness in narrow spaces: number of narrow passages successfully traversed.
7. Smoothness of the trajectory: relative to control effort.

4. STUDY CASE, TESTS WITH A NAVIGATION ALGORITHM

In general terms, in an undergraduate robotics course, students learn the theories about robotics and practice them by performing simulation studies. In parallel, students perform robotics exercises in the laboratory [14]. Following this methodology, a control algorithm was designed for the autonomous navigation of the mobile robot Giraa_02, and the performance assessment is described.

4.1 Mobile robotics platform

Robot Giraa_02 was designed according to the characteristics of the common vehicles used for robotics in education or research. It has a cylindrical structure of 30cm diameter and approximately 20cm height; It has 8 ultrasound and 8 infrared sensors distributed equally around the robot's circumference, for these experiments, only 8 infrared sensors were taken into account, these have a distance range = 26.5cm and detection cone = 15 degree. The vehicle has a differential drive system, and its position and orientation are provided by a magnetic compass and an odometry system based on an optical mouse. The robot can be observed in figure 4 [15].

[10] IROS2006 (Workshop on Benchmarks in Robotics Research), http://www.iros2006.org/wt.html

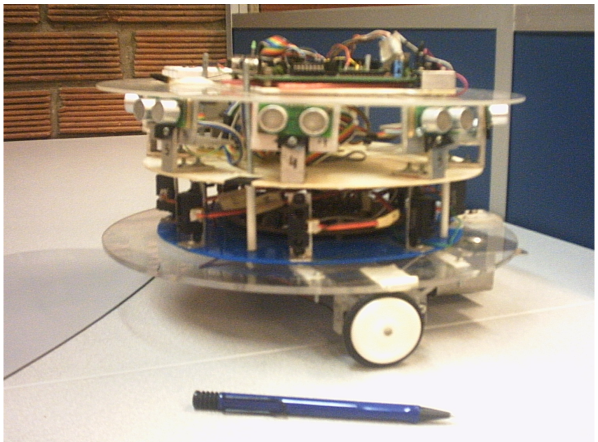


Fig. 4. Mobile robot Giraa_02

4.2 Navigation algorithm

A behaviors-based navigation algorithm was designed, inspired in AFREB (Adaptive Fusion of Reactive Behaviors) architecture [16], [17]. By using a neural network, an appropriate combination of the behaviors can be achieved; in this way, the system is able to perform complex tasks, such as navigation towards a goal, while evading obstacles in its path. The algorithm is depicted in Fig. 5, consists of the following modules: behavioral fusion, fusion supervisor, behavior primitives (1, 2,…n), and executor.

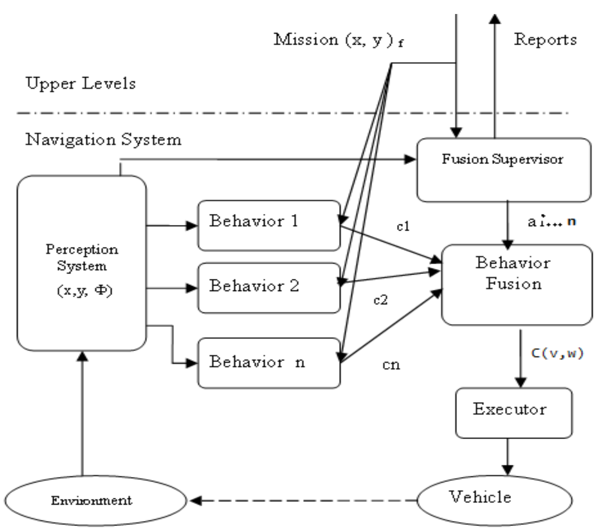


Fig. 5. Diagram of algorithm based in AFREB

Where:

(x,y,Φ) : current position and orientation

$(x,y)f$: final goal position

$c_1 \ldots c_n$: Behavior primitives output

a_i: Behavior weighs (coefficients)

$C(v,w)$: Emergent behavior (mission), linear and angular velocity vector

The primitive behaviors implemented are:

c_1: goal attraction

c_2: perimeter following (contour left - CW)

c_3: perimeter following (contour right - CCW)

c_4: free space

c_5: keep away (from objects)

If $c_1 \ldots c_n$ are the output of each primitive behavior, then the output of an emergent behavior, is:

$$C = \frac{\sum_{i=1}^{n} a_i \cdot c_i}{\sum_{i=1}^{n} a_i} \qquad (1)$$

Where a_i coefficients, with $0 \le a_i \le 1$, are found by an appropriate combination of measuring the provided information and collecting data by the perception system.

4.3 Metrics selection

The performance assessment of the navigation algorithm is based on some metrics described in [9].

Taking into account that the objective is to execute a navigation mission from a starting point to a final point (navigation mission towards a goal), an order of importance can be established for assessment of the navigation characteristics, as follows:

1. *The mean distance between the vehicle and the obstacles during the trajectory:* it considers the security of the trajectory and measures the risk taken by the robot in its movement towards the goal
2. *The distance covered by the vehicle between the starting point and the goal, and the time needed to complete the mission:* it measures aspects related to the planning of the trajectory.
3. *The smoothness of the trajectory:* it considers the quality of the trajectory according to the energy and time required for the movement.

These characteristics can be analyzed using the following set of performance metrics:

4.3.1 Security metrics

These metrics express the robot security while it travels through a trajectory, taking into account the distance between the vehicle and the obstacles in its path [18].

Security Metric-1 (SM1): Mean distance between the vehicle and the obstacles through the entire mission measured by all the sensors; the maximum value will be produced in an obstacle free environment. If the deviation of the index from its maximum value is low, it means that the chosen route had fewer obstacles.

Security Metric-2 (SM2): Mean minimum distance to obstacles. This is taken from the average of the lowest value of the n sensors. This index gives an idea of the risk taken through the entire mission, in terms of the proximity to an obstacle. In an obstacles free environment SM1 = SM2 is satisfied.

Minimum Distance (Min): Minimum distance between any sensor and any obstacle through the entire trajectory. This index measures the maximum risk taken throughout the entire mission.

4.3.2 Dimension metrics

The trajectory towards the goal is considered in its time and space dimensions. In general, it is assumed that an optimal trajectory towards the goal is, whenever possible, a line with minimum length and zero curvature between the initial point (x_i,y_i) and the final point (x_n,y_n), covered in the minimum time.

Length of the Covered Trajectory (P_L) is the length of the entire path covered by the vehicle from the initial point to the goal. For a trajectory in the x-y plane, composed of n points, and assuming the initial point as (x_1, $f(x_1)$) and the goal as (x_n, $f(x_n)$), P_L can be calculated as:

$$P_L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (f(x_{i+1}) - f(x_i))^2} \quad (2)$$

Where (x_i, $f(x_i)$) and i = 1, 2, . . . , n are the n points of the trajectory in Cartesian coordinates [19].

Control Periods (LeM): It is the amount of control periods. This metric relates to the number of decisions taken by the planner to reach the goal, if the robot moves with lineal and constant speed (v). This gives an idea of the time needed to complete the mission [18].

4.3.3 Smoothness metrics

The smoothness of a trajectory shows the consistency between the decision-action relationship taken by the navigation system, as well as the ability to anticipate and to respond to events with enough speed [20]. The smoothness of the generated trajectory is a measure of the energy and time requirements for the movement; a smooth trajectory translates into energy and time savings [21]. Additionally, a smooth trajectory is also beneficial to the mechanical structure of the vehicle.

Bending Energy (B_E): This is a function of the curvature, k, used to evaluate the smoothness of the robot's movement. For curves in the x-y plane, the curvature, k, at any point (x_i,$f(x_i)$) across a trajectory is given by:

$$k(x_i, f(x_i)) = \frac{f''(x_i)}{(1 + (f'(x_i))^2)^{\frac{3}{2}}} \quad (2)$$

The bending energy can be understood as the energy needed to bend a rod to the desired shape [22]. B_E can be calculated as the sum of the squares of the curvature at each point of the line k(x_i,f (x_i)), along the length of the line L. So, the bending energy of the trajectory of a robot is given by:

$$B_E = \frac{1}{n} \sum_{i=1}^{n} k^2(x_i, f(x_i)) \quad (3)$$

Where k(x_i, f(x_i)) is the curvature at each point of the trajectory of the robot and *n* is the number of points in the trajectory.

The value of B_E is an average and does not show with enough clarity that some trajectories are longer than others. Therefore, TB_E can be used instead; this metric takes into account the smoothness and length of the trajectory simultaneously.

TB_E is defined by $$TB_E = \int_a^b k^2(x)dx \quad (4)$$

and numerically, $$TB_E = \sum_{i=1}^{n} k^2(x_i, f(x_i)) \quad (5)$$

In a straighter trajectory, the values B_E and TB_E will be lower, which is desirable since the energy requirement is increased according to the increase in the curvature of the trajectory.

5. SIMULATION

A simulation framework was used for assessment of the navigation algorithm. This simulation software facilitates teaching and researching in mobile robot navigation. For the execution of a navigation mission between two points, an environment similar to offices was created, i.e. A 6m x 4m frame, structured environment with static obstacles, some obstacle borders are sharp, there are also straight line obstacles, and narrow zones. The robot was simulated according to the characteristics of the Giraa_02 robotics platforms used in the lab.

The path generated by the algorithm, in the scenario is shown in figure 6. The robot moves towards a goal, while avoiding obstacles, from the point (50,50) to (500,300). A complete analysis of the algorithm simulation is presented in [23].

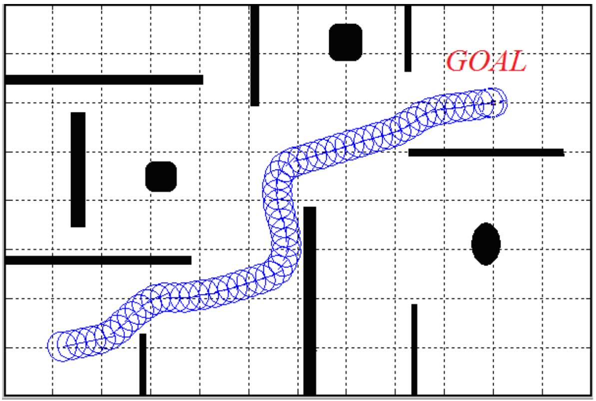


Fig. 6. Path generated by the simulation

6. TESTS WITH GIRAA_02

The real environment of the tests is displayed in figure 7, with the same dimensions of the simulated environment. Figures 8, 9 y 10 show the real motion of Giraa_02 in different tests moving from the point (50,50) to (500,300).

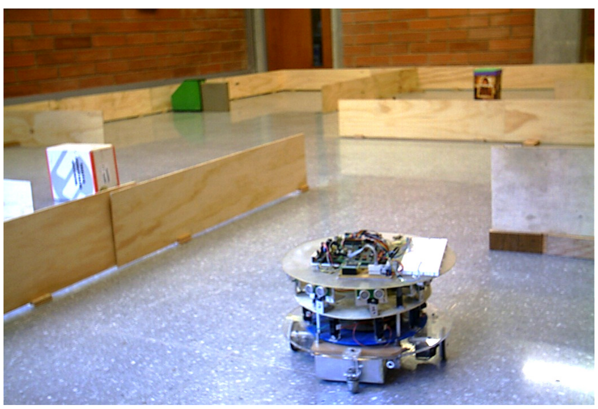


Fig. 7. Real environment for navigation tests of robot Giraa_02.

It was necessary to make some adjustments to the robot to achieve a satisfactory performance in the real environment, since some aspects of the system dynamics such as inertia, friction, noise, etc., are not covered by the simulation.

The magnetic compass is affected by magnetic fields present in the laboratory, power lines and other electrical devices. This directly influences the guiding direction of the robot.

The odometry system or position sensor must have a distance of no more than 2 or 3 mm from the floor to sense the movement correctly because it is based on an optical mouse. With this short distance between the robot and the floor some problems were caused due to friction with the floor, uneven parts (not totally flat floor) and the accumulation of small pieces of trash.

The operation of the infrared sensors is affected by the surrounding light (ambient noise), which prevented properly detection of obstacles and caused some collisions.

Accordingly, the adjustment to the software was related with a proper additional processing of the sensors signals to filter noise. It was also necessary to reduce the output of repulsive behavior, because it was dominating the overall response of the merger of behaviors, it soothed the contribution of the other and generated wrong paths. In addition, the minimum detection distance was modified in the contour following behavior.

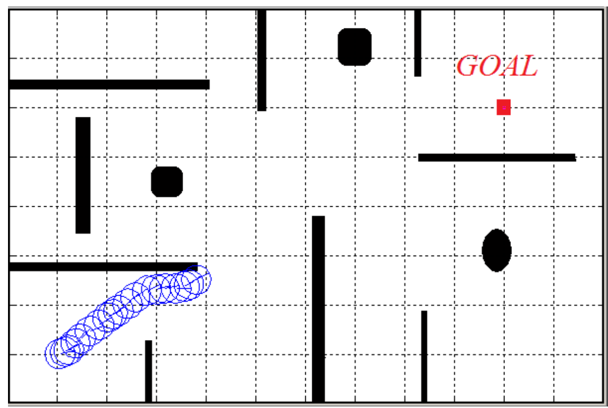


Fig. 8. The robot collides with obstacles, mainly due to problems in the perception system.

Figure 8 Figure shows that the robot actually guides and navigates to the target, but does not detect obstacles well, causing collisions.

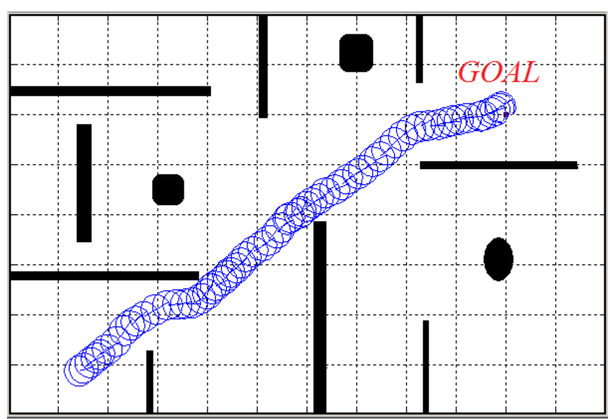


Fig. 9. After adjustments to the software, the robot is still clearing obstacles, mostly with thin edges.

After modifying the detection distance in the contours tracking behavior, response improves, as seen in figure 9, colliding only in very thin edges or borders, hard to perceive, then the detection distance is adjusted again.

7. RESULTS

Figure 6 shows the result of the simulation, the robot moves toward the target generating a smooth and safe path, having a mínimum distance of 7 cm from any obstacle.

During the actual experiment, after adjustments, satisfactory performance is obtained, Figure 10. In the path followed by the robot, it is observed that it manages to loosely evade the obstacles and get to the point (495.309), close to the target, with an error of about 3%.

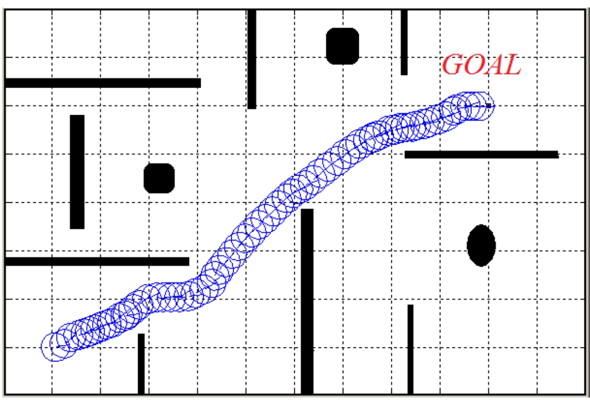


Fig. 10. The robot performs the mission

In table 1, results obtained from the simulation and the real test are summarized, according to the quality metrics described.

Table 1: Comparison of the robot Giraa_02 performance in simulation and in the real environment.

Performance Metric	Navigation Algorithm	
	simulation	real
SM1 [cm]	25.6	25.9
SM2 [cm]	17.3	18.9
Min [cm]	7	6.4
PL [cm]	581.9	533.4
LeM	292	196
TB_E	0.0846	0.4975

In general, the robot travels through clearways as indicated by SM1 and SM2.

Min index indicates that at any given time of the real mission, the robot passes 6.4 cm of an obstacle, in this sense, we can consider that the path was a bit more hazardous for the robot in the real test than in the simulation.

Due to the reduction of the repulsion performance output, the robot generates a goal-oriented trajectory consequently shorter and with less control cycles than in the simulation, this is reflected in the reduction of indexes LeM and PL.

The real trajectory is less smooth as indicated by TB_E, this is also identified by observing figures 6 and 10.

8. CONCLUSIONS

These suggestions for the assessment of navigation algorithms provide a tool for educational robotics. A very simple application example was presented. The obtained results demonstrate the need to establish a procedure that could be used to analyze and compare control algorithms using several performance metrics. This is an open topic of research. It has become necessary to establish proper approaches and benchmarking procedures, for example, using a benchmarking standard framework for navigation algorithm assays and performance evaluation.

This metrics can be applied in simulated environments, but the performance metrics evaluation is more important in real environments. Many of the challenges in robot navigation come from the challenges of real environments, such as uncertainty in the sensors and the errors in odometry, which are generally not considered in simulation.

ACKNOWLEDGEMENT

This work was partial result of the 2061080241 project. The authors would like to thank to ITM, UdeA, and Polytechnic Jaime Isaza Cadavid.

REFERENCES

[1] Weiss Brian Adam (2012) Multi-relationship evaluation design (mred): an interactive test plan designer for advanced and emerging technologies, PhD. Thesis in Mechanical Engineering, University of Maryland

[2] IEEE/RSJ IROS (2008) Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems (Nice-France, 26 September/2008).

[3] Gembench (2009) Workshop on Good Experimental Methodology and Benchmarking in Robotics Research and Applications - at the Euron Annual General Meeting (Leuven, Belgium, 6-8 April, 2009)

[4] Del Pobil A.P. (2006) Why do We Need Benchmarks in Robotics Research? in Proc. of the Workshop on Benchmarks in Robotics Research, IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China.
[5] Clawar (2008) Session on Benchmarking and standardization for clawar and mobile robots, Coimbra, Portugal, 8-10 September.
[6] Calisi, D., Iocchi, L., & Nardi, D. (2008). A unified benchmark framework for autonomous mobile robots and vehicles motion algorithms (MoVeMA benchmarks). In RSS workshop on experimental methodology and benchmarking, Zurich, Switzerland.
[7] Jacoff A., Sheh R., Virts A., Kimura T., Pellenz J., Schwertfeger S., Suthakorn J. (2012) Using Competitions to Advance the Development of Standard Test Methods for Response Robots. In Performance Metrics for Intelligent Systems PERMIS 2012, pp 182-189
[8] Raño, I., & Minguez, J. (2006). Steps towards the automatic evaluation of robot obstacle avoidance algorithms. In Proc. of workshop of benchmarking in robotics, in the IEEE/RSJ int. conf. on intelligent robots and systems (IROS).
[9]Muñoz, N.; Valencia, J. & Londoño, N. (2007). Evaluation of Navigation of an Autonomous Mobile Robot, Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, Washington, DC, EEUU.
http://www.nist.gov/el/isd/ks/upload/PerMIS_2007_Full_Proceedings.pdf
[10] EuronGEMSig (2008). Good experimental methodology guidelines, European Robotics Research Network (EURON), Tech. Rep., 2008. [Online] Available:
www.heronrobots.com/EuronGEMSig/Downloads/GemSigGuidelinesBeta.pdf
[11] J. Evans and E. Messina, "Performance Metrics for Intelligent Systems", *Proceeding of the Performance Metrics for intelligent Systems Workshop,* Gaithersburg, MD, August 14-16 of 2000.
[12] G. Cielniak, A. Treptow and T. Duckett, "Quantitative Performance Evaluation of a People Tracking System on a Mobile Robot", *Proceedings of the European Conference on Mobile Robots (ECMR)*, Ancona, Italy, 2005.
[13]Minguez, J. (2008). Robot Obstacle Avoidance Papers Using Experiments, Tech. Rep., 2008. [Online] Available:
www.heronrobots.com/EuronGEMSig/Downloads/GemSigGuidelinesBeta.pdf
[14] Seul Jung. Experiences in Developing an Experimental Robotics Course Program for Undergraduate Education. In: Education, IEEE Transactions on. Volume: 56, Issue: 1, Year: 2013, page(s): 129 – 136.
[15] Muñoz, N.; Andrade, C. & Londoño, N. (2006). Diseño y construcción de un robot móvil orientado a la enseñanza e investigación, In: Ingenieria & Desarrollo # 19. ISSN: 0122-3461
[16] Gachet D.; Salichs, M. A.; Moreno, L.; Pimentel, J. R. (1994). Learning Emergent Tasks for an Autonomous Mobile Robot. Proceedings of the International Conference on Intelligent Robots and Systems (IROS '94).
[17] Zalzala, A. & Morris, A. (1996). Neural Networks For Robotic Control, Ellis Horwood, 1996, pp.278.
[18] Alvarez, J. Planificación del movimiento de Vehículos autónomos basada en sensores. PhD Thesis, University of Oviedo, Oviedo, Spain, pp. 178, 1998.
[19] Y. Guo and J. Wang, "A new performance based motion planner for nonholonomic mobile robots", Proceedings of the 3rd performance metrics for the Intelligent Systems Workshop (PerMIS'03) NIST, Gaithersburg, MD, September 2003.
[20] J. Rosenblatt, "DAMN: Distributed Algorithm for Mobile Navigation. PhD. Thesis", Carnegie Mellon University Robotics Institute, Pittsburg, PA, 1997.
[21] S. Dongqing, "Aerial robot navigation in cluttered urban environments", PhD Thesis, The Florida State University, Florida, USA, 2006, pp. 87.
[22] E. Aguirre and A. Gonzalez, "Fuzzy Behaviors for mobile robot navigation, design, coordination and fusion", *International Journal of Approximate Reasoning*, vol. 25, 2000, pp. 255-289.
[23]Nelson David Munoz Ceballos, Jaime Alejandro Valencia and Nelson Londono Ospina (2010). Quantitative Performance Metrics for Mobile Robots Navigation, Mobile Robots Navigation, ISBN:978-953-307-076-6, Ed. InTech, Available from:
http://www.intechopen.com/books/mobile-robots-navigation/quantitative-performance-metrics-for-mobile-robots-navigation

A cross-generational robotics project day: Pre-school children, pupils and grandparents learn together

Martin Kandlhofer*, Gerald Steinbauer*, Sabine Hirschmugl-Gaisch† and Johann Eck†
*Institute for Software Technology, Graz University of Technology, Austria
Email: [mkandlho, steinbauer]@ist.tugraz.at
†University of Teacher Education Styria, Austria
Email: hirschmugl.gaisch@aon.at, hans.eck@ainet.at

***Abstract*—Using robotics to interest kindergarten children in science and technology in a playful way is a rather new idea in educational robotics. Aiming to foster the integration of robotics already in pre-school education this paper presents an innovative robotics project for kindergartens. Initialized by the Graz University of Technology and the University of Teacher Education, the project involved different scientific and educational institutions. The focus was put on the cross-generational aspect, integrating kindergarten children, pupils up to the age of thirteen as well as senior citizens in order to initiate a vital social process between the different age groups. Within the project a robotics day in a kindergarten offering eleven different hands-on experiments, where children could actively participate, was organized. The goal of the project presented in this paper was to familiarize children in pre-school age as well as young school students with science and technology using different robotics platforms as pedagogical tools. Aiming at the investigation of the impact of the robotics project a first qualitative evaluation was conducted.**

***Keywords*—RoboCupJunior, educational robotics, robotics in kindergarten, cross-generational aspects, qualitative evaluation**

I. Introduction

In the last decades educational robotics has gained increased attention. Several conferences and workshops deal with the use of robotics in education [1]. In addition initiatives like *RoboCupJunior (RCJ)* aim to interest young children and pupils up to the age of nineteen in science and technology [2]. On the contrary educational robotics with special focus on children aged between three and six years is less widespread. Science and technology are changing rapidly and young children have to be prepared for this development. The idea behind the concept of educational robotics in kindergarten is to use the robot as pedagogical tool to familiarize children in pre-school age with science and technology in a playful way.

By presenting an innovative cross-generational project for kindergartens this paper discusses how different robotics platforms could be integrated in the education of children between three and six years of age. Furthermore, it presents an interesting concept within the field of educational robotics: Different age groups (kindergarten children, pupils aged from eleven to thirteen, senior citizens) and different scientific and educational institutions (kindergartens, schools, universities) work together on a joint robotics project. Qualitative feedback was collected and analyzed within a first empirical evaluation. The aim of this evaluation was to investigate the learning effects and the medium-term impact (up to eight months) of the project on participating kindergarten children and pupils. Preliminary results and findings of the qualitative evaluation are presented in this paper.

The remainder of the paper is structured as follows: Chapter II deals with related research whereas chapter III gives a brief overview of the current situation of educational robotics in kindergartens in Austria. Chapter IV provides a detailed description of the kindergarten project followed by the presentation of preliminary evaluation results in chapter V. Chapter VI draws conclusions and discusses future work.

II. Related research

As the level of awareness and importance of educational robotics rose over the last decades a great number of conferences, workshops, papers and books have been addressing this topic [1], [3], [4]. Alimisis and colleagues [5] for instance provide in their book an extensive overview of the theoretical background as well as practical aspects of robotics in education.

In [6] the authors describe how robotics can act as a tool to teach pupils the basics of engineering and programming. In addition they conducted empirical studies in order to investigate why robots seem to motivate children, even if they were not technically interested beforehand.

Whereas the use of robotics in pre-school education is not as wide-spread as in primary and secondary school various papers and articles exist which describe robotics platforms and projects for young children. For instance the authors of [7] present the experiences made introducing robotics in a kindergarten using *Lego WeDo*. Children had to build a small robot step by step. Afterwards they interacted with the robot, which was actually programmed by a teacher.

The article in [8] describes the integration of robotics in early childhood education following a constructionist strategy (learning by designing, using concrete objects to explore, identification of powerful ideas, self-reflection).

Janka [9] presents the use of the programmable robot-toy *Bee-Bot* in pre-school education. Different activities and games for kindergarten children and teachers were designed and qualitatively evaluated. The focus of this research was based on robot programming instead of construction and design. It turned out that although all children involved in the study basically enjoyed playing with the Bee-Bot and were not afraid of using this new technology the robot itself was not interesting to them for a longer period of time. The author also stated that some of the children showed a basic understanding of the robot's control principles whereas others seemed to be too cautious to increase their self confidence during the work with the Bee-Bot.

A short look in the history reveals that already in the early 19th century the German pedagogue Friedrich Froebel, who coined the term 'kindergarten', developed a series of educational toys and hands-on learning strategies. Many modern learning tools, for instance the *Lego Mindstorms* robotics kit, are based on his work [10], [11].

III. BACKGROUND

Educational robotics for primary and secondary schools is well established in Austria. Among other initiatives a nationwide network of RoboCupJunior regional centers provides support for schools, teachers and pupils [12]. On the contrary only a few initiatives and projects can be found which use robotics in kindergarten and pre-school education.

One example would be the robotics course "Robots for Kids" which was set up in 2010 by the University of Applied Sciences Technikum Wien. The target group for this course are kindergarten children at the age of four to six years. Within the classes children can actively participate and in parallel they get a first impression of scientific working [10].

As another example the project "Technical and natural science in playschool" of Vienna University of Technology could be mentioned. Children aged between four and six years have the opportunity to visit different departments of the university and participate in experiments. Within this project one of the main topics was robotics.

Additionally, different scientific institutions and universities offer training courses and workshops for educators and children. For instance the Austrian Computer Society offers robotic workshops in order to teach kindergarten pedagogues how to integrate robotics into teaching.

The "Technisches Museum Wien" organizes workshops for children between the age of four and seven to teach them the basics of programming and robotics.

The initiative "Children visit Science" is an innovative approach within the context of kindergarten pedagogy in Austria. The intergenerational, cross-organizational project was originally initiated in 2010. The basic aim of this initiative is to provide pre-school children and pupils with access to different scientific fields and furthermore to give an insight into the research sector at different scientific institutions [13], [14].

In the first year the initiative comprised five educational modules, focusing on different topics (bioscience, experimental physics, criminalistics, chemistry, paper manufacturing). In spring 2012 a scientific project day on the subject of electrostatics and electricity was organized. Secondary school students in cooperation with their teachers prepared different hands-on experiments dealing with topics like how to establish a power circuit or how to test the conductivity of different materials. Pupils acted as guides explaining the experiments to kindergarten children. This concept formed the basis of the scientific robotics day described in section IV [13], [14], [15], [16].

Fig. 1: Two children working with the *Bee-Bot*

Almost all above mentioned robotics projects and workshops use the *Bee-Bot*, manufactured by the British company *PrimaryICT*, as a learning tool (see Figure 1). The small programmable wheeled robot, designed for pre-school and early primary school children, is a widely adopted tool within the context of educational robotics in kindergarten. It can be controlled according to the principles of the *Logo* programming language [17]. Using the buttons on the back of the robot (forward, backward, rotate left, rotate right) children can enter a sequence of commands. Each forward/backward instruction moves the robot 15cm in the corresponding direction whereas each rotation instruction turns the robot by 90 degrees without changing its current position [9].

IV. PROJECT DESCRIPTION

In November 2012 a cross-generational scientific kindergarten experiment day with special focus on robotics was organized as a joint project between a secondary school, a kindergarten, the University of Teacher Education and Graz University of Technology (TUG). The structure of the robotics day was based on the concept "Children visit Science" and the scientific project day on electrostatics and electricity described in section III.

One main objective of the robotics project day was to prepare contents of the area of robotics respecting pedagogical and didactic aspects as well as principles of educational robotics ([5], [18], [19], [20]). Therefore, members of the robotic lab at TUG together with kindergarten pedagogues and teachers developed eleven different hands-on experiments and educational games applying methods of research-based learning ([21]) and the technique of storytelling ([13], [22]). Respecting fundamental principles of educational robotics as stated by Frangou and colleagues in [18] children could actively participate, explore, test and interact with the robots.

During the project day at the kindergarten each experiment was carried out at a separate hands-on area, also referred to as ’experiment station’. According to the concept of an education partnership [23], secondary school students carried out and explained the experiments to kindergarten children and their grandparents. Pupils slip into the part of a teacher, accompanying the kindergarten children through their way of discovering and experiencing.

In preparation for their tasks pupils attended a half-day robotics workshop. Before this workshop they did not know any details about the experiments or the different tasks. The teacher only announced that she is looking for volunteers joining a robotics project. In the workshop pupils were first introduced to the basic concepts of robotics and the scientific background of each robotics experiment (e.g. explanation of sensor, motors, robot programming, and so forth). Students could choose their favourite robot to work with. Afterwards they got detailed instructions on how to carry out and guide different experiments.

To give the different age groups participating (pre-school children, pupils, senior citizens) a basic understanding of robotics and artificial intelligence the experiment stations were structured around following major items using different robotics platforms:

- the programmable wheeled robot *Bee-Bot* [9]
- functionality of sensors using the *LEGO Mindstorms NXT 2.0* robotic kit [24]
- the humanoid robot on the example of the *Hitec RoboNova* [25]
- mapping and object tracking using the *Pioneer 3 DX* robot [26]

Figure 2 shows the different robotics platforms used as well as the excitement of children and pupils while carrying out hands-on experiments. In addition Figure 3 provides an overview of experiments focusing on different types of sensors. Following a brief description of each covered topic.

A. *Telling a story using the Bee-Bot*

Based on the functionality of the Bee-Bot described in Chapter III two educational games were developed. The idea behind was to embed the tasks children have to accomplish into a story. In the first game children had to program the robot to follow a certain path on a special square grid mat. The path represented the different production stages in a glass factory (also see Figure 2a). The research question to the children was: ”Can you teach the Bee-Bot how to make glass?”.

The challenge of the second game was to program the robot moving from a starting point to an endpoint, stopping at certain intermediate positions on a square grid mat with fairy-tale motifs imprinted. The research question for this task was: ”Can you tell the story of the *bad wolf and the three little piglets* whereby the Bee-Bot is acting the wolf?”

B. *Functionality of sensors*

Seven hands-on experiments demonstrated the use and the functionality of the ultrasonic-, the light-, the sound- and the color-sensor. Children could interact with the different robots which were build using Lego Mindstorms. Research topics included: ”Follow the light”, ”Don’t drop from the table” (Figure 3b), ”Avoid collisions”, ”Sweet-serving service robot” (Figure 2c), ”Find and grab the can” (Figure 3d), ”Sort the color bricks” (Figure 3a) and ”Follow the noise” (Figure 3c).

C. *Humanoid robots*

Using the example of the humanoid robot RoboNova the basics of humanoid robots were demonstrated. Pupils could control the robot by sending commands via the infrared remote controller. Children had to watch the robot carefully and afterwards imitate its movements (Figure 2b). The research question was: ”Is a robot better at dancing than me?”

D. *Mapping and object tracking*

This experiment station dealt with the topics of mapping and object detection using the Pioneer 3 DX robot with a SICK laser scanner and a Microsoft Kinect camera (Figure 2d). First the robot autonomously created a map of the kindergarten. Children followed the robot on it’s way through the building. Afterwards the Pioneer performed an object tracking task using the Kinect camera. Children could actively interact with the robot by moving an orange ball. In parallel a member of TUG provided explanations on the functioning of the robot and the basic principles of mapping and object tracking. The tasks for the children were formulated as follows: ”Supporting the rescue robot” and ”Playing football with a real robot”

V. Results and preliminary evaluation

The first cross-generational robotics day was conducted respecting pedagogical and didactic aspects. Overall twenty-five kindergarten children participated. They had been divided into groups of three. Moreover ten pupils participated. Each group of children was accompanied by at least one grand-parent. The described approach combined two major benefits: On the one hand pupils learned about scientific topics not only during the preparation process but also by guiding and explaining the experiments to kindergarten children. On the other hand kindergarten children had the opportunity to learn and gather practical experiences together with pupils and senior citizens. In this context one important aspect was that pre-school children could actively participate in the experiments. Furthermore the integration of different age groups and different educational institutions fostered a vital social process between kindergarten children, young students, senior citizens as well as mentors, teachers and staff members of participating institutions. In general the concept of discovering and experimenting represents a valuable pedagogical approach within the area of pre-school education, fostering the learning process of children in a holistic way. In addition the robotics day formed the basis for a follow-up project at the kindergarten in order to deepen what children have seen and experienced [13], [14].

A. *Qualitative evaluation*

Within our plan to evaluate the cross-generational robotics project the first step was to investigate the impact on the age group of participating pupils. In the following step we will also investigate the impact on the group of kindergarten children.

(a) Glass factory robot (Bee-Bot) (b) Humanoid dancing robot (Hitec RoboNova) (c) Service robot (Lego Minstorms NXT 2.0) (d) Rescue robot (Pioneer 3 DX)

Fig. 2: Kindergarten children, pupils, students and senior citizens together carrying out hands-on robotics experiments on different robotics platforms

We conducted semi-structured interviews [27] to collect qualitative data as well as to get positive and negative feedback with school students who guided the experiments during the robotics day. In order to obtain information about the medium-term impact and the learning effects it was decided to conduct the interviews around six months after the robotics day. The interviews took place at the school directly. Seven out of ten pupils voluntarily agreed on participating in this study.

1) Methodology: The qualitative research technique of semi-structured interviewing is commonly used within the field of sociology, psychology, educational science and empirical software engineering [27], [28]. Preparing, conducting and analysing qualitative interviews are time consuming tasks. Nevertheless, we decided on applying this method since our aim was not only to obtain quantitative data but also to get personal feedback and collect additional information (i.e. interviewees' facial expressions, moods and feelings).

Based on the observations made during the robotics day and on discussions with teachers and pedagogues a set of questions, acting as a guideline during the interview, was designed. It was essential that those questions were formulated in an open-ended, non-directional way in order to avoid influencing interviewees' answers [29].

The first questions dealt with background information, information about the specific task performed as well as previous knowledge in the field of robotics. The main part dealt with pupils' experiences during the robotics day followed by questions asking for improvement suggestions and further experiences in the field of robotics made after the robotics day. The final question posed (only in case the interviewees did not already provide an answer) dealt with lessons learned from the pupils' point of view. Following a listing of the guiding questions[1]:

1) *Which grade do you attend?*
   a) *What is your favourite subject in school?*
2) *What was your task during the robotics day?*
   a) *Why did you choose this task?*
3) *What did you know of robots before you participated in this robotics project?*
4) *Please describe your experiences during the robotics day.*
   a) *Did everything work out as it was supposed to (conducting and explaining experiments, acting as a guide)?*
5) *How was the cooperation with the kindergarten children?*
   a) *Where the children interested in the experiments? Did they actively participate?*
6) *How was the cooperation with the senior citizens?*
7) *Do you remember some situation or some activity especially? And why?*
8) *What would you change on the next robotics day?*
9) *Did you make any experiences in the field of robotics after the project?*
10) *What did you learn within the scope of this robotics project?*

[1]All questions were translated to English since all interviews were originally conducted in German.

For later analyses all interviews were audio-taped. Interviewees were asked for their permission to record the conversation. Furthermore, parents were asked to sign informed consents describing the main purpose and the procedure of the interview as well as stating legal and ethical aspects. All collected data was treated confidentially and personal information was made anonymous.

2) Preliminary findings: For the analysis of qualitative data various different techniques could be applied (see [29]). Our approach was to transcribe all recorded interviews, to summarize inherent quantitative data and finally to perform a content analysis [30].

We interviewed seven students (four girls, three boys) aged from eleven to thirteen who currently attend grade two of secondary school. They all have basic knowledge of computers since this school provides one lesson of computer science every two weeks. Three pupils stated that they had previous experience with robot toys, one boy reported that he once watched a friend working with Lego Mindstorms and one girl already attended a Lego Mindstorms robotics workshop in primary school. The other two students never had anything to do with robotics.

As described in the previous section students participated in a half-day preparation workshop. Basically they could decide themselves which experiment to guide during the robotics day. Most pupils chose experiments which seem to fit their personal interests and talents. For instance one student interested in sports and physical education insisted on guiding the robot-dance station. Another student, who is a very talented speaker, decided for the Bee-Bot station where it was her task to retell a fairy tale while providing explanations on how to program the robot. Only one student reported that his robot was *"too complicated to handle"* and questions asked by visitors were *"too tricky"*. Asked for the topic and name of his station, the student had to think for a while until he could remember. It finally turned out that student's task was assigned by the teacher instead of chosen voluntarily.

Pupils also talked about their most memorable situations and experiences. One student for instance stressed out the special situation when he was controlling the humanoid dancing robot in front of a big audience. Similarly, two pupils talked about the joy of slipping into the part of a teacher, *"explaining things to little kids"*. Another student mentioned the great feeling of success when she illustrated the functioning of the robot to a girl from Romania which did not speak German at all[2]. Two pupils also remembered negative experiences (having trouble with a difficult kindergarten child; difficult technical questions by one grandparent; being afraid to provide explanations in English).

One aim of this qualitative evaluation was to find out what interviewees actually think about lessons learned and knowledge gained. Following a brief overview of students' statements:

- kindergarten children understood the functioning of the different robots very fast
- robotics is fascinating but it's much harder than expected that robots actually do what programmers want them to do
- many different robotics platforms and types of robots exist
- constructing and programming of robots mean a lot of work
- teamwork is important if you want to construct and program a robot
- the robotics project was an opportunity to improve English and presentation skills
- programs have to be written first and afterwards transferred to the robot

In sum all seven students were enthusiastic about their participation in the robotics project. Suggestions for improvement included the integration of one or two *"bigger robots with arms and legs or tracks"*. The overall feedback was mainly positive although interviewees also mentioned some problems and challenges during the robotics day (i.e. jamming robot gearwheels, unexpected robot behaviour, being nervous while speaking in front of an audience, providing explanations in English[3], tricky questions, troubles with difficult children). However, pupils pointed out the 'positive feeling' after handling these issues successfully (either on their own or by asking for assistance). During the interviews they still talked about 'their' robot and 'their' experiment station, even half a year later. Based on those statements and on the observations made during the interviews it could be concluded that pupils, despite problems and some negative experiences, were satisfied and felt proud of their achievements and that they identified with the chosen task and robots.

The interviews also revealed that the cross-generational concept worked out well. Although one of the interviewees complained about very complicated questions asked by senior-citizens all other pupils said that is was great fun to carry out robotics experiments together with pre-school children and their grandparents. Kindergarten children were fascinated by the robots, asked a lot and even tried to programme robots (especially the Bee-Bot) on their own. This shows that robotics was the perfect common topic for all involved age groups and that it has great potential to bring together kindergarten children, school students and senior citizens.

Student's statements and stories told indicate that both pupils and kindergarten children gained various technical and social skills during the robotics project. Furthermore, it's also worth mentioning that three months after the robotics day all ten students who guided the experiments, decided to attend an advanced robotics workshop at Graz University of Technology.

As previously mentioned the focus of this first evaluation was put on participating young students. The next evaluation phase will include the group of kindergarten children.

B. Further feedback and observations

Next to the evaluation described in the previous section we also obtained qualitative feedback from kindergarten pedagogues, grandparents, parents and pre-school children. In sum the feedback was mainly positive. For instance some parents reported that both children and their grandparents were motivated to build robots on their own after participating in the robotics day (i.e. using Lego Mindstorms). One teacher told about a child with special needs which also participated in the robotics day. The day after both the child's occupational therapist and it's psychologist noticed a significant improvement of it's behaviour. In addition kindergarten pedagogues reported that children were very enthusiastic about their first robotics-experience and still, half a year later, talking about the robots, asking *"when they will return"*.

In order to collect qualitative data directly at the robotics day, techniques of participant observation were applied [31]. We used both passive as well as active participation methods (field notes, informal interviews, discussions). In addition we also took pictures and videotaped the experiments. Considering ethical and legal aspects all collected data was treated confidentially. Beforehand parents were informed and asked for their permission to take pictures and to videotape experiments. Gathered data is still being analyzed, further findings will be published and discussed at a later date.

[2]In this context it is important to mention that the native language of all participants (pupils, children, teachers, senior citizens) was German since the robotics day took place in Austria.

[3]Pupils' native language was German.

(a) Color sensor (sorting color bricks)

(b) Light sensor (detecting the edge of the table)

(c) Sound sensor (following the sound)

(d) Ultrasonic sensor (detecting and crabbing the can)

Fig. 3: Experiments focusing on different types of sensors

VI. Conclusions and Future Work

Science and technology develop rapidly. In order to prepare children it is important to familiarize them already in kindergarten with science and technology. In this paper a novel concept for integrating robotics in kindergartens has been presented. The cross-generational, multi-institutional robotics project combined different robotics platforms in order to address kindergarten children, school students as well as senior citizens. Different scientific and educational institutions cooperated and organised the first robotics experiment day at a kindergarten. Children, pupils, senior citizens and visitors together explored eleven different hands-on robotics experiments.

The paper also discussed preliminary qualitative evaluation results. Within the plan to evaluate the cross-generational robotics project the first step was to investigate the impact on the age group of participating pupils. Pupils who guided the robotics experiments were interviewed in order to obtain positive and negative feedback as well as to perform a first investigation on the learning effects. Furthermore, qualitative feedback from kindergarten pedagogues, grandparents, parents and pre-school children was obtained. For latter analysis field notes and videos were made and pictures were taken during the robotics day. Preliminary results of a first data analysis indicate that using robots as pedagogical tools in kindergartens could be one way to achieve the goal of familiarizing kindergarten children with science and technology in a playful way.

All collected data of the first robotics day is still being analysed. In order to refine and improve the contents of the kindergarten robotics day presented in this paper further interviews with participating children as well as teachers and kindergarten pedagogues will be conducted. Further steps also include the investigation of the impact on the group of kindergarten children. Therefore both qualitative and quantitative evaluation methods will be applied. Based on the findings and on the lessons learned from the first robotics day further project days in different kindergartens in Austria will be organized. In addition a more detailed quantitative and qualitative evaluation on the medium- and long-term impact of such robotics days in kindergartens will be conducted.

Acknowledgements

The work has been partly funded by the European Fund for Regional Development (EFRE), the federal government of Slovenia and the Land Steiermark under the Tedusar grant. The robotic day at the Kindergarten Rosental was a module from the series "Children visit Science" in cooperation with the Interdisciplinary Center for Teaching Methodology at the University of Teacher Education Styria.

References

[1] David Obdrzalek, editor. *Proceedings of the 3rd International Conference on Robotics in Education (RiE 2012)*, Prague, Czech Republic, September 2012.

[2] E. Sklar. A long-term approach to improving human-robot interaction: Robocupjunior rescue. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, 2004.

[3] M. Moro and D. Alimisisi, editors. *Proceedings 3rd International Workshop 'Teaching robotics, teaching with robotics'*, Riva del Garda, Italy, April 20 2012.

[4] Ansgar Bredenfeld, Alexander Hofmann, and Gerald Steinbauer. Robotics in education initiatives in europe - status, shortcomings and open questions. In *International Workshop 'Teaching robotics, teaching with robotics', SIMPAR 2010*, Darmstadt, Germany, November 2010.

[5] Dimitris Alimisis and Chronis Kynigos. Constructionism and robotics in education. In Dimitris Alimisis, editor, *Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods*. School of Pedagogical and Technological Education (ASPETE), 2009. ISBN 978-960-6749-49-0.

[6] Marian Petre and Blaine Pricer. Using robotics to motivate back door learning. *Education and Information Technologies*, 9(2):147–158, 2004.

[7] Fausto Ferreira, Ana Dominguez, and Emanuele Micheli. Twitter, robotics and kindergarten. In M. Moro and D. Alimisisi, editors, *Proceedings of 3rd International Workshop 'Teaching robotics, teaching with robotics. Integrating Robotics in School Curriculum*, Riva del Garda, Italy, April 20 2012.

[8] Marina U. Bers, Iris Ponte, Catherine Juelich, Alison Viera, and Jonathan Schenker. Teachers as designers: Integrating robotics in early childhood education. *Information Technology in Childhood Education*, 2002(1):123–145, 2002.

[9] Pekarova Janka. Using a programmable toy at preschool age: Why and how? In *Teaching with robotics: didactic approaches and experiences. Workshop of International Conference on Simulation, Modeling and Programming Autonomous Robots (SIMPAR 2008)*, 2008.

[10] Kerstin Stoeckelmayer, Michael Tesar, and Alexander Hofmann. Kindergarten children programming robots: A first attempt. In *Proceedings of 2nd International Conference on Robotics in Education (RiE 2011)*, Vienna, Austria, September 2011.

[11] Y. B. Kafai, K. A. Peppler, Q. Burke, M. Moore, and D. Glosson. Froebels forgotten gift: Textile construction kits as pathways into play, design and computation. In *9th International Conference on Interaction Design and Children*, Barcelona, Spain, June 2010.

[12] Alexander Hofmann and Gerald Steinbauer. The regional center concept for robocupjunior in austria. In *First International Conference on Robotics in Education*, Bratislava, Slovakia, 2010.

[13] IMST. Kinder reisen durch die Wissenschaft. Report, 2011. Report by the IMST initiative (Innovations make schools top).

[14] Sabine Hirschmugl-Gaisch, Hans Eck, and Helmut Jungwirth. Kinder reisen durch die wissenschaft. In *Fachtagung fuer elementare Bildung*, Graz, Austria, September 2011.

[15] Hans Eck and Werner Gaggl. IMST Regionales Netzwerk Steiermark, Bericht 2011/2012. Final annual report, 2012.

[16] Sabine Hirschmugl-Gaisch and Hans Eck. *Von magischen Koepfen und leuchtenden Wanzen*. 2012.

[17] Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, 1993.

[18] Stassini Frangou, Kyparissia Papanikolaou, Liliane Aravecchia, Luc Montel, Silviu Ionita, Javier Arlegui, Alfredo Pina, Emanuele Menegatti, Michele Moro, Nello Fava, Stefano Monfalcon, and Irene Pagello. Representative examples of implementing educationa robotics in school based on the constructivist approach. In *Workshop Proceedings of SIMPAR 2008; Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS*, pages 54–65, Venice, Italy, November 2008.

[19] Marjo Virnes and Erkki Sutinen. Topobo in kindergarten: educational robotics promoting dedicated learning. In *Proceedings of the 17th International Conference on Computers in Education*, Hong Kong, November 2009.

[20] Edwar Romero, Annette Lopez, and Omar Hernandez. A pilot study of robotics in elementary education. In *10th Latin American and Caribbean Conference for Engineering and Technology*, Panama City, Panama, July 2012.

[21] Rudolf Messner. *Schule forscht: Ansaetze und Methoden zum forschenden Lernen*. edition Koerber-Stiftung, 2009.

[22] Sandra Masemann and Barbara Messer. *Improvisation und Storytelling in Training und Unterricht*. Beltz, 2009.

[23] Martin Textor. *Bildungs- und Erziehungspartnerschaft in Kindertageseinrichtungen*. Books on Demand, 2011.

[24] Seung Han Kim and Jae Wook Jeon. Programming lego mindstorms nxt with visual programming. In *International Conference on Control, Automation and Systems*, Seoul, Korea, October 2007.

[25] David Grunberg, Robert Ellenberg, Youngmoo E. Kim, and Paul Y. Oh. From robonova to hubo: Platforms for robot dance. *Progress in Robotics; Communications in Computer and Information Science*, 44:19–24, 2009.

[26] Giorgio Grisettiy, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.

[27] Siw Elisabeth Hove and Bente Anda. Experiences from conducting semi-structured interviews in empirical software engineering research. In *11th IEEE International Software Metrics Symposium (METRICS 2005)*, 2005.

[28] Martin Kandlhofer, Gerald Steinbauer, Petra Sundstroem, and Astrid Weiss. Evaluating the long-term impact of robocupjunior: A first investigation. In *Proceedings of the 3rd International Conference on Robotics in Education (RiE 2012)*, Prague, Czech Republic, September 2012.

[29] Uwe Flick, Ernst von Kardorff, and Ines Steinke. *A Companion to QUALITATIVE RESEARCH*. SAGE Publications, 2004.

[30] Kimberly A. Neuendorf. *The Content Analysis Guidebook*. SAGE Publications, 2002.

[31] Danny L. Jorgensen. *Participant Observation: A Methodology for Human Studies*. Sage Publications, 1989.

Educational Robotics for Promoting 21st Century Skills

Amy Eguchi
Division of Education
Bloomfield College
Bloomfield, NJ, USA
amy_eguchi@bloomfield.edu

Abstract— **This paper introduces an educational robotics course offered as one of the Interdisciplinary Studies Courses under General Education category at a liberal art college that serves predominately underprivileged population of students from neighboring communities in New Jersey. It also presents the case study to examine participated students' learning from the course. The results show that, although the focus of the course is the educational robotics and programming to control robots created with LEGO Mindstorms, the students identified their learning of collaboration and cooperation skills as well as communication skills as one of the best learning outcomes from the course.**

Keywords—robotics in education; 21st century skills; Introduction

I. Introduction

This paper presents a case study of an undergraduate level educational robotics course offered as one of the Interdisciplinary Studies Courses under General Education category at a liberal art college. Bloomfield College serves predominately underprivileged population of students from neighboring communities in New Jersey, such as Newark and Oranges. Since Fall 2006, the educational robotics course has been offered every semester (twice a year) for last 6 years. After the first offering, students' learning during the educational robotics course was evaluated using the instructor's observation of student's in-class activities and reflective essays focusing on their learning experience. This paper aims to report on the students' learning especially with regard to their learning of 21st Century Skills.

II. 21st Century Skills

21st Century Skills have been the focus of educational reform in several countries including the U.S., Australia, Finland and Singapore. Especially in U.S., the focus on 21st Century Skills is the core of the educational reform. The Partnership for 21st Century Skills, a national organization (http://www.p21.org/) that advocates for 21st century readiness for every student, states:

> In an economy driven by innovation and knowledge ... in marketplaces engaged in intense competition and constant renewal ... in a world of tremendous opportunities and risks ... in a society facing complex business, political, scientific, technological, health and environmental challenges ... and in diverse workplaces and communities that hinge on collaborative relationships and social networking ... the ingenuity, agility and skills of the American people are crucial to U.S. competitiveness [2].

The Partnership for 21st Century Skills focuses on the 21 Century Skill Framework, which identifies 21st Century student outcomes and skills:

Core Subjects and 21st Century Themes:
- *Core Subjects – English, World languages, Arts, Mathematics, Economics, Science, Geography, History*
- *21st Century Themes – Global awareness; Financial, economic, business and entrepreneurial literacy; Civic literacy; Health literacy*

Learning and Innovation Skills:
- *Creativity and innovation skills*
- *Critical thinking and problem solving*
- *Communication and collaboration skills*

Information, Media and Technology Skills:
- *Information literacy*
- *Media literacy*
- *ICT*

Life and Career Skills:
- *Flexibility and adaptability*
- *Initiative and self-direction*
- *Social and cross-cultural skills*
- *Productivity and accountability*
- *Leadership and responsibility*

Among those skills, 4Cs (Critical thinking and problem solving, Communication, Collaboration, and Creativity and innovation) are core skills for our students to be successful in the future.

Assessment & Teaching of 21st Century Skills, another organization with international collaboration based in Australia (http://atc21s.org/), organizes 21st Century Skills into four broad categories as follows:

- **Ways of thinking.** Creativity, critical thinking, problem-solving, decision-making and learning
- **Ways of working.** Communication and collaboration
- **Tools for working**. Information and communications technology (ICT) and information literacy
- **Skills for living in the world.** Citizenship, life and career, and personal and social responsibility

Both organizations emphasize the importance of creativity, critical thinking, communication and collaboration (4Cs) as key of success in the 21st century. In next section, why educational robotics help promote 21st century skills among young students is explained.

III. ROBOTICS IN EDUCATION (RIE)

Educational use of robotics for school-aged children has been around for more than a decade. However it has been observed in the last several years that popular interest in robotics has increased astonishingly [2]. In addition, the availability of robotics for both post-secondary level education and school-aged children is growing rapidly [3 and 4]. Mataric argues that robotics has "the potential to significantly impact the nature of engineering and science education at all levels, from K-12 to graduate school" [3, para 1]. In higher education, robotics is use mostly with the courses for computer science/engineering related areas. Educational robotics tool, for example, LEGO Mindstorms set, is usually used with introductory level courses [4 and 5]. For example, Drew, Esposito et al. point out that LEGO Mindstorms, an educational robotics kit widely available around the world, has been integrated into curriculums at many higher education institutions across the world including MIT, Brown University, University of Maryland, Tufts University, University of Aarhus at Denmark, University of Utrecht in the Netherlands, Trinity College Dublin in Ireland, and University of Manchester in the UK [5]. For grades of K-12, most robotics activities are extra-curricula (i.e. after school programs and summer campus) [2, 6 and 7]. Elkind [8] points out that educational robotics open a door for helping children learn about mathematics and scientific concepts through the method of inquiry, as well as for developing technological fluency. The systematic study of scientific literature on the use of educational robotics in schools by Benitti, which focuses on quantitative results, identifies that most of the studies have focused on the fields of mathematics and physics [2]. It also indicates that the skills developed through educational robotics are thinking skills (observation, estimation and manipulation), science process skills/problem-solving approaches, and social interaction/teamwork skills. Several studies have also shown that educational robotics provides effective learning opportunities for students in both content areas such as physics, biology, geography, mathematics, science, electronics, and mechanical engineering, and also critical academic skills, such as writing, reading, research, creativity, collaboration, critical thinking, decision making, problem solving, and communication skills [6, 9 – 18].

One of the reasons why educational robotics is an effective learning tool is that educational robotics helps create a *fun* and *engaging* learning environment that keeps students interested and engaged in learning. Educational robotics is *fun* because it provides *hands-on* learning experience. Also, it is a great *tool* for project-based learning. With project-based learning, students work in groups to "explore real-world problems and challenges. With this type of active and engaged learning, students are inspired to obtain a deeper knowledge of the subjects they're studying" [19]. Educational robotics creates a great environment for students to encounter and developed solutions for *real-world* problems and to demonstrate their learning through the robots they developed. Following section introduces the "Educational Robotics as Learning Tool" course, which has been offered as a general education course.

IV. EDUCATIONAL ROBOTICS AS LEARNING TOOL

"Robotics as Learning Tool" course was first offered in Fall 2006. Initially, the goal for offering the educational robotics course was for our pre-service teachers to learn to use this hands-on teaching tool so that they could use it in their classrooms in the future. However, the first educational robotics course was decided to be offered as an Interdisciplinary Studies Course under General Education (GE) program. The GE program at Bloomfield College offers a variety of courses aiming to foster the development of skills and knowledge needed to be successful in college and in the world. The program identifies seven areas of competence that are integrated into both the GE and major areas: Aesthetic Appreciation, Communication Skills, Community Orientation & Citizenship, Information Literacy, Multicultural Awareness, Problem-Solving & Critical Thinking Skills, and Scientific & Technological Skills. This course is structured to integrate three of the GE competencies; Communication Skills, Problem Solving & Critical Thinking Skills, Scientific & Technological Skills. Successful GE courses provide a key to success to the students at the Institution, which serves predominately Black and Hispanic population of students. Many of the students are also first generation to attend college or even first generation to graduate from high school. The institution is committed to enabling students, particularly those who have traditionally been excluded from higher education, to realize their intellectual and personal goals.

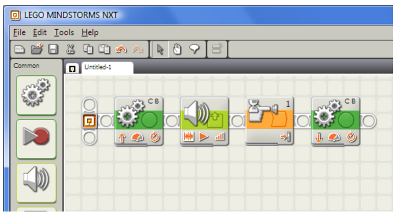
A. Course Overview

Fig. 1.LEGO NXT robot

"Educational Robotics as Learning Tool" is a semester long course that meets twice a week for one hour and forty minutes each session. LEGO Mindstorms Robotics Invention System NXT ("Mindstorms kit"; Figure 1) is used for the course.

LEGO Mindstorms was used because it is easily accessible for anyone if students wish to continue with robotics. The Mindstorms kit comes with a programmable brick, motors and several different sensors including a touch sensor, light sensor, rotation sensor, sound sensor, and ultra-sonic sensor (distance sensor). Programming can be done with PC or MAC to perform very simple tasks to very complicated and useful tasks. For programming, we use NXT-G programming Software (Education version), which is a standard programming software for the kit. NXT-G programming environment provides a simple graphic programming environment (Figure 2), not like usual text-coding programming languages. The graphic environment provides a very useful tool for teachers teaching young children and those who are new to programming since it uses a drag-and-drop function with icons that are pre-programmed for some tasks. Entities used in programming, such as motors and sensors are represented as icons in a small menu section on a computer screen. Students need to simply *drag* an icon that they want to use for their program and *drop* it on a blank space on the screen to create a *code*. This graphic environment is highly visual and provides a good first programming experience with procedural programming concepts. This is also a good programming environment for first time programmers at undergraduate institutions especially for those who are not intending to be a Computer Science major or become a programmer.

Fig 2. Simple NXT-G Program



This makes a robot go forward for 1 motor rotation, make a sound, wait till a touch sensor on port 1 is pressed, then go backward for 1 motor rotation.

With this course, throughout a semester, students work in groups, usually either as a pair or a group of three. Students in each group share one robot and one computer. The intention is to promote collaborative learning more than if each student were to use his/her own robot and computer. We start each session by review programs created by each group in the previous session followed by a discussion on how to improve the programs. This promotes collaborative learning among the whole group. After the review, the instructor introduces one or two new programming concept(s) using SmartBoard to show examples on NXT-G. Following the brief lecture, the groups receive tasks (mini-projects) that require employing the new concepts. For the rest of the session, students work in groups to complete the tasks. This allows each group to progress at their own pace. However, this arrangement creates diverse differences of progress between the groups in class and makes it hard for one instructor to provide sufficient help needed. To solve this issue, advanced students are assigned by the instructor to help other groups. This encourages everyone to start offering help when asked – another way of promoting collaboration in the classroom setting.

After sessions, students are required to write a weekly reflective journal. They are required to reflect on their learning experience using the blog feature of the institution's Blackboard. Towards the end of a semester, students work on a final project. Each semester, the final project may be different. Students spend about a month for completing the final project, which is showcased on the final day of a semester. Each student also produces a lab report of the group's final project and a written reflection of their learning throughout the semester.

B. Revision of the Course Focus to Robotics Technologies

After its inaugural semester in 2006, several revisions have been made with the course curriculum. One of the biggest revisions is its focus on current robotics technologies. As Barker, Nugent, Grandgenett and Adamchuk [20] emphasize, in recent years, "we have seen an influx of new robotic systems in a variety of areas from space explorations to agriculture, from households to manufacturing production and medical institutions, and certainly through education" (p.xix). There are States in the US (currently two – California and Nevada) that have signed a new law that legalizes autonomous cars driving on city streets [21 and 22]. In Korea, Japan and China, there are robotic teacher's aids developed to assist teachers in classroom [23-27]. This trend of rapid technological development, both with robotics and any other technologies that we use in everyday life, including smartphones, tablet and computers, indicates the need of our next generation to have interests in as well as understanding of those technologies. *Technological literacy* (understanding of technology as a user and developer) should be part of 21st Century Skills along with information literacy in the near future. For this reason, current curriculum of the course has put more emphasise on the history of robotics development and current status of the technology. This part of the curriculum begins with the introduction of personal robots (AIBO produced by Sony in 1999) to the development of humanoid robots – QRIO by Sony, ASIMO by Honda, NAO by Aldebaran, and more. This is followed by the introduction of current robotics technologies. The instructor has created an online course list with online news on cutting-edge/latest robotics technology under the Robotics Resources section of Blackboard, which serves as a depository system for the curriculum and constantly updated by the instructor. Recent year's "current robotics technologies" list includes search and rescue mission robots from Japanese nuclear disaster, an autonomous car developed by Google, Robonaut by NASA, and a robot arm controllable by human brain signals.

V. Assessment of Student Learning

For the assessment of student learning for this study, the final reflective essays from two semesters (Fall 2011 & Fall 2012) were analyzed using text coding with quasi-grounded theory. The data from only fall semesters were used because the curriculum for spring semester is slightly different from that of fall semester, which include extra curricula activity. To keep the consistency of the students' learning experience for this study, the data from the fall semesters were used for the analysis. Total of 27 students enrolled in the course for those two fall semesters (17 students in Fall 2011 and 10 students in Fall 2012). Out of 27 students, 18 students completed the final assignment electronically (10 students in Fall 2011 and 8 students in Fall 2012), which were used for the analysis. The number of students who complete a semester is usually lower than the number of enrolled students each semester because of various reasons. The results indicate that the students in Fall 2011 & Fall 2012 highlighted the following learning in their essays:

1. they learned collaboration/team work skills (100%)
2. they learned about robotics and technology - increased interests in those areas (83%)
3. they enjoyed/had fun with the course (78%)
4. they learned to be creative/think creatively (67%), and
5. they learned problem-solving skills (67%).

In following sections, first two items (collaboration/team work skills and interests in robotics and technology) are explained in detail.

A. Learning to Collaborate

Although students' learning of collaboration/team work was high in the first study which was done after the first semester (Fall 2006), the result from Fall 2011 & Fall 2012 shows that all students highlighted their learning of collaboration skills in their essays. In Fall 2006 result, it was indicated that there was one student who felt that s/he could not build collaboration skills due to the lack of time the group worked together. Because of the structure of the course that forces students to work together, it is natural for students to notice the collaboration is the focus of the course. However, for all of the students studied to discuss their learning of collaboration and team work in their final essay in length is significant. Interestingly enough, several students stated that, in general, they are not in favor of working in groups mainly because they think group work ends up *unfair*. One student expressed:

> Working in groups was something I hated the entire time I was in high school, because when I used to work in group, only a portion of the group's members would be working. What made my anger even worst is that they would get the same grade as I, and they were just sitting there doing nothing while I was working.

However, course gave her a new perspective on collaborating with others:

> Well the only reason I work with a partner in robotics was because it was required; and I learned several lessons from it. First, working with a partner makes the assignment ten times easier than it is. It makes it easier because, as partners we divided our tasks by two, and each one of us would work on something different at the same time. Secondly, I got to share my ideas with that partner and his ideas too, and we just put everything together at the end. Sometimes when I felt too sick to work, he would work for me, and I would do the same. The third lesson that I have learned from working with a partner is not a good one, and I am glad it happened in this class, so it won't happen in other classes. The problem was that my first partner did not want to work with me, but he never had the guts to tell me that. One day he got up and went to work with another group without even telling me, I only heard that I did not have a partner anymore, I got myself another one and we did way better than the one before. But I learned never to divide our tasks, but switch them often. Because if we divide our tasks for example when one of us leaves the group, the one staying won't be able to do everything because he only used to work on a specific task.

Although it was a difficult experience, it gave her a valuable lesson on collaboration. Another student also shared that this course gave him a new lesson. He states:

> The good thing about this was working in pairs (or in my case, a group of three). Working in a group was perhaps the best idea for this class. I usually prefer to work independently, and for reasons, I still stand by that notion. Yet, sometimes I caught myself in a bind with ideas on what to use and when. I would've never thought to use an ultra-sonic sensor to locate cups, and this is a fact. My partner's brains, on the other hand, seemed to know exactly what to do. It took us a long time to figure out how to get our robot around that circle without going straight, but after many trial and errors, and a little hint, it's like a light bulb went off in our heads. Working together has helped a lot to get the work done.

In the class, the instructor strongly emphasizes and encourages students to help each other. She frequently asks advanced students to teach others. It is not only because she cannot provide help to eight to ten groups at once but also because this strategy gives students the valuable lesson of collaboration. One student explained his experience:

> There were many classes where we had to help out the rest of the groups and give out hints and advice whatever project we had to do. We even went so far as troubleshoot their work so we could tell them what was wrong or where their mistake was, so that they could go back and figure out the issue and fix it. For instance, the week of October 25-27, we acted as teachers to the class. We had to help the class with pushing cups out of a circle, without leaving the circle, until all the cups were removed from the inside of the circle. Then we had to help them understand how to make the robot follow black circle without ever going straight.

Those are one of the difficult programming tasks, which they successfully taught the other groups. This indicates their mastery of the programming skills required in class since teaching is the highest form of learning. The collaborative environment that this course provides the students not only helps them to learn collaboration skills but also enhance their learning.

B. Promoting Interests in Robotics and Technological Literacy

The result shows that the additional focus on the development of robotics and robotics related technology in the course after recent revision has proven to enhance the interests in robotics and robotics. Majority of the students (78%) highlighted their learning of robotics and robotics related technology in their essays. One student expressed her discovery:

> Using the NXT's [LEGO Mindstorms] helped out to understand a little bit about electronic devices. Majority of people take for granted the many machines around them. It's not on purpose though. It's a simple, pure lack of knowledge on people's part. The more technology that is created, the less people understand it.

Prior to taking this course, robots, for many of the students, were things in science fiction movies. One student described:

> I learned a lot about technology and robotics as a whole. I've always thought of robots as technology of the future. Most people think of robots as *future human beings*. The fact is most robots today are here to do jobs human beings used to do.

Another also commented:

> I really did not think much of robots till I started the class. I thought robots were like the robots in the movies but I was wrong. There is more to robots than just them taking over the world and being the bad guys. Robots are more than that. Robots are the future and will help us in life. Robots are good things and they are good for us.

Because of the influences of science fiction movies, some students expressed that they believed that robots in the future would be like human beings or even take over the world. At the end of the semester for them to be able to state "I learned that some of these science fiction movies that I would watch that involved robot were somewhat impossible" is quite important learning. Another student added:

> The notable piece of the situation is that the human has complete control over the robot at all times, the robot is not permitted to freely think, it acts as an extension of the human. … As much as a fear of robots becoming aware may be irrational it is something that is implanted into all of us courtesy of Hollywood. It raises interesting questions and reminds us of how fragile we are compared to the machines we are capable of building. However, I am definitely excited to see where the discoveries being made with the assistance of robots and where the ever-changing world of robotics will take humanity.

VI. Future Considerations

The overall results suggest positive learning experience among the students through this course. Although this course teaches the programming of LEGO Mindstorms robots, what the course aims to teach students is above the content knowledge that this course provides. This course is more or less a content-less course targets more on students' skill acquisition – skills that they need to be successful as 21^{st} century citizen. All the targeted skills – collaboration/team work skills, communication skills, creative thinking, and critical thinking/problem-solving skills (4Cs) that the course focuses on are visible for students through the reflection of their learning experience. Although their prior experiences with group work both from their primary & secondary education, and college seem to be negative, having the positive collaborative learning experience through this course has changed their perception of collaboration. Follow up study on how their perception of collaboration will influence their performance in their future courses will help us understand the impact from this course better.

From the study, the other aim of the course to open their mind and interests in robotics and robotics technology and emhance students' technological literacy have appeared to be successful as well. One of the students also commented:

> I now find out robotics is an interesting tool because the way the course ended left me open minded about some things in the robotic field of life.

One student in particular changed his major from Social Study – Criminal Justice to Engineering to pursue his career in Industrial Engineering because of his learning through the course. He expressed his learning:

> Finally, at the end of my first semester as a college freshman, I can say that this course has really opened my eyes up to the world and what is to be expected from this generation of college graduates. Innovation is what is going to drive the economy and the very way of life.

I hope this course will continue to have such an impact on our students especially because our institution serves predominantly Black and Hispanic population of students and those are the population whose representation in STEM field needs to be increased.

This course happened to be a course that students feel uncertain about their learning adventure at the beginning of a semester. Sixty seven percent of students from Fall 2011 and Fall 2012 described that they did not know what they were going into at the beginning of the semester, and 78% of the students enjoyed the course. One student summarized the experience:

> When I first began the semester, I honestly believed that I was not going to enjoy my Robotics class. I began the class with the mentally of just doing what I had to do and leaving. The truth is, I learned many great things in this class and I enjoyed it very much.

Although at times, the learning could be very frustrating as about half of the students expressed in their reflection, when they try to excel while having fun with their learning, the overall outcome becomes positive. The challenge for educators is to find the best way for each one of us to create such a fun learning experience for our students. Educational robotics is one of the best learning tools for creating such learning experience for students. This course is a course offered at an undergraduate institution; however, the author also teaches the same curriculum to grade school students. It might be interesting to study younger students' perception of their learning experience of 21st Century Skills through educational robotics in the future.

REFERENCES

[1] Partnership for 21st Century Skills, "21st Cenutry Skills, Education & Competitiveness Guide – A Resource and Policy Guide,"
Retrieved April 15, 2013 from http://www.p21.org/storage/documents/21st_century_skills_education_and_competitiveness_guide.pdf, 2008.

[2] F. B. V. Benitti, "Exploring the educational potential of robotics in schools: A systematic review," Computers & Education **58**: 978-988, 2012.

[3] M. J. Mataric, "Robotics Education for All Ages," American Association for Artificial Intelligence Spring Symposium on Accessible, Hands-on AI and Robotics Education, 2004.

[4] A. Cruz-Martin, & J. A. Fernandez-Madrigal, et al., "A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses," Computers & Education **59**: 974-988, 2012.

[5] J. Drew, J. & M. Esposito, et al., "Utilization of Robotics in Higher Education," The 21st Annual Information Systems Education Conference (ISECON 2004), New Port, RI, EDSIG and Foundation for Information Technology Education, 2004.

[6] E. Sklar and A. Eguchi, "RoboCupJunior - Four Years Later." In Proceedings of RoboCup-2004: Robot Soccer World Cup VIII, 2004.

[7] A. Eguchi, "Educational Robotics for Elementary School Classroom," in Proceedings of the Society Information Technology and Education (SITE), San Antonio, TX, AACE, 2007.

[8] D. Elkind, "Forward," Block to Robots. M. U. Bers. New York, NY, Teachers College Press: xi-xiv, 2008.

[9] E. Kolberg, E. and N. Orlev, "Robotics Learning as a Tool for Integrating Science-Technology Curriculum in K-12 Schools," in Proceedings of the 31st ASEE/IEEE Frontiers in Education Conference. Reno, NV, 2001.

[10] D. Oppliger, "Using FIRST LEGO League to Enhance Engineering Education and to Increase the Pool of Future Engineering Students," in Proceedings of 32nd ASEE/IEEE Frontiers in Education Conference, 2002.

[11] E. Sklar, A. Eguchi, and J. Johnson, "Examining the Team Robotics through RoboCupJunior." In the Proceedings of the Annual Conference of Japan Society for Educational Technology, 2002.

[12] E. Sklar, A. Eguchi, and J. Johnson, "Scientific Challenge Award: RoboCupJunior - Learning with Educational Robotics." AI Magazine 24(2): 43-46, 2003.

[13] M. Carbonaro, M. Rex, et al. (2004). "Using LEGO Robotics in a Project-Based Learning Environment." Interactive Multimadia Electronic Journal of Computer Enhanced Learning 6(1), 2004.

[14] I. R. Nourbakhsh, I. R., E. Hamner, et al., Formal Measures of Learning in a Secondary School Mobile Robotics Course, In the Procssings of 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, IEEE Computer Society, 2004.

[15] S.Atmatzidou and S. Demetriadis, Evaluating the role of collaboration scripts as group guiding tools in activities of eduational robotics, in the Proceedings of 2012 12th IEEE International Conference on Advanced Learning Technologies, Rome, Italy, IEEE Computer Society, 2012.

[16] D. P.Miller, I. R. Nourbakhsh, et al., Robots for Education, Springer Handbook of Rootics, B. Siciliano and O. Khatib (Eds). New York, NY, Springer-Verlag New York, LLC: p. 1283 – 1301, 2008.

[17] D. Alimisis and C. Kynigos, "Constructionism and Robotics in Education," Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods. D. Alimisis. Athens, Greece, School of Pedagogical and Technological Education, 2009.

[18] S. Atmatzidou and S. Demetriadis, "Evaluating the role of collaboration scripts as group guiding tools in activities of eduational robotics," in Proceedings of 2012 12th IEEE International Conference on Advanced Learning Technologies, Rome, Italy, IEEE Computer Society, 2012.

[19] Edutopia, "Problem-based learning." Retrieved October 20, 2012, from http://edutopia.org/php/keyword.php?id=037.

[20] B. Barker, G. Nugent, et al., "Preface," Robotis in K-12 Education: A New Technology for Learning, B. Barker, G. Nugent, N. Grandgenett and V. I. Adamchuk (Eds), Hershey, PA, Information Science Reference (IGI Global)**:** xviii-xxiv, 2012.

[21] J. Belezina, "Nevada approves regulations for self-driving cars," Gizmag, retrieved March 20, 2012 from http://www.gizmag.com/nevada-autonomous-car-regulations/21507/.

[22] Lavrinc, D. (2012). "Autonomous Vehicles Now Legal in California," retrieved from October 20, 2012 from http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDAQFjAA&url=http%3A%2F%2Fwww.wired.com%2Fautopia%2F2012%2F09%2Fsb1298-signed-governor%2F&ei=Z3VsUY7ZBqrq0AHpv4CgCw&usg=AFQjCNGTrLwVM287cY7SS7whvQPPUo34pw&bvm=bv.45175338,d.dmQ

[23] C-.W. Chang, J.-H. Lee, et al.,"Exploring the Possibility of Using Humanoid Robots as Instructional Tools for Teaching a Second Language in Primary School " Educational Technology & Society 13(2): 13-24, 2010.

[24] G.-D. Chen and C-W. Chang, "Using humanoid robots as instructional media in eleme.ntary language education," in Proceedings of 2008 Second IEEE International Conference on Digital Games and Intelligent Toys Based Education, Banff, BC, IEEE, 2008.

[25] J-E. Chen, L.-T. Yeh, et al., "Development of an Emotional Robot as a Teaching Assistant," in Proceedings of Edutainment, Banff, Canada, Springer-Verlag Berlin Heidelberg, 2009.

[26] Demetriou, D., "Robot teacher conducts first class in Tokyo school. The Telegraph" London, UK, The Telegraph Media Group Ltd., May 2009, retrieved March 20, 2013 from http://www.telegraph.co.uk/technology/5311151/Robot-teacher-conducts-first-class-in-Tokyo-school.html.

[27] J.-H. Han and D.-H. Kim, "Physical Learning Activities with a Teaching Assistant Robot in Elementary School Music Class," Journal of Convergence Information Technology **5**(5): 30-38, 2010.

Exploring Spatial Relationships: A Strategy for Guiding Technological Problem Solving

Dorothy Langley
Davidson Institute of Science Education
& Holon Institute of Technology
Israel
Langley@hit.ac.il

Yair Zadok
School of Education,
College for Academic Studies
Or Yehuda, Israel
yair_z@mla.ac.il

Rami Arieli
Davidson Institute of Science Education
Weizmann Institute of Science
Rehovot, Israel
Rami.Arieli@Weizmann.ac.il

***Abstract*— This paper refers to engaging high school physics and computer science majors in challenging design projects which seek to activate and implement the often inert formal content knowledge within the context of designing and constructing systems dealing with real world engineering challenges in robotics and electro-optics. The growing realization of the benefits to individual students and to state economies, of providing science learners with opportunities to expand their knowledge, skills and experience of knowledge-based technological design has led to seeking instructional strategies to facilitate the transition from traditional school settings to project based learning environments. In this paper we suggest that visualization of the problem space and guided exploration of its spatial relationships can promote the elicitation of relevant formal knowledge and lead to creative solution design. These methods are described in the context of designing and programming robot navigation and in the context of developing remote distance sensors.**

Keywords—project based learning; visualization; spatial relationships; guidance; robot navigation; non-contact distance sensor; inert knowledge

I. Introduction

There is wide agreement that the science and technology education should develop learner abilities for problem solving, independent critical thinking, creative and inventive skills and productive team work. Project-based science and technology learning provides a natural environment for achieving such goals, thus enabling the shift from traditional, teacher centered instruction towards meaningful, student centered learning [1, 2, 3]. The recently published New Generation Science Standards has embraced these insights, and includes an explicit focus on Engineering Design [4].

Many programs promoting project-based-learning in the context of engineering design for high school students, have been initiated in different countries during the past decade [e.g. 5, 6, 7]. Many of these programs are based in institutions of higher education. This may be due to the realization that engineering is vital for national economies and that high school students should be encouraged to join departments of engineering in colleges and universities. Robotics courses naturally fall into this category of instructional activities. Some high school curricula include Robotics as a study domain [e.g. 8].

Abundant evidence has been gathered showing that students involved in electronics and robotics projects display high motivation to confront complex tasks and show originality and creativity [9, 10].

This paper refers to engaging high school physics and computer science majors in challenging design projects which seek to activate and implement the often inert theoretical content knowledge [11] within the context of designing and constructing systems dealing with real world engineering challenges. The examples will be taken from two programs within the School of Contemporary Science run by the Davidson Institute for Science Education, in the Weizmann Institute of Science: Physics and Industry[1] (Appendix), and Computer Sciences, Academia and Industry[2].

II. The Challenges of Project based Education

Traditional k-12 education provides very few opportunities for science learners to develop attitudes, skills and knowledge which are necessary for confronting technological problems and succeeding in project-based learning. Thus, even high ability learners lack the habits of mind and the practical skills for designing a technological system and progressing towards producing a working artifact. There is a growing realization that learning and creative thinking are complex processes. The linear project design prescriptions commonly found in the engineering and technology literature, are poor approximations of the ways these processes unfold in reality, or of the work methods of experts [12] . To this one needs to add the realization that problem-solving and thinking skills are context-bound. Thus, very little transfer can be expected between problems in different domains [13]. Swartz & Perkins [14] stress that learners need direct instruction regarding the cognitive skills required for problem solving in a specific domain. We should not expect that formal knowledge acquired in physics or geometry lessons will be automatically invoked in the context of programming robotic motion or designing remote-distance sensors. It is the instructors' responsibility to

[1] http://davidson.weizmann.ac.il/en/content/physics-and-industry
[2] http://davidson.weizmann.ac.il/en/content/computer-sciences-academia-and-industry

involve the learners in activities intended to elicit relevant formal knowledge.

III. VISUALIZATION AS A PROBLEM-SOLVING STRATEGY FOR TECHNOLOGICAL PROJECTS

Visualization is one of the powerful strategies for problem solving in general, and creative technological problem solving, in particular. Visualization involves transforming a physical situation or event into an explicit verbal or graphical representation, in a form that helps define the problem and promotes progress to one or more solutions [15]. We intend to show how exploring and explicating spatial relationships can be instrumental in understanding a technological problem and designing creative solutions [16].

Spatial ability is a collection of specific skills related to the ability to receive, mentally process and spatially present information [17]. Spatial thinking includes the following stages in visual information processing: 1. Visual perception of object having spatial properties. 2. Mental rotation. 3. Physical or visual representation of spatial information - Visualization. In the following we shall refer to "**spatial relationships**" as the manner in which spatial properties are related to each other.

Spatial abilities are considered particularly relevant for learning mathematics, science and technology at school and a vital condition for vocational education and engineering careers [e.g. 18, 19]. Researchers believe that spatial abilities can be developed through drill and experience [20]. Educators have suggested advanced technological environments for developing spatial abilities in the fields such as mathematics, science, engineering and medicine [e.g. 21, 22].

Visual spatial abilities are often framed within the domain of 3D perception [e.g. 23, 24]. However, basic plane geometry offers a rich arena for developing an understanding of spatial relationships. Experiential visual perception allows children to identify circles and triangles, but visual experience by itself is unlikely to lead to the formulation of the many relationships that exist between lines and angles. Formal instruction in mathematics and geometry is responsible for teaching some basic spatial relationships such as Pythagoras' theorem; sum of the internal angles in a triangle; ratio of the circle circumference to its radius; ratio between the sides of similar triangles; ratio of the sides of a right angle triangle; definitions of geometric loci. Likewise, formal instruction in physics is responsible for teaching basic relationships related to motion, light propagation and electrical properties. All this formal knowledge is usually represented in concise "formulas", which high school students use extensively in text book problem solving. The challenge for project instructors is activating this stored knowledge in the context of designing technological problem solutions.

IV. EXPLORING SPATIAL RELATIONSHIPS: THE CASE OF REMOTE DISTANCE MEASUREMENT

The need for reliable non-contact distance measurement exists in a variety of science and technology fields such as traffic control, robotic navigation, automated manufacturing, vehicle safety, helping the visually impaired, astronomy and astrophysics, land surveying, acoustic design and ballistics. Non-contact distance measurement is required in many of the students' projects. The instructional design involves applying structured thinking skills and activating formal knowledge from physics and mathematics, as the following sequence will show.

A challenge is presented to the students as a trigger for thinking about the problem of remote distance measurement.

"You are standing on the bank of a wide river, and you need to measure its width without physical contact with the opposite side. You are free to walk along the river bank and you can see a tree which is growing near the opposite side. You have a meter ruler and a protractor. "

1. The initial step is a two dimensional visualization of the key physical elements of the physical situation: the river, with the observer and the tree on opposite banks. To this we add a line representing the **unknown distance** "river width" (Fig. 1a).

2. The next step in the problem analysis involves explication of the meaning of "seeing" as receiving visual information (light) from a remote object. The concept of "line of sight" and the notion of light traveling along straight lines are invoked.

3. The virtual line of sight is added to the visualization, connecting the tree and the observer. The line forms an angle alpha with the river bank. Now, the triangle is visualized, indicating tangible and virtual properties. (Fig. 1b)

Fig. 1a

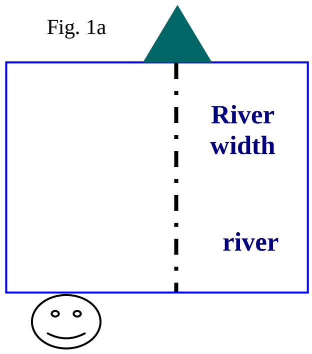
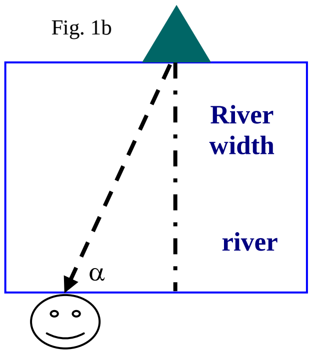


Fig. 1b



4. The following discussion is "How can we measure the angle alpha?" This will involve using the meter ruler to concretize the line of sight and the protractor to measure the angle. (Fig. 1b)

Fig. 1c

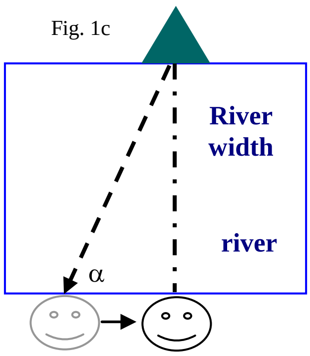
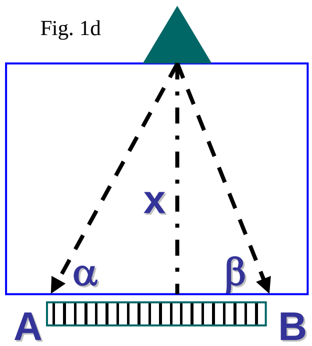


Fig. 1d



$X = AB/(1/\tan\alpha + 1/\tan\beta)$

5. What additional information do we need to calculate the river width? The observer needs to move a known distance along the river bank, and repeat the previous process. The simplest option is to move to a spot directly opposite the viewed object. (Fig. 1c)

6. The general solution for any two points of observation (Fig. 1d).

A. Sample implementation in Project Models

A laser beam is directed at a rotating plane mirror, placed at a known distance L (Fig. 2). The light is reflected, the angle of deflection being α. The reflected beam hits the target, and is reflected diffusely. A directional detector collects light at right angles to the original laser beam. When the detector sends a signal, **x= L* tan α**

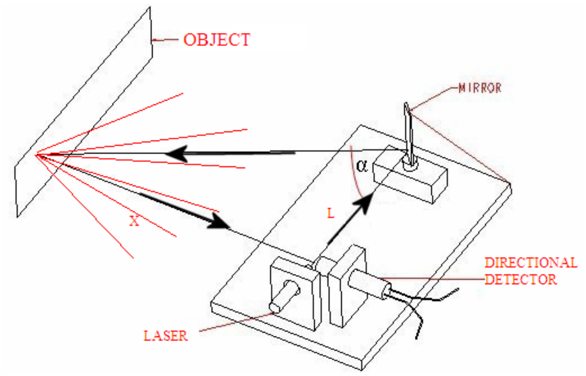


Fig. 2. Sample set up for remote distance measurement

1) Constructing a working sensor prototype

The system measures the distance (**x**) to an object to which there is a clear line of sight. In the given system, L is a constant determined by the system design requirements. Measuring different distances requires controlling the deflection angle of the laser beam (α). The laser beam is reflected from a small, front surfaced, plane mirror which is connected to the axis of a stepper motor which moves in a stepwise manner, advancing at a prefixed angle in response to electrical pulses[3]. This rotation should continue until a signal is received from the detector. The deflection angle (α) is twice the angle at which the stepper-motor has advanced from the initial reference position.

Progressing from the conceptual scientific solution to the working technological system necessitates adding a programmable interface which collects information from the designed sensors and produces visual displays and operational outputs which can drive external components (e.g. stepper motors, light sources, buzzers). In our projects we have incorporated the Arduino programmable processor board[4]which can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.

2) Measurement Algorithm

The stepper motor has a pre-set single step angle. It is controlled by a driver board (hardware and software) which receives the electrical pulses. The Arduino control board is programmed to supply electrical pulses to the stepper motor driver board, and count the number of supplied pulses. The detector system is connected as input to the Arduino board. Each time the Arduino receives a signal from the detector, the angle of deflection (α) and the related distance (x) are calculated. Repeated distance measurement of a moving object can be used for measuring velocity (speed and direction) and acceleration.

B. Different solutions based on similar triangles

Lens images can be used for determining the distance to a distant object with a known dimension, such as distance between car headlamps (Fig. 3). Geometric optics is taught in the 10th grade and students can be expected to "know" the relationship between the positions of objects and their images.

In a realistic situation, the end points of the object either emit light or reflect ambient light. A thin converging lens is placed so that the principal axis is at right angles to the object. The images of the extreme points are created on the focal plane. The object distance X is related to the distance between the extreme images (hi):

$$\mathbf{X = f * ho / hi}$$

$\mathbf{h_i}$ can be measured manually or by using a detector array placed along the focal plane. By identifying the extreme detectors that respond at a given moment, it is possible to measure and record changes in the distance.

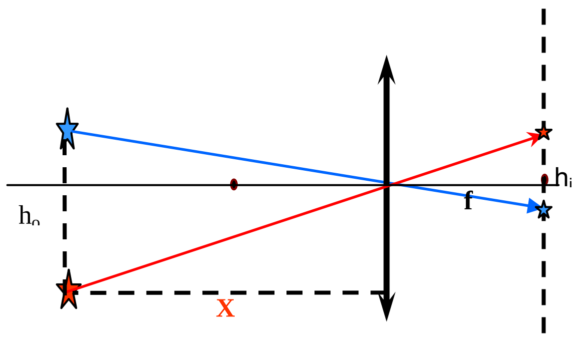


Fig. 3. Visualizing the spatial relationship between the size of the distant object and the size of the image on the focal plane.

V. EXPLORING SPATIAL RELATIONSHIPS: THE CASE OF LEGO ROBOT NAVIGATION

The Lego robot[5] advances using two wheels which are powered by separate motors (Fig. 4). The motors can be programmed to rotate at different rates in a forward or reverse direction. The robot's motion is determined by programming instructions related to traveled distance and the rotation rate of the driving motors, as well as by signals received from activated sensors.

[3] http://www.solarbotics.net/library/pdflib/pdf/motorbas.pdf
[4] http://www.arduino.cc

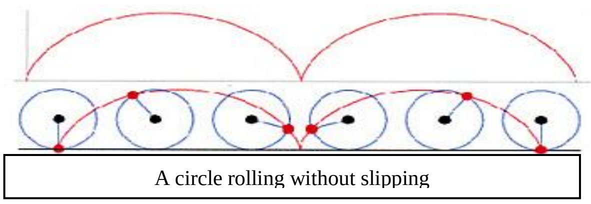
[5] http://firstlegoleague.org/productEV3

Fig. 4. Sample Lego Robot

The system records the total rotation angle of each motor. Programming the robot motion and calculating distances, necessitates an explicit understanding of the spatial relationships related to circles and the physical relationships between translation and rotation. Novice robot programmers tend to adopt a "try it and see" trial and error strategy, to achieve their goals. Robot project instructors can use the following sequence to promote deeper thinking about the relationship between the motion of the wheels and that of the entire robot.

A. *Advancing in a straight line*

When a circle rotates around the point of contact with a line, the centre advances a distance equal to the circumference for each completed rotation. This is called "rolling without slipping" (Fig. 5)[6].



To achieve motion along a straight path, both motors must rotate at the same rate. To calculate the traversed distance, the reading of the motor angle counter is read and the value is fed into the appropriate cell, divided by 360 and multiplied by the wheel circumference (2πr).

Fig. 5. Visualization of a wheel rotating and advancing

	Rotation angle	Wheel circumference (cm)	Distance covered
Both wheels	Θ°	2π*3.5	$\Theta^{\circ}\cdot\frac{7\pi}{360}=\Theta^{\circ}\cdot 0.061$

The instructional sequence starts with students observing the straight line motion of a pre-programmed robot. The next step is a graphical visualization of the motion of a wheel and a formal expression of the relation between the rotation angle and the distance traveled. After the students copy the instructor's program into their robots, and write a procedure for displaying the distance - they realize that the actual path differs from the theoretical calculation due to friction and other imperfections.

[6] http://rgbhodoi.blogspot.co.il/2011/12/going-down-fast.html

B. *Changing direction - making a 90^{o} turn*

Robot navigation requires changing direction and turning corners. There are two options for achieving a 90^{o} turn: 1. Using one stopped wheel as a pivot, and the distance between the wheels as the radius.

	Distance covered	Rotation angle
Outer wheel	$\frac{2\pi R}{4}$	$\frac{360\cdot 0.5\cdot\pi}{2\pi\cdot r}=\frac{90\cdot\pi}{r}=\frac{90\cdot 17}{3.5}=437.14^{\circ}$
Inner wheel	0	0 motor stopped

2. Turning around an axis midway between wheel centers, by rotating one wheel in a forward direction and the other in a backward direction, at the same rates (Fig. 6).

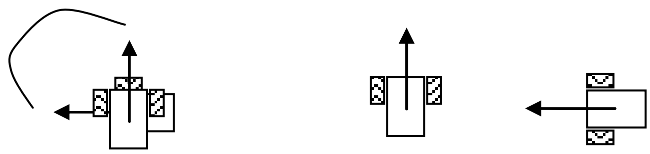


Fig. 6. Visualizing the spatial relationships in pivoting around the robot center

	Distance covered	Rotation angle
Both wheels	$\frac{2\pi R}{4}$	$\frac{360\cdot 0.5\cdot\pi\cdot R}{2\pi\cdot r}=\frac{90\cdot R}{r}=\frac{90\cdot 8.5}{3.5}=218.57^{\circ}$

The instructional sequence starts with the students manipulating the robot wheels, trying to achieve a turning motion. Students suggest the idea of making the wheels rotate in opposite directions, based on videos they have seen of the way tanks achieve turning motion. The students implement their understanding of straight motion and turns by programming the robot to travel along the sides of a square. This implementation leads to an increased awareness of the gap between theory and reality, and the need for the use of sensors to control robot navigation.

C. *Navigating a circular path*

Robot navigation often requires planning a path that avoids collisions with obstacles, thus requiring deviation from a straight line. Due to the robot breadth, the inner and outer wheels traverse arcs of different radii. The difference in radii equals the distance between wheel centers (Fig. 7).

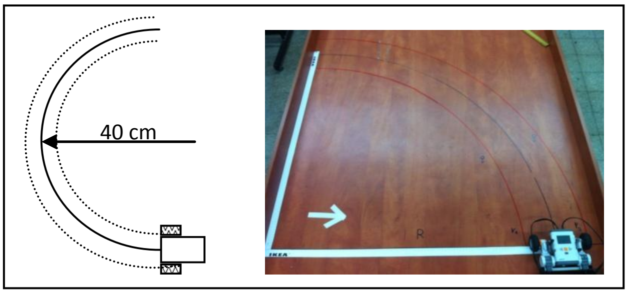


Fig. 7. Visualization of the different paths taken by the wheels and robot center

Assuming the wheels roll without slipping, each full rotation advances the wheel center by the wheel circumference. Since the wheels have equal radii, the inner wheel should complete fewer rotations than the outer wheel, as the robot advances along the curve. Thus, the rotation rates of the wheels need to be coordinated according to the difference in the length of the inner and outer arcs. For example, the distance between a Lego robot's wheel centers is 17.0 cm, and the radius of each wheel is 3.5 cm. For the robot to follow a semi-circular path with a median radius of 40 cm, an outer radius of 48.5 cm and an inner radius of 31.5 cm, the following spatial relationship will need to be established:

	Distance covered	Rotation angle	Motor power ratio
Inner wheel	$\frac{2\pi \cdot 31.5}{2}$	$\frac{360 \cdot \pi \cdot 31.5}{2\pi \cdot 3.5} = 1671.42^o$	$\frac{31.5}{48.5} = 0.6495$
Outer wheel	$\frac{2\pi \cdot 48.5}{2}$	$\frac{360 \cdot \pi \cdot 48.5}{2\pi \cdot 3.5} = 2057.14^o$	

D. Finding the diameter of an outlined circle

Fig. 8 shows the circle circumference outlined by a painted black line. The robot's light sensor recognizes the circle's outline. The robot starts out from a random point on the circle's edge and moves into the circle in a random direction. How should the robot be programmed to obtain the circle diameter?

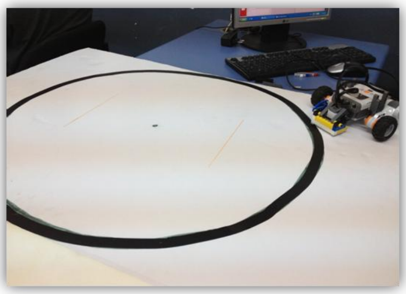


Fig. 8. The robot about to travel into the outlined circle

The suggested solution invokes the spatial relationship between the diameter and inscribed angle (Fig. 9).

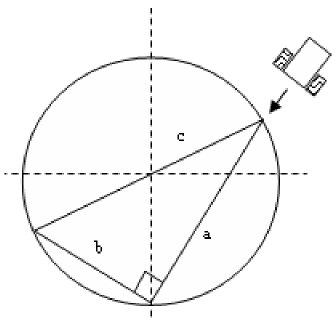


Fig. 9. Visualizing the spatial relationship between the diameter and the inscribed angle resting on it.

1. The robot moves into the circle and measures the distance between the first and second points on the circumference - side a. (Fig. 10).

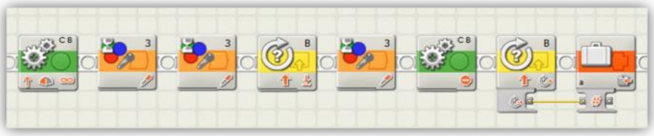


Fig. 10. Programming the robot to move and measure side a.

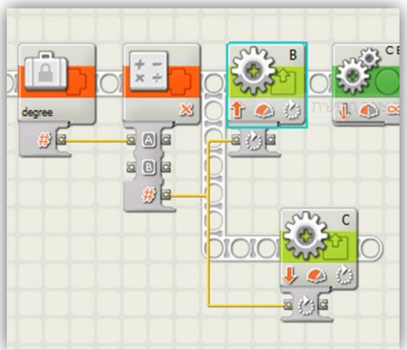


Fig. 11. Programming the robot to make a 90 degree turn around its center.

2. The robot makes a 90 degree turn inside the circle (Fig. 11).

3. The robot reverses out of the circle, then advances until it detects the circle outline. The program shown in Fig. 10 is now repeated and this time it measures the distance of side b.

4. The length of the hypothenuse is calculated using Pythagoras's theorem. Finally, the result is displayed.

The decision to construct a sequence for the "90 degree turn", rather than employ the available dark green block, was intended to promote better of the spatial relationship between circle radius and rate of rotation for different wheels..

VI. SUMMARY

There is wide agreement throughout the science education community that science learners of all ages should be provided with opportunities to experience and practice the art and craft of engineering design. This can best be achieved by project-based learning, which can be carried out within the school curriculum or as extra-curricular elective programs, often hosted by the education or engineering departments of institutions of higher education.

In this paper we have focused on the technique of guiding students by exploring and explicating spatial relationships that can be found within their project space. High school science learners acquired a store of formulas in their science and math school lessons. Many of these formulas describe spatial relationships within systems. Students are accustomed to activate this stored knowledge in the context of end-of-chapter problem solving. We have provided several examples of the ways in which this stored and often "inert knowledge", can be activated by discussions, static and dynamic visualization and structured guidance.

The transition from traditional, teacher centered, text book based instruction to challenging problem solving and technological design is non-trivial, even for high ability science learners. Students must be provided with mental and technical tools and sufficient guidance to help them succeed in the unfamiliar learning environment. Students also need to become aware of the differences that exist between theoretical models and material reality. For example, the focus of a lens is not a mathematical point and the robot's wheels are not perfect circles. This is achieved by comparing experimental results with expected theoretical values, and refining the engineering design to compensate for these effects.

The activities we have described represent a small sample of activities that have been implemented in our work with high school science majors over the past decade. Initial testing indicated the absence of cognitive bridges between theoretical math and physics knowledge vs. material reality. We have collected ample evidence that exploring spatial relationships in real life problem contexts promoted the creation of such bridges. Students' new insights were expressed in the working models they created, solving a variety of seemingly different technological problems by implementing core ideas. For example, the triangulation method for remote distance measurement described in section IV in this paper, has been implemented in systems of transportation (collision avoidance and smart road signs), security (intruder detection and location) and assisting visually impaired persons.

Experience has taught us that achieving the desired transformation of the students' mental world view does not occur spontaneously. It requires an instructional design of revisiting the analysis of spatial relationships in a variety of contexts, using suitable pedagogical tools.

References

[1] Y. Harpaz, Hook, bait and fish: Thinking curriculum approaches (Hebrew). Branco Weiss Institute, 2005.

[2] L. Resnick, Learning in school and out. *Educational Researcher*, 19(9), 13-20, 1987.

[3] A. Zohar, Learning, Thinking and Learning to Think (Hebrew). Branco Weiss Institute, 1997.

[4] Achieve, Inc. on behalf of the twenty-six states and partners that collaborated on the NGSS . *Next Generation Science Standards* (*Engineering Design in the NGSS*). Achieve, Inc., 2013.

Online: http://www.nextgenscience.org/next-generation-science-standards

[5] EPICS - Engineering Projects in Community Service

Online: https://engineering.purdue.edu/EPICS

[6] Project Lead the Way (PLTW)

Online: http://www.materialseducation.org/educators/new/2007/docs/14-Rethwisch.pdf

[7] Engineering Design via Community Service Projects

Online: http://sydney.edu.au/engineering/high-school/index.shtml

[8] H. Altin, "National mechatronics and robotics elective course for upper secondary school level", in D. Obdrˇzálek, (Ed.), *Proceedings of the 3rd International Conference on Robotics in Education*, Prague, Czech Republic, September 13 – 15, 2012.

[9] D. Barlex, Organizing project work. In F. Banks (Ed.), *Teaching technology* (pp. 124–143). London: Routledge, 1994.

[10] M. Barak, & Y. Zadok, Robotics projects and learning concepts in science, technology and problem solving, *International Journal of Technology and Design Education*, 19(3), 289-307, 2009.

[11] C., Bereiter, & M. Scardamalia, "Cognitive coping strategies and the problem of “inert knowledge”", in S. F. Chipman, J. W. Segal, & R. Glaser (Eds.), *Thinking and learning skills: Research and open questions*, (Vol 2, pp. 65–80). Hillsdale, NJ: Lawrence, Earlbaum Associates, 1985.

[12] M. Barak, & Y. Doppelt, Integrating the Cognitive Research Trust (CoRT) programme for creative thinking into a project based technology curriculum. *Research in Science & Technological Education*, 17(2), 139-151, 2009.

[13] D. N. Perkins, & G. Salomon, Are cognitive skills context bound? *Educational Researcher*, 18(1), 16-25,1989.

[14] R.J. Swartz, & D.N., Perkins, Teaching thinking: Issues and approaches, Pacific Grove, CA: Midwest Publications

[15] T., Tseng, & M. Yang, "The role of spatial-visual skills in a project-based engineering design course", in Proceedings of the 118th ASEE Annual Conference & Exposition, June 26-29, Vancouver, BC, Canada, 2011.

[16] M., Welch, D., Barlex, & H. S. Lim, Sketching: Friend or foe to the novice designer? International Journal of Technology and Design Education, 10(2), 125-148, 2000.

[17] S. Olkun Making connections: Improving spatial abilities with engineering drawing activities. International Journal of Mathematics Teaching and Learning, 2003.

Online: http://www.cimt.plymouth.ac.uk/Journal/sinanolkun.pdf

[18] M. C., Linn, & A. C. Peterson, Emergence and characterization of sex differences in spatial ability: A meta-analysis. Child Development, *56*(6), 1479-1498, 1985.

[19] D. Velichová, Geometry in engineering education. European Journal of Engineering Education, 27 (3), 289-296, 2002.

[20] T., Tseng, & M. Yang, The role of spatial-visual skills in a project-based engineering design course. In Proceedings of the 118th ASEE Annual Conference & Exposition, June 26-29, Vancouver, BC, Canada, 2011.

[21] D.F. Lohman, "Spatial abilities as traits, processes, and knowledge", In R.J. Sternberg (Ed.), Advances in the Psychology of Human Intelligence, (Vol. 4, pp.181-248). Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

[22] S., Hsi, M. C., Linn, & J. E. Bell, The role of spatial reasoning in engineering and the design of spatial instruction. Journal of Engineering Education, (86) 2, 151–158, 1997.

[23] K., Ruthven, S., Hennessy, & R. Deaney. Constructions of dynamic geometry: A study of the interpretative flexibility of educational software in classroom practice. Computers & Education, 51 (1), 297–317, 2008.

[24] R. A. Gorska, Spatial Imagination - An Overview of the longitudinal research at Cracow University of Technology. Journal for Geometry and Graphics, 9 (2), 201-208, 2005.

[25] S., Sorby, Educational Research in Developing 3-D Spatial Skills for Engineering Students, International Journal of Science E*ducation*, 31(3), 459 – 480, 2009.

[26] D., Langley, R., Arieli, & B-S,.Elyon, "Enhancing four science learning dimensions: The Physics & Industry experience". Paper presented at the 2010 Girep conference, 23-26 August, Reims, France, 2010.

Appendix

The Physics and Industry (P&I) program is a 15 month, extracurricular, accredited program for 11^{th} & 12^{th} grade physics majors. The students meet on a bi-weekly basis at the Davidson Institute for Science Education, extending their physics knowledge into the field of electro-optics and constructing working models in response to authentic real world technological problems. Project topics include optical surveillance of restricted premises, assisting blind persons, preventing vehicle collisions, color recognition etc.

Detailed descriptions of the P&I instructional framework can be found in previous presentations [25]. For the purpose of the current paper we will focus on ideas for solving the problem of remote distance measurement which activate stored formal knowledge and contribute to an improved understanding of spatial relationships.

Programming in the Real World: Initiation and Motivating Challenges of Entrepreneurship

Daniel López[1] and Martha I. Cárdenas[2]
[1]Robotics Laboratory, Department of Technology, Institut Font del Ferro
Camí de la Ciutadella, SN. E-08389 Palafolls, Barcelona, Spain
iespalafolls@xtec.cat
[2]Department of Languages and Computer Systems (LSI)
Universitat Politècnica de Catalunya
Campus Nord Jordi Girona, 31. E-08034 Barcelona, Spain
martha.ivon.cardenas@upc.edu

***Abstract*—Since nowadays the Enterprise and the High Technology go together hand to hand, trying to reach the control and understanding of real world processes, high school students may recreate it with robotic simulations and enhance their practical training in entrepreneurship. Educational Robotic projects within the field of Technology, Science and Maths, may encourage them to create and develop solutions for many real applications which includes their design, monitoring and control. The project presented in this paper is an educational experience that aims to simulate real processes either industrial or quotidian integrating both Arduino [1] with Scratch for Arduino (S4A) [2] platforms. Further, is given to the student a professional orientation in order to develop and motivate their entrepreneurship abilities.**

***Index Terms*: Educational Robotics, Arduino, S4A, Real World, Entrepreneurship**

I. Introduction and Motivation

Since nowadays the Enterprise and the High Technology go together hand to hand, trying to reach the control and understanding of real world processes, high school students may recreate it with robotic simulations and enhance their practical training in entrepreneurship. Educational Robotic projects within the field of Technology, Science and Maths, may encourage them to create and develop solutions for many real applications which includes their design, monitoring and control. In previous works [7], [11] it has been described that Robotics in an educational context is essential for creating in students ingenious, creative, digital and communicative skills. Further, it is an engine of innovation when causes changes in ideas, attitudes and relationships among students. It also changes the way of thinking and interaction between them, arriving to achieve entrepreneurship skills. This project is an educational experience that aims to simulate real processes either industrial or quotidian integrating both Arduino [1] with Scratch for Arduino (S4A) [2] platforms. The student will learn how to use the Arduino open-source electronics platform which try to reproduce real processes simulated by means of sensors attached to the board. Simultaneously, he will learn how these processes, previously designed at the board, can be simulated also in the computer using the S4A platform. This software let control the Arduino as a sensor input platform. Then, as a result, the students can perform both physical and virtual interactions with each of the designed processes. In this project also, they will get encourage to be entrepreneurs analysing the commercialization of these Arduino-based projects.

Initially, the project begins with exercises developed with little work on Scratch [3] and Arduino which will serve to initiate the student, easily, in programming on a graphical environment by means of blocks. In order to solve the practical activities, various algorithms are proposed. Then, the students solve it doing the automation and control of the device with a computer connected to the USB port model. For this case are used inexpensive and readily available materials. The activities are easily assimilated and designed for develop their own capabilities. In the next stage, the entrepreneurship has two main objectives:

- Assisting students in making decisions about their professional training program in order to adapt their skills to working conditions in a continuous process of change.
- Coordinating and guiding the students of the need for entrepreneurship, as a general approach that can be useful in professional activities, and in everyday life.

From this perspective, the concept of entrepreneurship may include the two aspects: a basic education in entrepreneurial attitudes and experiences that will enhance the autonomy, initiative and self-confidence.

Fig. 1: Robotic process applied to logistics.

Finally, we intended to extend the project to the field of robotics, designing activities where real world robotics systems are simulated and applied to entrepreneurship. The details are described in section VI of this paper [9].

II. Project objectives

The general objectives are the following:

- Creating a learning tool through the simulation of quotidian or industrial processes.
- Developing technological, cognitive and entrepreneurship skills with appropriate levels proposed activities.
- Promoting new technological challenges derived of the project.
- Working transversal between subjects e.g. Mathematics, Technology and Science.
- Developing initiative and entrepreneurial potential.
- Trying to discover better solutions and highlight the importance of creativity in the entrepreneurial process.
- Distinguishing the various stages to launch an entrepreneurial project.
- Understanding the structure of companies and their objectives and then apply them to design a virtual company that will launch to market their products.
- Developing a business plan from the products created in the project.

III. Methodology

This project is aimed at students of the last course from Secondary School (4th ESO degree) but it can be extended to all the Secondary levels up to higher level degrees. During the sessions, the students have performed different Arduino assembly and the associate algorithms designing then in an increasingly complex stage. They use specific sensors to interact successfully with the environment, with the aim of solving specific tasks. All sessions students work in teams of 5-6 students, encouraging teamwork, cooperation and collaborative learning and problem solving. Before the sessions, the software is prepared on laptops, and the students receive all the guidelines of the teacher including the workbook. There are also established the rules for use and care of equipment, as well as individual and team responsibilities. The whole process of student learning process will be recorded in the student workbook designed for this purpose [10]. Additionally, as an individual homework, students are able to enjoy the material in the website of the Institute and design a virtual simulation. The used tools are versatile and simple assuring a student autonomy. Summarizing, they allow a quick understanding of its operation and let develop the pupils creativity.

IV. Project contents

Eight joint sessions with individual and group tasks were performed in the project [7]. These sessions were planned in such a way that all the student had the same material and duration time to work. Each session lasted three hours. Sessions were performed weekly for eight weeks. The different practical activities took place in the facilities of Can Batlle, in Palafolls, Barcelona (See figures 13, 14 and 15). In table I is detailed the program schedule of the project.

TABLE I: Program Schedule

Session	Practical Activities
1	**Introduction to Arduino:** - Project initialization - AVR and Arduino microcontroller: programming - Documentation and Arduino kit delivery
2	**Introduction to Scratch environment:** - First program - Discovering basic structures - Conditionals and iterations - Logic operators, variables and lists - Creation of basic programs
3	**S4A: Real world simulation:** - Introduction to S4A: communication between Scratch and Arduino - Practical examples
4	Project I: Security alarm for a steam generator
5	Project II: Mobile traffic light for public works
6	Project III: Control of a pumping station
7	Project IV: Piano 3.0 simulation
8	Entrepreneurship

A. Practical Sessions

Project I - Security alarm for a steam generator

Problem description of project I: A steam generator is a form of low water-content boiler, similar to a flash steam boiler. An industrial laundry has installed a steam generator and asked us to design the control system pressure of the generator. If the pressure exceeds the safety limit then will connect to an emergency alarm and stop the generator out of service until they make a security review. Figure 2 shows both the virtual environment and the code used by the students to recreate the real process. The code is showed in detail in figures 3 and 4.

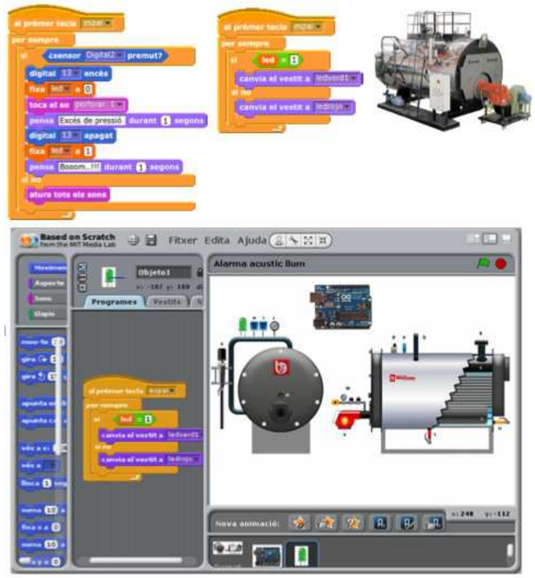


Fig. 2: Project I - Security alarm for a steam generator

S4A Code Description:
In the main program detailed in figure 3, pressing the space key in an infinite loop does the following: if the sensor input 2 is pressed, activates the digital input 13, off the LED, touch the alarm sound and shows the message on the screen overpressure. Then, off the digital input 13 and led lights. On the other hand, if digital is not pressed will stop the whole process.

Fig. 3: Project I - Main code.

This auxiliary code detailed in figure 4, corresponds to an object code applied to the LED object which has two costumes: the green color when the led lights and the red color when the led is off.

Fig. 4: Project I - Auxiliary code.

Project II - Mobile traffic light for public works

Problem description of project II: The local city Council has provided a mobile traffic light to road users and residents affected by public works in an intersection with an awesome traffic jam. Although signalized intersections are in place for increased safety, the use of this mobile traffic light let the efficient traffic and avoid accidents. Then, the students has to analyse the problem, design the objects and create the code (see figures 5 and 6) .They also have to deal, as in real world, with the adjustment of the signal delay, adapting the code to current traffic patterns. As a result, is assured an improved traffic flow getting reduced intersection delays, lower air pollution/vehicular emissions, and reduced gasoline consumption.

Fig. 5: Project II - Mobile traffic light for public works: S4A environment.

Fig. 6: Project II - Mobile traffic light for public works: S4A simulation code.

S4A Code Description:
As detailed in figure 6, if the space key is pressed then, in a forever loop, is activated the digital input 13 while digital inputs 10 and 11 are not. Then, is visualized the green light

for 5 seconds. Then, the flip-flop effect of the yellow light is created activating 4 times the digital input 11. Finally, is activated the digital input 10 during 5 seconds while the other input 13 and 11 are not.

Project III - Control of a pumping station

Problem description of project III: We were commissioned to control the correct functioning of an automatic system which activates a synchronized group of water pumps in an industry dedicated to textile dyeing.

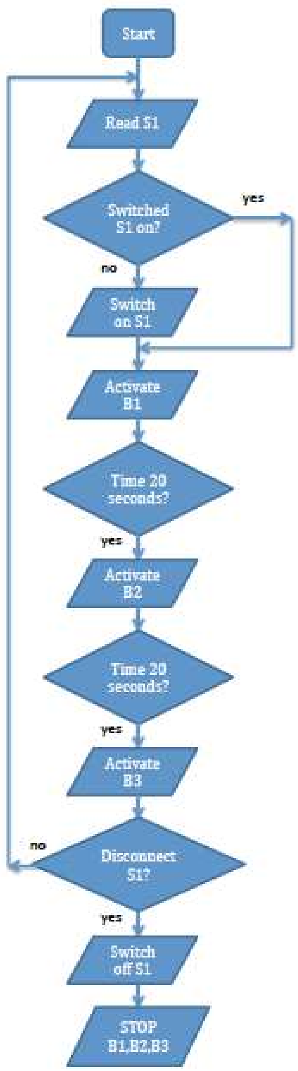


Fig. 7: Project III - Flowchart of the whole process.

We'll have to control the following premises; The switch S1 starts the pump B1, within 20 seconds, the pump B2 is connected, within 40 seconds pump B3 will run. If the switch disconnects the pumps stopped. The operation of the pumps are indicated with LEDs. A flowchart of the whole process is visualized in figure 7, the S4A environment is detailed in figure 8 and the S4A code is detailed in figure 9.

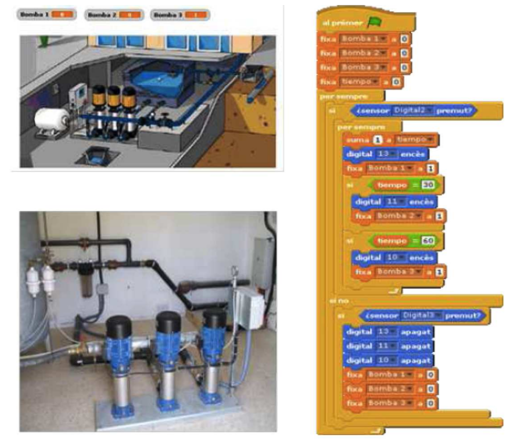


Fig. 8: Project III - Control of a pumping station: S4A environment and code detailed below.

Fig. 9: Project III - Control of a pumping station: S4A code.

In order to give more realism to the process, students added three different sounds each one representing a pump. This simulation combinated with the control variables let to test effectively the simulation process and adjust the code parameters.

Project IV - Piano 3.0 simulation code

This project is an example of an investigation research done by a high school student. It aims to simulate a virtual piano with S4A. The related application consists of controlling a PC keyboard with a multiple music selection for playing the desired music as in a real piano. Students worked in teams in order to create different sounds for users interested in the product. Figures 10) and 11) show the codes created by the students. Firstly, students draw a diagram and designed the piano model in Scratch as can be observed in figure 12. Basically, they assigned an upper case letter from keyboard to a key note. Then, they designed the software which had two object codes: one for the white keys of the piano and the other for the semitone keys of the piano.

Fig. 10: Project IV - Code of the white keys of the piano

Fig. 11: Project IV - Code of the black semitone keys of the piano.

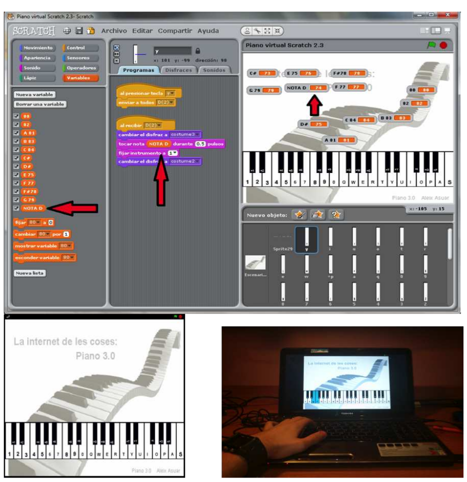


Fig. 12: Project IV - Piano 3.0 simulation environment.

In general, during the project, students had to deal with some Scratch limitations like the number of the assigned key notes. They solved it adding new variables and using the space key together with the letter keys. They created in total 12 additional variables.

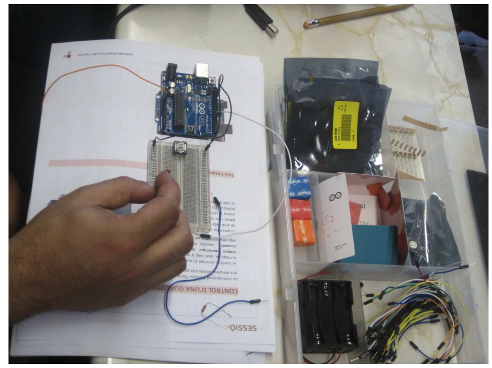


Fig. 13: In this practical session students use the micro-controller Arduino material and the workbook.

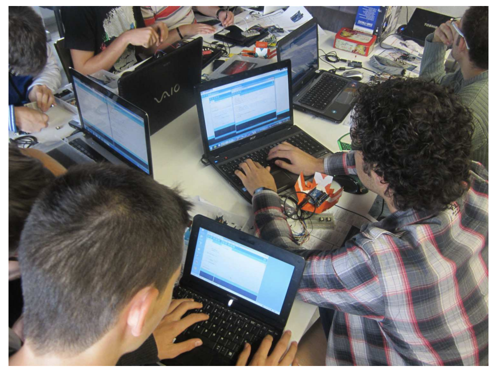


Fig. 14: In this practical session students are working in team

Fig. 15: In this practical session students work under the professor guidance

Entrepreneurship

Entrepreneurship involves the development of a range of capacities from the initiative and decision to creativity and reflection [8]. In this sense, the student contributes to the achievement of basic skills, such as autonomy and initiative and competence of learning to learn, Also, they work capabilities that enhance their leadership, taking responsibility, learning from mistakes and take risks, and being able to continue to learn more effectively and independently. In his development they can undertake personal projects that can be implemented in reality, realizing their own itinerary academic and professional.

The students will have the role of businessmen having a meeting with the members of those who have chosen the same virtual enterprise. So one students should represent the group, and another, will be a secretary. To ensure the interaction of the group, the members will alternate roles.

Some brainstorming: Derived from the practical activities, students recreate a real problem like this: *Your group is a specialist team currently working in electronics for a Corporation. Before making the project feasible, the team has received an order in which asked design, simulate and control a security alarm for a boiler. The designed system has to effectively detect any abnormalities. It has to be ready in 2 weeks, both the prototype and the virtual simulation. Would you be able to do?*

V. Evaluation and Resources

Evaluation is a key element in the educational practice, allowing in each time, collect information about learning process of the student [5], [15]. It is also necessary to make value judgements helping to the guidance and reinforcing this learning process [16]. With this purpose, the students were evaluated during the sessions. In this project, a student workbook was included as an item helping the instructor in the evaluation phase. Firstly, it was applied the direct observation of the student in order to assess his attitude and daily work (attention in class, assistance, answers to questions and participation in the classroom). Secondly, the professor used an evaluation model sheet [11], [12]. This evaluation model explained below consists of a grid of the items with a series of options that would be marked by the professor (See figure 16):

- Planning and Design Criteria.
  1) Creativity and Design.
  2) Use of resources (tools, materials and sources of information).
  3) Digital competence.
- Organization Criteria.
  1) Personal autonomy.
  2) Organization of assembling a logical structure.
- Technical Criteria.
  1) Knowledge of control elements (main parts).
  2) Knowledge of mechanical elements (auxiliary parts).
- Interaction Criteria.
  1) Oral communication.
  2) Written communication (drawings, sketches, writing).
  3) Teamwork.

1 — **Evaluation Criteria** — Student name:

ROBOTICS — ELEMENTARY- PRIMARY - SECONDARY

Evaluation Criteria	5 A	4 B	3 C	2 D	1 E	FINAL MARK	Professor comments
Planning and Design Criteria							
1. Creativity and Design.	☐	☐	☐	☐	☐		
2. Use of resources (tools, materials ...)	☐	☐	☐	☐	☐		
3. Digital competence.	☐	☐	☐	☐	☐		
Organzation Criteria							
4. Personal autonomy.	☐	☐	☐	☐	☐		
5. Organization of assembling a logical structure.	☐	☐	☐	☐	☐		
Technical Criteria							
6. Knowledge of control elements.	☐	☐	☐	☐	☐		
7. Knowledge of mechanical elements.	☐	☐	☐	☐	☐		
Interaction Criteria							
8. Oral communication.	☐	☐	☐	☐	☐		
9. Written communication (drawings, writing).	☐	☐	☐	☐	☐		
10. Teamwork.	☐	☐	☐	☐	☐		
Total score:							☐ 0-10 E ☐ 11-20 D ☐ 21-30 C ☐ 31-40 B ☐ 41-50 A

Fig. 16: Model of evaluation sheet used during the project.

Moreover, a questionnaire about the project was given to the students intended to get the feedback and to capture its educational impact. The obtained results were used in determining the project success and to support future related projects to launch. The questionnaire had 5 categories that the students were asked to rate evaluating it in *very interesting*, *interesting* and *not interesting* for each topic. As a result, as can be seen in figure 17, just the 11% answered not interesting, compared to the 73% very interesting and the 16% interesting. The categories were: Scratch programming, S4A programming, Arduino programming, Arduino hardware and Entrepreneurship.

Fig. 17: Questionnaire about the whole project. The blue color corresponds to label 1 which represents the 11% who answered *not interesting*, the red color corresponds to label 2 which represents the 73% who answered *very interesting* and the green color corresponds to label 3 which represents the 16% who answered *interesting*.

VI. Further Project Extension: Robotics Applied to Entrepreneurship

Following on from the above project we aim to extend it to real areas in which robotics is involved, all in an interesting and enjoyable way. Then, we proposed two real applications to simulate using LEGO Midnstorms [4], [14] both explained in detail below:

- Practical R1: Automated line for filling, packing and palletizing bottles of wine.
- Practical R2: Intelligent system for logistic operations by autonomous industrial robots path followers.

A. Practical R1: Automated line for filling, packing and palletizing bottles of wine

Description of the problem: A company of Valladolid has developed an automated system for the wine making industry. It's a line for filling, packing and palletizing bottles of wine whose benefits cost savings and reduced time invested in these processes. The system is the result of a study of mechanical engineering, robotics and industrialization. Figure 18 shows the real process and a virtual environment of the process.

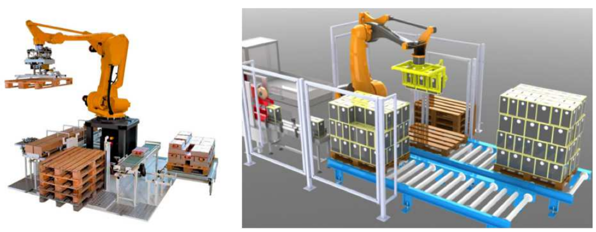


Fig. 18: Real process and virtual environment of R1.

Also, is done a study of the materials and technology needed to carry out the whole project. The process also includes a schedule of the robot and other equipment. Is needed the use of advanced sensors to detect when the bottle is filled and a camera vision to verify that the work of the robot is correct.

In order to adapt it to our practical, a LEGO Mindstorms kit model has to be designed by the students and they have to program it in NXT-G software. Figure 19 shows the LEGO construction and the NXT-G code for the simulation.

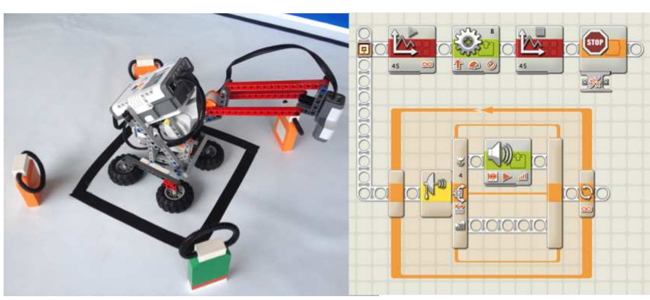


Fig. 19: LEGO construction and NXT-G code of R1.

Figure 20 shows the diagram of the robotics process: the red circle shows the rotation line of the ultrasonic sensor, the orange circle shows the robot situation and the blue circles simulates the place where the pack is picked or not, depending on whether it is empty or not.

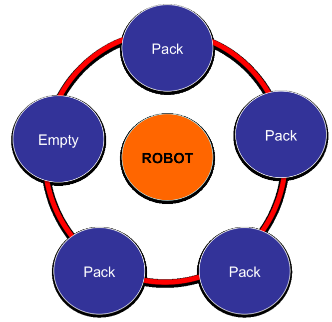


Fig. 20: Diagram of the robotics process of R1.

B. Practical R2: Intelligent system for logistic operations by autonomous industrial robots path followers.

Description of the problem: An enterprise specialized in logistics and transportation need a robotic system guided and monitored in an autonomous way. Images of the real process and the whole industrial coordination are summarized in figures 2, 21 and 22. Then, is proposed the design of an intelligent system which uses autonomous industrial robots path followers. With this purpose, students built the robots and design the code following the NXT-G code instructions provided by the software, both detailed in figure 23.

Fig. 21: Real process image of practical R2.

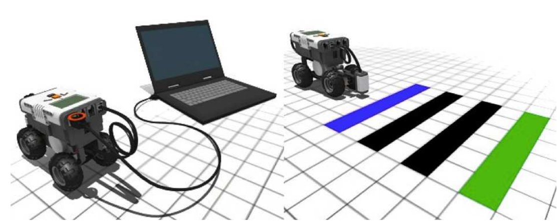


Fig. 22: LEGO image and monitoring of practical R2.

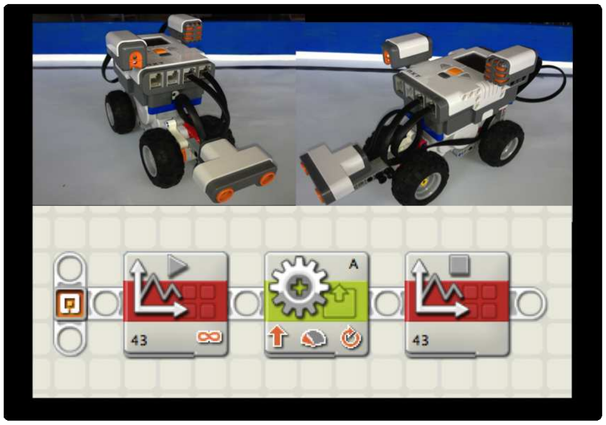


Fig. 23: LEGO kit and NXT-G code of practical R2.

VII. Conclusion

In this paper we have described succinctly and briefly the effort done by the Institute Font del Ferro. The acceptance by the educational community has been very positive, proof is the broad participation in the various events that the institution has organized. The union between real industrial processes and the virtual world has allowed us to work and give a entrepreneurship orientation. Moreover, the academic and professional guidance to our students let them see new possibilities in the field of young entrepreneurs. Science and technology are powerful tools for all the students who joined this project. The evaluation of the project, generally, was positive in all its aspects, because students worked creatively and without pressures, pleased to learn to learn, participate and build.

Acknowledgment

We would like to thanks to the Educational Community of the Institut Font del Ferro, the Association of the Families (AMPA) and the Palafolls Council.

References

[1] ARDUINO (official site) http://www.arduino.cc/
[2] Scratch for Arduino (S4A) (official site) http://seaside.citilab.eu//
[3] SCRATCH (official site) http://www.scratch.mit.edu/
[4] LEGO.com Mindstorms (official site) http://www.mindstorms.com.
[5] Barbier, J. M. (1993): La evaluación en los procesos de formación. *Barcelona: Paidós.*
[6] Baum, D.(2000) Dave Baum's Definitive Guide to LEGO Mindstorms. *Apress, Emeryville, CA.*
[7] Cárdenas, M.I. and López, D.(2012) IV Jornada Catalana de l'Ensenyament de la Programació a Primària i a l'ESO. *CESIRE-Aulatec.* Departament d'Ensenyament, Generalitat de Catalunya.
[8] Drucker, P. F. (1986) Innovation and entrepreneurship; practice and principles. *The Journal of Continuing Higher Education. Volume 34, Issue 1.*
[9] Jones, J.L., Seiger, B.A. and Flynn, A.M. (1993) Mobile Robots: inspiration to implementation. *Wellesley, M.A. AK Peters LtD.*
[10] Kafai, Y. and Resnick, M. (1996) Constructionism in Practice: designing, thinking and learning in a digital world. *Mahwah, NJ: Lawrence Erlbaum.*
[11] López, D. and Cárdenas, M.I. (2012) Robòtica Educativa: Una metodologia motivadora, integradora, dinàmica i creativa per a la transició en el sistema educatiu. *II Jornades TIC i Educació. Institut de Ciències de lEducació Josep Pallach de la Universitat de Girona.* ISBN 978-84-8458-411-7 (p.85-88)
[12] López, D. and Cárdenas, M.I. (2012) Formació Grup de Treball de Robòtica Educativa. Centre de Recursos Pedagògics (CRP) del Maresme II. Departament d'Ensenyament, Generalitat de Catalunya.
[13] Papert, S. (1986) Constructionism: A new opportunity for elementary science education. Proposal to the National Science Foundation. MIT Media Laboratory.
[14] Resnick, M. (1989) LEGO, Logo and Life. *In Langton C. (eds.) Artificial Life. Boston, MA: Addison-Wesley.*
[15] Stufflebeam, D. L. and Shinkfield, A. J. (1987): Evaluación sistemática. Guía teórica y práctica. *Madrid: Paidós-MEC.*
[16] Tejada, J. (1999) La evaluación: su conceptualización. Jiménez, B. (ed.): Evaluación de programas, centros y profesores . *Madrid: Síntesis.*

Elements of Scrum in a Students Robotics Project - A Case Study

Reinhard Gerndt
Ostfalia University of Applied Sciences
Wolfenbuettel, Germany
Email: r.gerndt@ostfalia.de

Ina Schiering
Ostfalia University of Applied Sciences
Wolfenbuettel, Germany
Email: i.schiering@ostfalia.de

Jens Lüssem
University of Applied Sciences Kiel
Kiel, Germany
Email: jens.luessem@fh-kiel.de

***Abstract*—Robotics competitions allow self-organised learning in a quite natural way. In the last years, we have observed an increasing complexity in these competitions. At the same time, the search for an adequate project organisation became more and more important. As traditional project organisation methods failed, we adopted Scrum and tried to adapt this agile methodology to student projects.**

***Keywords*—Student projects, robotics competitions, project management, self-organisation, agile methods, Scrum, project-based learning**

I. Introduction

Student projects are an integral part within our robotics teaching activities. In our teaching philosophy [1], we combine traditional learning approaches with project-based learning [2]. Competitions offer an interesting environment for student projects [3].

Furthermore, robotic competitions provide an excellent motivation for students to study in a self-organised manner, which opens widely the path for natural curiosity. Competitions provide clear functional objectives and measures of success. Consequently, requirements (in form of rules for the respective competition) and deadlines (e.g. the day when the competition takes place) are not questioned. Furthermore, competitions offer a means of assessment outside of the university grading system.

A couple of years ago, when our student groups started with participating in competitions, they were nearly self-organised. Lecturers were in the role of experts in robotics.

Since then, we have seen an increasing number of competitions with more and more sophisticated technical and scientific objectives. Reaching a sufficient quality level and good rankings in the competitions with student groups became an increasingly challenging undertaking.

Instead of then taking the role of a project manager and lead the student group, the authors chose a different approach. It was perceived that the self-organised student groups were such a success story that self-organisation should not be given up too quickly. Especially young students benefit enormously from these experiences - such as building teams, or managing changes.

In the past years, it thus became more and more crucial to find a self-organising project management approach that preserves the motivational aspects and leads to at least satisfying results in the competitions.

To address these challenges, we investigated to which extent agile methods like Scrum can be used for the management of student projects.

The remainder of the paper is organized as follows. In Section 2, we introduce the robotics competitions, our student teams have participated in. Section 3 describes the project management methodologies we applied. Sections 4 and 5 focus on agile methods and their applicability in student projects. In Section 6 we report our first experiences in using Scrum for student robotics projects. Finally, we summarise the main findings and describe future work in Section 7.

II. Robotics competitions and their complexity

Robotics competitions differ in many ways. There have been and still are competitions related to robotic cars, aerial vehicles, military robotics, just to mention some. One of the most prominent robotics competitions is the RoboCup [4]. It is based on the challenge to play a game of soccer with a humanoid robot team against the human world soccer champions in the year 2050. Many of the aspects of this paramount objective are targeted in individual leagues, some of those further subdivided into sub-leagues and partial competitions. To foster exchange with other robotic fields, some peripheral leagues and competitions, which are not immediately related to robotic soccer have also been introduced to the RoboCup. Aside of the specific functional objectives, different aspects of complexity [5] relate to the partial competitions. As the most obvious aspects, in this section, we present the targeted competitions with respect to the robotic complexity and the task complexity. The robotic complexity covers hardware and immediate, i.e. low-level control or kinematic complexity. The task complexity describes the complexity of the functionality a robot may have to implement for a competition.

A. Mixed-Reality competition

Initially, the student group joined the RoboCup Mixed-Reality competition [6]. The challenge is a robotic soccer game with up to 11 small wheeled robots per robot team, playing with a simulated, virtual ball. The mixture of real robots and virtual environment and ball led to the name. The main task is implementing a cooperative, possibly swarm-like, behaviour of a group of robots. Small, differential drive cubic robots with a volume of approximately 8 cubic centimetres are used as players. The playing field is a horizontally mounted screen to display the virtual field and the ball. The robots are controlled via an optical link by software agents, running on a standard PC. Process information, like the robot position is made available to the agents by means of additional software packages.

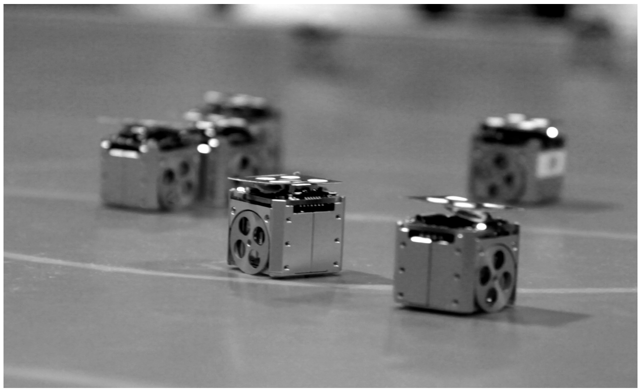


Fig. 1. Standard mixed reality robots used in RoboCup competition

The basic system was developed beforehand by a specialized development team and made available for the implementation of robotic soccer functionality by the student groups. Differential drive robots have a straight-forward kinematic model, such that robotic complexity is comparably low. The task for the student group is to implement the cooperative behaviour of a group of robots to play a game of robotic soccer in software. The limited complexity of the agents and loose coupling of system components allows for individual students implementing the entire functionality or behaviour of a robot. Thus the student group is facing a relatively low complexity at the robot and the task level.

B. RoboCup kid size humanoid competition

Following the initial successes, the significantly more complex RoboCup kid size humanoid competition [7] has been addressed as next major step. The challenge currently is a three vs. three robotic soccer game. Initially, the main task is realising robots that are capable of kicking a ball to a goal in a more or less sophisticated way. The size of the humanoid robots with 18 or more drives is in the range of 30 - 60 cm. The field is six times four meters. A tennis ball is used to play the game. Goals, the ball and the robot teams are colour coded. The robot is controlled by one or more on-board computers and carries all its human-like sensors.

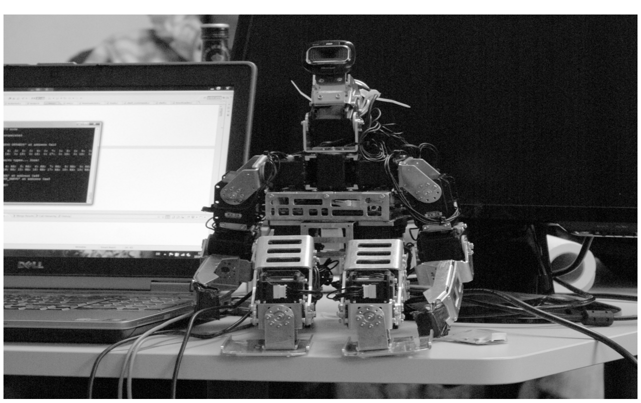


Fig. 2. Current humanoid robot, adapted from open source robot plans

Designing and building the robots now is part of the competition. In addition, low-level control and kinematics became significantly more complex. Furthermore, acquiring information on the environment in sight of unreliable data from the sensors and wear of hardware adds to the robotic complexity. The task complexity is basically comparably to the complexity in the Mixed-Reality competition. However, introducing a real ball slightly added to the task complexity. As a general property, the humanoid robots require a closer co-operation at software and hardware level and at the hardware-software interface and thus required closer cooperation between members of the development team.

C. RoboCup@work competition

As a currently final step, the RoboCup@work competition [8] related to an industrial workshop situation with the Youbot, a miniature version of a real mobile industrial robot, has been addressed. The challenges within the competition include navigating in a workshop situation and manipulating and transporting work pieces. The robot consists of a 60 x 40 cm omnidirectional mobile base with very user-friendly kinematics and a 60 cm industrial robot arm with five degrees of freedom. It has been delivered operational with a Robot Operating System (ROS) basic software [9].

Fig. 3. Kuka YouBot robot without team-specific enhancements

Thus a basic operational software and hardware platform was available from the very beginning, like in the Mixed Reality competition. However, all sensors, like cameras and laser range scanner had to be selected, integrated and maintained by the student group. Typically software libraries were available to access the sensors. Thus the robotic complexity was lower than in the humanoid competition, but significantly higher than in the Mixed Reality challenge. Functional requirements include localizing, identifying and grabbing different work pieces and carrying out different tasks, thus resulting a higher task complexity. With a larger community, working with the Youbot, reliability issues are less dominant. However, the large set of (sub-) functions that make up a robot task requires close cooperation of the student group at software level.

The robot and task complexities of the three competitions we participated in are summarized in the following figure.

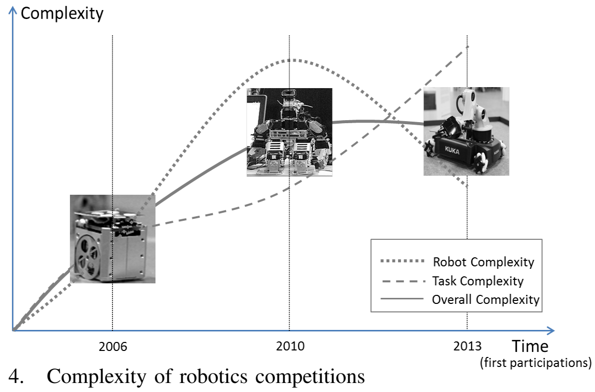


Fig. 4. Complexity of robotics competitions

III. Philosophy and the Management of Complexity

A. Teaching philosophy: roles and requirements

In robotics education we combine “traditional” learning methods with learning-by-teaching and problem-based learning approaches. For us, problem-based learning is not only an additional teaching method, it’s rather the most important brick in our teaching strategy [3] which follows the European Qualification Framework.

Robotics competitions offer a wide range of problems student groups can tackle. In our teaching philosophy, students must have the chance to solve this kind of problems (practically) on their own. Therefore, we avoid an involvement in the day-to-day project work. Consequently, we act as experts and are often in the role of an advisor or mentor. So, self-organisation is our main requirement for the student group.

Further sources for requirements are the students themselves. Our more technical oriented student team aims to focus on robotics (i.e. hardware and software development).

Competition organisers are a last important source of requirements. The organisers set the rules for the competition including constraints on the robots and the underlying infrastructure.

The most important requirements are shown in figure 5.

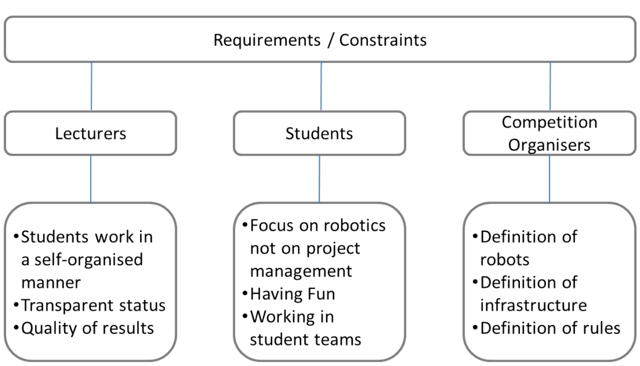


Fig. 5. Requirements and constraints

B. Managing complexity: project organisation

During the last seven years our student team participated in three different competitions (see figure 4). At the beginning we did not pay too much attention on project organisation or project management. This attitude changed drastically as we started in the Kid Size Humanoid League Competition.

1) Mixed-Reality Competition: Initially, a self-organising merely unstructured approach was chosen. Every member of the group felt responsible for the overall outcome. The specific implementation that was to be used for an official competition event was chosen by internal competition. Different competencies among group members had some influence on the selection process, but rarely influenced the decision during this phase of project management. If intermediate results, especially during an official competition, indicated the necessity for changes in the software, specific branches of the implementation have been developed from the selected version and selected by immediate comparison. In some rare cases, the group switched to one of the previously discarded implementation and carried on with it after unanimous decision.

Eventually, a hierarchical project management structure evolved. Students had to take over organisational duties, like interacting with the competition organisers and organising travel and accommodation and thus turned into acting management students. They, however, often could not cope with the high, also emotional stress during competition events sooner or later and resigned. It is worth to mention, that members with high technical and scientific competencies always concentrated on the technical and scientific work and did not take over management duties.

In the technical domain, clearly bounded responsibilities evolved and all individual tasks were covered by individual members of the team. The boundaries, however, due to the considerably low complexity and loose coupling evolved naturally and required no specific agreements among group members. Definition of interfaces was obvious and none to very little project management activities were required.

2) Kid Size Humanoid League Competition: Driven by the success, subsequently activities in the humanoid sector have been started. Initially the same project management and iterative individualized design procedures have been used. Members of the group concentrated on their specific segment. They defined individual measures of success and evaluated the development outcome against their own partial test cases. Often, the official competitions were the only integration tests. However, with a high degree of independence students individually prefer to add new features instead of working for quality. By constantly mixing debugging and developing new features, they jointly never reach a release that could be used as a fall-back position.

From the accomplishments it became obvious that, with the significantly more complex functionality and higher interdependencies among system components, now a more extensive project management was a necessity. Furthermore, the considerably small team relied on finding synergies to handle the overall complexity of the robotic system.

3) RoboCup@Work competition: The activities related to the industrial robot started from a similar point. However, with a clearly defined hardware and software architecture and some basic software functionality available, the well-known iterative approach could be followed for some time. However, eventually, by improving existing hardware and software components and adding new functionality, the overall complexity rose in such a way that project management now became necessary. As a consequence an agile approach was proposed to the student group.

C. Student feedback

In order to confirm our impression, we carried out a survey among students with at least 6 months of involvement in the robotic work group. In one section of our questionnaire, we asked students for their priorities in the robotic projects. The results showed a clear priority for a self-organised approach over guidance by a lecturer. Priorities for individual work and teamwork were almost leveled, with a small bias towards teamwork. Spending time for meetings or individual work was leveled, like having fun versus achievements in competitions. As an interesting result students claimed to prioritise quality over adding new features, which was not fully in line with the impression of the authors, while guiding the team.

Another part of our questionnaire was dedicated to project management methods. According to the feedback, all students were quite familiar with the waterfall model. The waterfall model is an early model for software engineering where the phases requirements, design, implementation, test and maintenance are performed sequentially. V-model and the iterative approaches like the spiral model were known less and the spread of familiarity was larger. The V-model is a an enhancement of the waterfall model. In the V-model, the development of tests for the test phase is already started after the requirements are specified. The spiral model was developed based on the experiences made with the waterfall model. The central idea is that it is challenging to develop complex projects in one step. Hence the project is divided into so called interations. In each iteration a development cycle as in the sequential models is performed. Hence experiences with first prototypes and changes can be included into the next iteration.

Agile methods, e.g. Scrum were known even less with a considerable large spread of familarity, even after some exposure during the project. However, voting on the expected suitability of the respective methods showed a clear preference for agile and a little less for iterative models. V- and waterfall models were considered not suitable. In general the findings correlate with the project management approaches that have been used so far and currently are used (fig. 6).

Agile approaches and their usage in our student group will be discussed in detail in the following sections.

IV. Agile Methodologies

The investigation of agile methodologies was started as a reaction to experiences made with the waterfall model in the 1990s based on ideas of lean production. At around the same time the imperative programming paradigm was accompanied and in parts replaced by object oriented programming. Also the phase of the so-called new economy started which led to shorter time-to-market and therefore also to shorter product life cycles and frequent changes of requirements during projects. These requirements were difficult to realise with existing methodologies like the waterfall model, but also with the iterative methodologies.

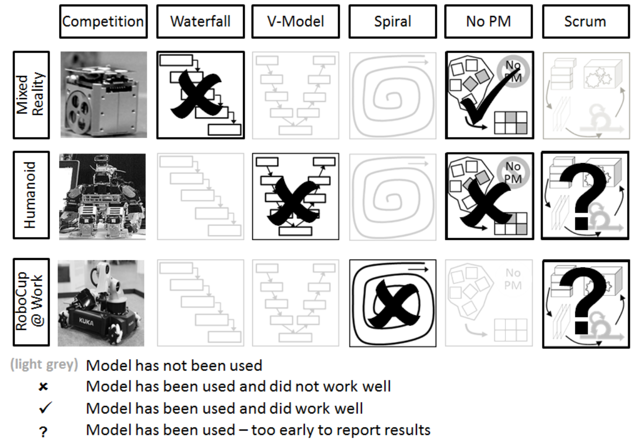


Fig. 6. Project management approaches in the students' team

The aim of agile processes developed in this period was to create an environment that facilitated the collaboration of people and tried to enhance the productivity of developers. The focus is on creating value for users and allow for recent changes of user requirements. The most known agile methodologies developed by this time were XP [10] (eXtreme Programming) and Scrum [11]. For an overview of agile methodologies and a review of studies about adoption and experiences of these methodologies see [12]. XP and Scrum are both often used in teaching environments, which are specialised courses concerning XP [13], general software engineering courses [14] or the management of student projects integrated in courses [15], [16]. Since these case studies were very positive in general, we tried to adapt agile methodologies to self-organised student groups.

The agile manifesto describes the values incorporated in agile methodologies [17]. The following values are the basis for agile methodologies and processes:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

In the following we give a short introduction to XP and Scrum which are the best known examples of agile methodologies.

A. XP

eXtreme Programming (XP) was developed by Kent Beck, Ward Cunningham et al. [10]. The focus of XP is on the close communication with users, extremely short development cycles which are called iterations. They are typically 1 to 2 weeks long. During this iteration the typical phases of software engineering (analysis, design, coding and testing) are performed. We present XP here by stating the XP practices according to Beck [10] which are often used also outside of XP:

- *The Planning Game* During the planning game the next release and the tasks of the iteration are planned.

- *Small releases* Small releases are realised that are put into production quickly. It is possible to have e.g. daily releases or weakly releases.
- *Metaphor* The system as a whole and all components should have names that are easy to remember and relate to the use of the component.
- *Simple design* The architecture of the software should be as simple as possible. There are no preparations for future features. Additional complexity that is no longer needed is reduced by Refactoring.
- *Testing* Programmers write unit tests to test their code continuously during development. These tests must run flawlessly to demonstrate that a feature is finished.
- *Refactoring* Programmers restructure the code and adapt the internal structure of the system as often as appropriate without changing its external behaviour. This can be assured by testing. The combination of Testing and Refactoring is called Test Driven Development.
- *Pair programming* Two programmers work together. One programmer is writing code while the other one reviews it, thinks about improvements and is able to give immediate feedback. The roles are switched regularly.
- *Collective ownership* The code is collectively owned by the team. Every programmer has access to the whole code and is able to change code everywhere.
- *Continuous integration* The software is build and integrated regularly. This could be done several times a day or at least always when a developer completes a task.
- *40 hour week* To work overtime should be avoided as a general rule, because the quality of the code would not be appropriate when written in a stress situation.
- *On-site customer* It is expected that real users are available the whole day for the team to answer questions.
- *Coding standards* Write Code according to agreed Coding Standards. The resulting code should be easy to read.

B. Scrum

Scrum was developed at the same time as XP (amongst others) by Jeff Sutherland and Ken Schwaber. This software development framework consists of the following elements (see [18]): Scrum is based on short releases of about 2-4 weeks which are called *Sprint*. To organise the development in these short releases the following elements are defined:

- **Roles:** Product Owner, Scrum Master, Team
- **Ceremonies:** Sprint Planning, Sprint Review, Daily Scum Meeting
- **Artifacts:** Product Backlog, Sprint Backlog, and Burndown Chart

The *Product Owner* is the representative of the users. The responsibility of this role is to define the features to be realised. These features are described and prioritized in the *Product Backlog*. This is often realised in the form of User Stories. Beside the prioritization the complexity of the user stories is estimated. This is often realised by estimating not the effort of user stories but the relative complexity. To avoid the influence of other participants often a so called planning poker is used. At the end of the Sprint the Product Owner has to check and accept the results of the Sprint.

The *Scrum Master* has to ensure that the collaboration inside the team leads to cooperative work and that the Scrum process is followed.

The team organises the development work and defines the *Sprint Goal*. This is based on features of the Product Backlog. The selected features are transferred to the *Sprint Backlog*. The team plans the development work in the *Sprint Planning* at the beginning of the Sprint. There, concepts concerning the architecture and ideas for tests are detailed. Also open issues can be clarified with the Product Owner.

Based on this plan the team is responsible to finish the Sprint Goal at the end of the Sprint and to present the results to the Product Owner. The progress of the team is denoted in a *Burndown Chart* where finished user stories are marked. A user story is finished, if it is ”done”. This notion of done incorporates thorough testing of the solution. Every day there is a short (stand-up) meeting, the so called *Daily Scrum* to discuss questions, talk about the progress and organise the work of the day. In Scrum the team chooses appropriate methods for the development work. There are no regulations concerning the organisation of software development from the Scrum methodology. Often some of the practices of XP are used, but these practices are selected by the team and are not mandatory.

At the end of the Sprint there is a review of the results and the process in the *Sprint Review*. Afterwards the next Sprint starts again with the Sprint Planning.

V. Elements of Agile Methodologies Proposed to Student Group

The specific situation in a mostly self-organised student project imposes a number of constraints on the selection of the project management elements. However, the necessity of a project management to handle the technical complexity and organise the work sets limits to the constraints. As a result a not fully homogeneous set of elements of agile methodologies has been identified as a base for project management in such projects.

From the experience of project management in student projects a very critical point was the difficulty in estimating efforts. The reasons are the lack of experience and that the projects are very innovative and hence the technical challenges are not clear at the beginning. Furthermore, hardware failures and the resulting procurement of replacement parts often introduce delays. Therefore, project plans were typically too ambitious and the students left work packages semi-finished, because of the pressure of the plan.

Another important area are the special aspects of working with students: The work must be fun, because the students

Fig. 7. Scrum Meeting of Robotics Team

are working voluntarily. Also they are pursuing their studies and are attending lectures and doing study projects as a main priority. Hence the time they got for the robotics activities is difficult to plan and interfaces between work packages are critical. Therefore team building and flexible planning are important aspects.

An important point is the communication not only inside the single activities but the transparency and sharing of knowledge between the activities. The idea here is to try to build upon common experience e.g. concerning architecture like the Robot Operation System (ROS), blackboard architecture and artificial intelligence.

Therefore a project management methodology for self-organised teams of students in robotics projects should address the following aspects:

- Easy estimation of work packages
- Innovation oriented flexible project planning
- Quality checks and testing
- Team building
- Transparency

In general the values of the agile manifesto stated above are well suited to address the special challenges of student projects as stated in the description of the requirements: The individual and the team are in the focus of the work, recent changes to the plan are accepted and working software is the aim. Also quality management is incorporated and the agile methodologies are inherently transparent. Therefore we proposed to the student group to use an agile methodology for project management as stated above.

For the decision concerning the agile methodology to use we considered the following aspects: XP is very much concentrated on organising the software development itself by prescribing engineering practices, whereas Scrum allows the team to choose the practices for development themselves. Also XP assumes that a user is always available (On-Site Customer) whereas Scrum accepts that a representative is available at least during Sprint Planning and Review. Since some of the XP practices as Pair Programming, On-Site Customer and the strict use of unit tests for Testing are difficult to realise in a robotics project, we decided to use a variant of Scrum adjusted to the special requirements of self-organised student projects.

We started with Sprints of 2 weeks length and user stories to formulate a Product Backlog. With this form of a rough planning in general and a detailed plan made just at the beginning of the sprint it was easier for the students to estimate efforts and plan communication and interfaces in the project. The project planning in the form of Sprint Planning turned out to be more realistic because the students had to work on their User Stories (work packages) until they were finished and had to explain progress and difficulties to their fellow group members where fruitful discussions followed. But we observed that until now often User Stories were not finished in one Sprint which is an issue.

Daily Scrum Meetings are not possible, because the students do not work on a daily basis on their projects. The compromise was the relatively short length of the Sprints. We used at least weekly meetings for the last weeks before the competition.

The testing and quality checks which are the basis for the definition of done of a user story were difficult to realise: We used continuous integration and coding standard as elements of XP for improved code quality. But the use of test driven development and automated unit testing in general is not possible to this extent in a robotics environment. Hence the quality checks were reduced to simulations where possible and tests with the robots, which is quite resource consuming. Also we perceived the issue that mechanical elements broke or at least changed their behaviour over time, e.g. due to wear, which makes the notion of done difficult.

The role of the Product Owner could not be realised until now. Hence the only external checks are the competitions and some workshops. This is not sufficient and we are evaluating other ideas as getting externals as Product Owner, having group events where results are presented or planning presentations to international guests of the university etc. as a substitute. At the moment the Sprint Review is realised in form of a group meeting accompanied by the advisers.

The students experienced the Team role as very helpful. The role of the Scrum Master was taken by one of the authors in the form of a coach.

Instead of Burndown Charts and Scrum Boards we used the ticket tools Redmine [19] in combination with the continuous integration system Jenkins [20] as a source of continuous feedback to the group. Since the parts of the group working together are relatively small it was important to realise the transparency of all projects. The Product Backlog and the Sprint Backlog were also realised via this ticket tool. A monitor with an actual status of the projects is placed in each laboratory. The central idea was to choose tools that are open source and used frequently for the organisation of software projects.

VI. Experiences and Ideas for Adjustments

In general, the feedback of the students was very positive. They appreciated the transparency and communication. The regular meeting was excellent to create a team spirit and allowed them as a group to lead the project. Hence it is a very promising approach for self-organised student projects in robotics. In the following we discuss our experiences and

Fig. 8. Status of Projects in the Laboratory

present the ongoing discussion with the group about adjustments of the methodology.

With a group of volunteers with changing time budgets, the stringency of the Scrum philosophy is problematic. Students are not available full time and sprints may need to be readjusted dynamically to account for unforeseen external and internal to the project distractions. External distractions may result from specific requirements of students to follow lectures, prepare assignments or earn money for a living. This specifically holds for undergraduate and graduate students. As a major internal source of distraction, we identified hardware failures, which required lengthy repair or ordering of replacement parts. Based on the observations, any unforeseen event to cause an extra work load of about 2 days is considered as a distraction. Apparently, any individual shorter distraction can be mostly compensated within a typical sprint period. However, if distractions result in delays and not meeting the time-line too often, the self-improving estimation of effort for specific tasks is jeopardised. Furthermore, not meeting deadlines becomes a regular case. Having a set of identical hardware unit available helps to carry on with a task with less delay. However, identifying deviations from an expected behaviour as a hardware issue still consumes time. Furthermore, hardware units need to be in an equal state. This, however, may be a challenging if not infeasible requirement for hardware systems that suffer from wear or from performance deterioration due to consumption of material, building up of heat or other physical effects.

Fig. 9. RoboCup team during a competition

On the other hand, agility very much is in line with the expectations of a group of volunteers working on a project. Agile methods account for a high degree of self-organisation. With their typical culture of frequent meetings, they foster a high degree of transparency. However, lacking a clear *Product Owner* role during the development phase negatively affects the quality of the results. Often only the competitions take over a *Product Owner* role to identify the major shortcomings of the current implementation. For many of the RoboCup student groups, the authors observed the most agile development phase between the national competitions in spring that communicate clear user requirement and the international event in summer. Additional smaller events may contribute to a 'virtual' *Product Owner*, but need to be considered carefully with respect to the objectives and the time budget.

One of the major problems before the introduction of Scrum was project planning and the estimation of work packages. We noticed that User Stories and discussing only the next Sprint supported the students in the group to structure their work better than before. At the moment many of the User Stories are too long to be finished in one Sprint (so called *Epics*) and therefore could not be finished in one Sprint as intended. This was a compromise since the group realised a complete change to a blackboard architecture. Concerning this issue we will need to extend the length of Sprints from 2 to perhaps 4 weeks and to get experience how to break down User Stories. This is one of our main goals for the next period. Additionally, the group has to decide how often they want to meet to discuss their actual tasks for status and feedback. Since the student group has now experiences in the methodology they are able to decide as a group how to adjust Sprints and additional meetings.

In the third section of our questionnaire, students were asked for recommendations to organise the Scrum approach. There was a clear vote for sprints of two weeks with 'Daily' Scrum meetings once a week. This is in line with an average involvement of about 2 days per week in the robotics group activities. Hence the student group needs to investigate how to break up user stories and must concentrate on estimation of user stories. As an additional tool the Planning Poker will be evaluated.

Although students were often not able to finish their task during the Sprint, the regular meetings helped them to explain the reasons to the entire group and discuss the status openly and straightforward because of the notion of 'done'. Hence the regular meetings improved the communication in the group and the transparency between the different activities. Some of the students addressed that they would like to try a Scrum Board instead of the solution with Redmine used at the moment to enhance transparency further.

Additionally to the more reliable status, the fact that the students had to explain their progress to their friends in the group led to a better commitment of the students that work voluntarily for their projects. These positive effects were also described in the case studies about agile methodologies in a teaching context.

Beside these positive effects of Scrum we still perceive the following issues. The role of the Product Owner is still open. This role is in some ways taken by the student group itself and there is the final check during the competition. But an

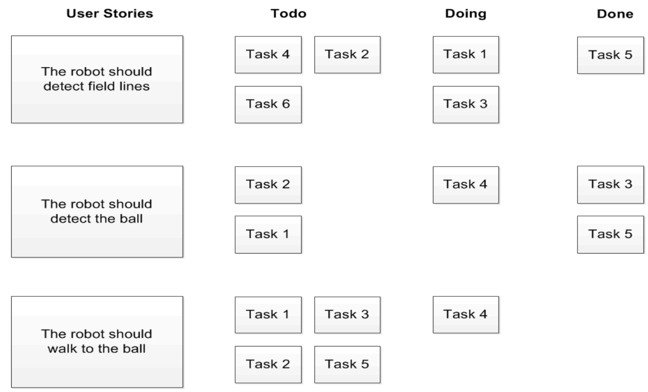


Fig. 10. Example of a Scrum Board

accepted Product Owner outside the group would help to lead the development to the features needed in the competition.

Another issue is testing and quality assurance. In a robotics project automated tests are only possible if there exists a simulation environment. A test with robots is always very resource consuming. Concerning quantity and type of tests, no clear vote could be extracted from the answers in the questionnaire. The issues concerning the Product Owner (resp. Customer in the case XP is used) and testing were also perceived in [13]. In [16] and [14] the instructor is chosen as the Product Owner which is not possible here because the student group should not be guided by an instructor in this setting.

The authors consider carrying out more tests in simulation environments as an analogy to regression testing during continuous integration in software engineering and integrating regular testing of the competition requirements at least at the end of each Sprint. It also needs to be investigated how to reduce the "cost" of testing and how to reach a mindset that test is as important as development and that it is valuable to invest time for testing.

Additionally, a more in-deep investigation of the alteration of hardware over the time is required, because this has an immediate influence on the predictability of effort for user stories and the quality of the resulty.

VII. Conclusion and Future Work

Competitions offer a motivating environment for student-driven projects. The higher the complexity of this environment, the more crucial are the project management methodologies used - not only to ensure good results in the competition, but also to preserve the motivation of the student groups. Here, traditional project management approaches like the waterfall model or the V-model failed.

Agile methods enabled the student group to manage a number of aspects of a complex project in a self-organised way. By introducing basic elements of Scrum, the student group got more control over the project. Starting from this baseline the student group discussed and decided how to address the issues experienced. Hence the student group was enabled to adjust their own way of work to the needs they perceive. We feel that this is an important step to tackle the complexity of advanced robotics competitions while keeping alive the concept of self-organised student groups.

However, there still are a number of open questions, minor shortcomings and incompatibilities between Scrum and a self-organised system development which are mainly the length of Sprints, breakdown of User Stories, the role of the Product Owner, testing and quality assurance and the alteration of hardware over time. Based on the experiences and ideas presented here, these issues will be further investigated in order to allow student groups to shape their own agile methodology based on Scrum.

References

[1] J. Lüssem, F. Pavkovic, U. Samberg, and A. Struck, "Combining learning paradigms to ensure successful learning outcomes in the aera of sotware development," in *Proceedings of 3rd International Conference on Education and New Learning Technologies (EDULEARN 2011)*. IATED - International Association of Technology, Education and Development, 2011, pp. 81–87.

[2] J. Dewey, *Experience and Education*. Touchstone, 1938.

[3] R. Gerndt and J. Lüssem, "Mixed-reality robotics - a coherent teaching framework," in *Proceedings of 2nd International Conference on Robotics in Education (RiE 2011)*, R. Stelzer and K. Jafarmadar, Eds. INNOC - Austrian Society for Innovative Computer Sciences, 2011, pp. 193–200.

[4] "Robocup," http://www.robocup.org/.

[5] J. Dessimoz, Ed., *Cognitics*. Roboptics Editionsr, 2011.

[6] R. Gerndt, M. Bohnen, R. da Silva Guerra, and M. Asada, "The robocup mixed reality league - a case study," in *The Engineering of Mixed Reality Systems*, 2010, pp. 399–418.

[7] "Robocup - soccer humanoid league," http://www.robocup.org/robocup-soccer/humanoid/.

[8] "Robocup@work," http://www.robocupatwork.org/.

[9] "Robot operating system," http://www.ros.org/.

[10] C. A. Kent Beck, *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional, 2004.

[11] K. Schwaber, "Scrum development process," in *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA)*, 1995, pp. 117–134.

[12] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9, pp. 833–859, 2008.

[13] P. Lappo, "No pain, no xp observations on teaching and mentoring extreme programming to university students," *Agile Aliance*, 2002.

[14] A. Shukla and L. Williams, "Adapting extreme programming for a core software engineering course," in *Software Engineering Education and Training, 2002. (CSEE T 2002). Proceedings. 15th Conference on*, 2002, pp. 184–191.

[15] D. Sanders, "Using scrum to manage student projects," *J. Comput. Sci. Coll.*, vol. 23, no. 1, pp. 79–79, Oct. 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1289280.1289295

[16] L. Pinto, R. Rosa, C. Pacheco, C. Xavier, R. Barreto, V. Lucena, M. Caxias, and C. Figueiredo, "On the use of scrum for the management of practcal projects in graduate courses," in *Frontiers in Education Conference, 2009. FIE '09. 39th IEEE*, 2009, pp. 1–6.

[17] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, "The agile manifesto," *http://www. agilemanifesto. org/principles. html. Acesso em*, vol. 7, no. 08, p. 2009, 2001.

[18] K. S. J. Sutherland, "The scrum papers: Nuts, bolts, and origins of an agile process," 2007.

[19] "Redmine," http://www.redmine.org/.

[20] "Jenkins," http://jenkins-ci.org/.

Building a network of schools on Educational Robotics in Tuscany area

Elisa Buselli, Francesca Cecchi, Giacomo Santerini, Pericle Salvini, Paolo Dario
The BioRobotics Institute,
Scuola Superiore Sant'Anna
Pontedera (PI), Italy
e.buselli@sssup.it

***Abstract*—Educational robotics aims at fostering the development of scientific and technological knowledge starting from school level. This work presents the creation of a network of schools on educational robotics in Tuscany, Italy. Robotics Laboratories involved Primary, Secondary and High school students. Different activities were design for each school level, based on the students age and on the school curricula. Examples of case studies are presented, showing how to adapt Robotics to different contests. Results showed that this approach gives emphasis to engaging students to concept, feeding their curiosity, encouraging autonomous exploration and experimental activities to discover concept and theory.**

Keywords— school network; educational robotics; constructivism.

I. INTRODUCING EDUCATIONAL ROBOTICS IN SCHOOL

IN the last decade Educational Robotics has gained a lot of attention among both research institutes and schools. Despite this wide interest, only few schools have included robotics in their standard *curricula* for the lack of both time and specific preparation. Moreover, many high specialized initiatives are presented around the world, but often it is difficult to connect different experiences. This work presents the creation of a network of schools on educational robotics in Tuscany, Italy. Other networks examples at national level refer to associations (such as Scuola di Robotica [1], Fondazione Mondo Digitale [2]) involved on the whole Italian territory with different activities.

The aims of the network is to participate in the education of young generations by providing schools with human and technological resources to carry out several kinds of activities, based on the conviction that robotics can be a useful tool for teaching and learning in a funny and constructive way.

Our approach to educational robotics is strongly characterized by:
1) the promotion of interdisciplinary projects: it seeks to exploit not only the technological and scientific potential of robotics, but also its connections with other school subjects;
2) the generation of a critical attitude towards technology: the assumption is that students should not be passive receivers or users of technology, but they should be taught what is inside the technology and how it works in order to generate in them a more responsible use as well as insights on the possible risks that technology may raise.

The philosophy behind educational robotics refers mainly to Seymour Papert theories [3], which described the advantages of using simple construction kits and programming tools for educational purposes. According to Papert's perspective, children, by using robotic kits could become active participants in their learning and creators of their own technological artefacts, not just users of devices that others had made for them [4].

Our objectives are:

- Improving teacher effectiveness: by enhancing professional development (new teaching methodologies, new educational tools); helping teachers integrate technology into their lessons and promoting students' problem-solving, critical thinking, and collaboration skills.
- Strengthening education at all levels: involvement of children and teenagers from 3 to 19 years old. Starting from introduction to Robotics, to experimental activities, to programming, up to participation in robot contests. In addition, educational activities based on robotic platforms play a relevant role in increasing students' motivation and engagement, favouring learners-centred learning and problem-solving abilities.
- Promoting excellence in STEM (Science, Technology, Engineering and Math) Education: advanced interdisciplinary course on teaching with robotics: Math, Physics, Sciences, Biology, Humanities, Philosophy, Language and Linguistics, Art. Robotics is a multi and inter-disciplinary subject that allows for many connections with school subjects; this bridge allows to reach a larger audience (including different genders).
- Developing better technologies and stimulating the growth of better users: providing pupils with a better understanding of the science and technology inside products of daily use and an overview of the main ethical and social implications of technological and scientific advancement. In a society where innovation and economic success are increasingly tied up to the mastering and responsible applications of science and technology, it becomes urgent to provide the new generations with the basic elements of science and technology to be the future innovators.

We strongly believe that Robotics is the key for pursuing our scope:

- Robotics is attractive and engaging.
- Robotics promotes a learner-centered learning.
- Robotics favors problem solving.
- Robotics fosters creativity.
- Robotics is a multidisciplinary subject and allows developing several knowledge and competences.
- Robotics encourages group work.
- Robotics supports hands-on experiences and experimental activities.

II. THE NETWORK OF SCHOOLS

The Scuola Superiore Sant'Anna (SSSA) team promoted Robotics Education approach since the early 1990s both collaborating with public schools and organizing visits and public events at the BioRobotics Institute. SSSA has collaborated with various schools to participate and organise robotic competions at national level (RoboFesta - IPSIA Fascetti, Pisa; RomeCup - Roma; RoboCup Junior) and at international level. Finally, SSSA has participated in high level robot competitions (i.e. Micro Robot Maze Contest in Nagoya 2005)

From 2010, SSSA team started a more structured activities aimed at creating a solid network of school working on Educational Robotics Laboratories, which strives to foster the development of scientific and technological knowledge starting from school level.

A. *The LELR Network in Valdera*

The Valdera area, is one of greatest economic areas of Tuscany, in Italy. The analysis of the main sectors of the local economy shows an area with great potentialities in the field of innovative technologies. This area is characterized by the strong influence of the mechanical division of Piaggio, the large company known for the Vespa and for other popular brands of two-wheeled vehicles. In Valdera all the Municipalities are members of the Valdera Union which has the aim to jointly exercise a variety of features and services, in order to exploit the potentially competences of the 15 municipalities associated. In particular, in the branch of Education, the Union supports and encourages the creation of a common training system in collaboration with all the institutions, agencies and associations that are present in the area. For this reason, on November 2010 a pact called "Agreement for the Education of the Community" has been signed in order to define a common educational plan to follow the trajectories of the scientific territorial development. This pact, signed by Unione Valdera; Scuola Superiore Sant'Anna, Rete Costellazioni - a local network of schools -, Pont-Tech, and the Municipality of Pisa, will try to encourage the creation of an integrated training system based on Local Educational Laboratories with a shared planning in order to improve education in public schools. The choice of Robotics is not accidental in fact the economy of the Valdera area relies heavily on mechatronic skills and technologies. Fig.1 show the geographical area of the school network.

Fig. 1. Valdera area

The Local Educational Laboratory on Robotics (LELR) has started its activities since December 2011. The laboratory involved six pilot schools: 2 high schools, 2 secondary schools, and 2 primary schools. About 10 tutors, among which PhDs students in biorobotics, robotics researchers and technical staff of the BioRobotics Institute of Scuola Superiore Sant'Anna, have made themselves available for collaborating with teachers in designing and developing robotics related activities. Usually a number of 5/6 meetings between SSSA tutors and school teachers are planned in order to design and carry out the activities. Tutors may be invited to collaborate during school time in teaching activities together with teachers.

Starting from 2010 SSSA team has organized about 10 training courses for teachers and has expanded the LELR from 6 pilot classes up to 12 Institutes (4 primary schools, 4 secondary schools and 4 high schools), 20 Classes for a total of about 450 Students. On school year (2011-2012) more than 80 hours of class laboratories were organized.

B. *The Network growth in Tuscany*

In 2011 the LELR team join a new project on education. The ACARISS (*Acrrescere le conoscenzesull'ambiente e i rischi connessi all'inquinamento coinvolgendo le scuole con la sperimentazione*) project connects students of more than 30 secondary schools around Tuscany, distributed in the provinces of Firenze, Livorno, Lucca, Siena, Pisa, Pistoia and Prato. Among this schools, 14 classes form 11 Institutes join Robotics activities.

Table I shows the growth of the LELR school network among three years.

TABLE I.

Year	*schools*	*classes*	*teachers*	*students*
2010/2011	6	6	8	120
2011/2012	12	20	15	450
2012/2013	14	22	21	500

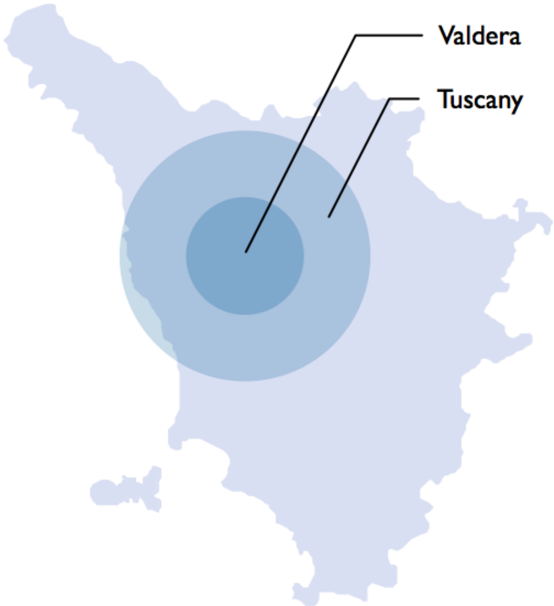


Fig. 2. Tuscany school network

III. CASE STUDIES

Robotics Laboratories involved Primary, Secondary and High school students. Different activities were design for each school level, based on the students age and on the school *curricula*. In the next subparagraphs some examples of case studies were presented, showing how to adapt Robotics to different contests.

A. Primary school

In Primary schools (6-11 years old students) activities were presented as multidisciplinary laboratories.

Some preparatory activities were carried on using the Bee-Bot platform, which can be programmed manually and allows introducing what is a sequence of instructions. Playing with Bee-Bot is a useful step, motivating and appreciated by children at their first meeting with robotics and programming and could be a key aspect for introducing robots in primary schools. Bee-Bot allows conveying different educational contents, in a new and more concrete way without requiring high technical competences by the teachers. Some examples of educational activities are: design of novels and the city of grammar (literature), spatial orientation activities (geography), drawing plane figures (geometry), flowcharts (logic and technology). Bee-Bot is extremely appreciated especially by children with learning disabilities; they are involved in the laboratory on robotics with specific activities and so learn programming and easy verify their work.

Fig. 3. Multidisciplinary activities in primary school

The laboratory on educational robotics with one of these classes started on 2010 and after the preparatory activities here showed, in the last year students (10 years old) faced a new challenge by participating at the First Lego League [5]. This competition is a robotics program for 9 to 16 year olds where students' work is programming an autonomous robot (using the LEGO® MINDSTORMS® robot set) to score points on a thematic playing surface and creating an innovative solution to a problem as part of their project. The students raised a third place at the interregional qualifications arriving at the national finale in Rovereto on March 8-9, 2013. This is an example of improvements of students' interests and skills in robotics and a good schedule of educational activities by teachers along the three year of laboratory. Moreover, teachers reported all the activities on a website blog, where students parents can share pictures and comments [6].

B. Lower Secondary school

In Lower secondary schools (11-14 years old students), the LELR was strongly connected with Math's *curriculum*.

The activities in a first class (11 years) with the Pro-Bot platform, is presented as an example of combining Robotics with Math. This robot can be programmed manually by a keyboard and can draw a line when it moves. This solution was adopted to study geometric shapes and angles. Students should give the right sequence of instruction (direction, time and rotation angles) in order to design geometric pictures. For example, students tried to write the right instruction to draw the word "ROBOT" (Fig.4). This activity supports geometry, Cartesian coordinate system, measurement instruments, ability to sustain opinions on hypothesis and trial and error approach.

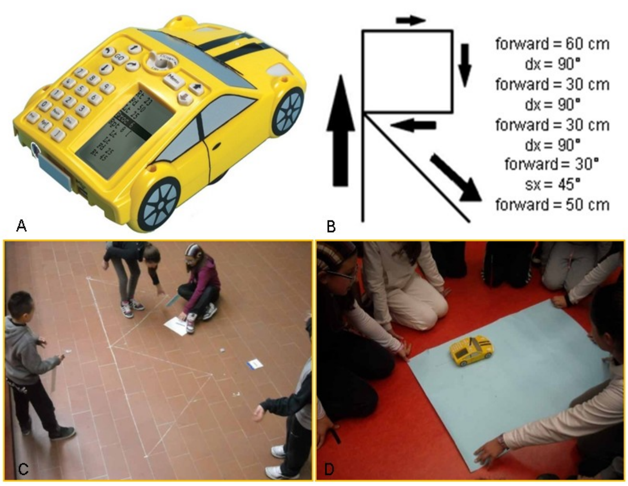


Fig. 4. Pro-Bot activities connected to geometry

Another activities was carried out in a third class (14 years) on the law of motion. The main goal in that experience is to let students understand the concept of Average Speed. Average Speed is defined as the distance covered by a moving object divided by the time it takes to cover that distance. To let students understand that concept, a hands-on methodology was used: we measured the average speed by mean of a Robot moving in straight line on a blank table, on which there were sticked some reference stripes. Students were invited to annotate the instant of times in which the robot passed the stripes, using a chronometer. In order to let students appreciate the concept of Robot, and not as a blind moving thing, we have equipped it with a pair of sensors: A light sensor and a touch sensor. These sensor were used during the experiment: The light sensor stimulated a 'BIP' sound once the robot passed a black stripe, whilst the touch sensor was used to stop the robot motors once it reached the end of the path, hitting and obstacle. The robot was programmed with 3 different speeds, and the measurements were repeated three times for each speed. Every measurement was carried on by a different student: this served to explain the concept of Measurement Error, and the uncertainty of the measured physical quantity.

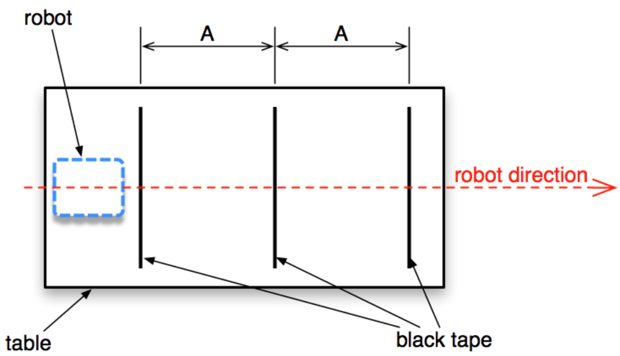


Fig. 5. Schematic for the activities on the law of motion

Fig. 6. Experimental activities on the law of motion

C. *High school*

High school students has a different background and it is possible to introduce more complex subjects.

In particular some activities were developed in which last year students (19 years old) could participate in the work of a real research laboratory in the framework of a national program called “alternanza scuola lavoro”. This program is based on a network including educational choices of the schools, professional requests of the local companies and on the personal formative needs of the students.

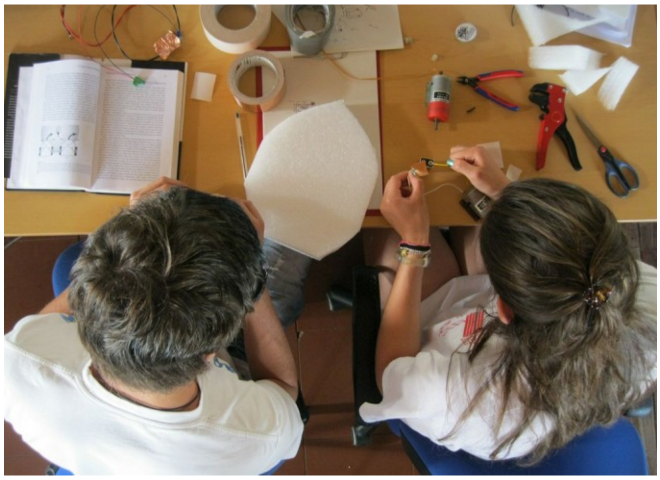
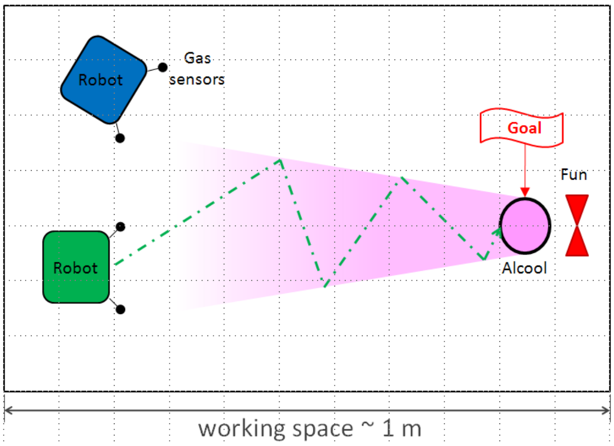


Fig. 7. High school students working in the laboratory

From the collaboration with a Industrial and Technical Institute [7] a new robot was created purposively developed by the school. The work started from the task to build a robot that could detect gas vapors. The robot included obstacle avoidance sensors, gas sensors and implemented a dedicated algorithm to gas source detection.



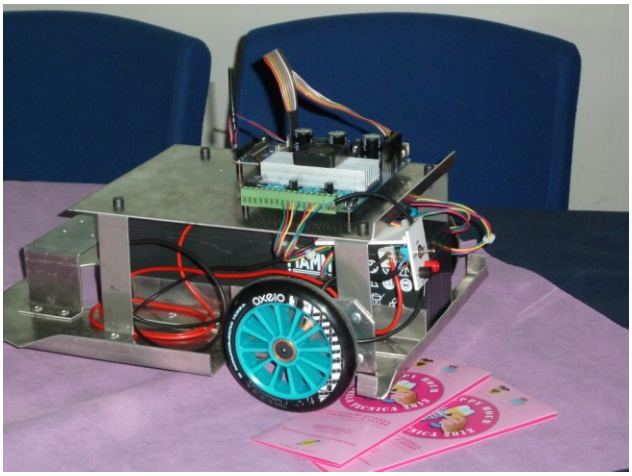


Fig. 8. Robot built in collaboration with a Industrial and Technical Institute

IV. Conclusions

This work presented the creation of a network of schools currently running in Tuscany on educational robotics. Lessons of robotics are carried on in more than 20 classes of 14 Scholastic Institute. This approach gives emphasis to engaging students to concept, feeding their curiosity, encouraging autonomous exploration and experimental activities to discover concept and theory. The students show a strong enthusiasm in using robotic tools, thus it could be an effective means to transmit scientific contents. A quantitative analysis of the outcomes is currently on-going, in order to evaluate the advantages of this new approach with respect to traditional teaching methodologies. Preliminary results on 3 class of Lower Secondary School reveals that students are really engaged: 89% affirm that he/she liked the activities, 75% of assert that he/she would like to use this type of approach also with other subject and 67% declare to be willing to study scientific or technological subjects in future. Regarding the teachers, they all evaluate the activity as very efficient; on the other hand they complain about their small autonomy in the use of the robot and suggest to organize focused courses for teachers.

Acknowledgment

This work was partially realized in the framework of the ACARISS project, funded by the Regione Toscana - PAR FAS 2007-2013, azione 1.1.a.3, Fondo per le Aree Sottoutilizzate, Delibera CIPE 166/2007

References

[1] http://www.scuoladirobotica.it/

[2] http://www.mondodigitale.org/

[3] S. Papert, Mindstorms, Children, Computers and Powerful Ideas. New York: Basic books, 1980

[4] F. Martin, B. Mikhak, M. Resnick, B. Silverman and R. Berg “To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines”, Morgan Kaufmann Series in Interactive Technologies, Robots for kids: exploring new technologies for learning, 2000, pp. 9-33.

[5] http://www.firstlegoleague.org/

[6] http://errecomerobot.blogspot.it/

[7] http://www.youtube.com/watch?v=1obe9Xg8fO8

A simple computer vision based indoor positioning system for educational micro air vehicles

A. Ferner, R. Scheikl, T. Kittenberger
Department of Embedded Systems
University of Applied Sciences Technikum Wien
Vienna, Austria
thomas.kittenberger@technikum-wien.at

***Abstract*—Computer vision is one of the main research fields for micro air vehicles. Therefore many teams around the globe are developing different camera systems which are directly mounted on a MAV or observe and track it from a fixed position. This paper presents a simple and cheap solution for a lightweight and precise onboard computer vision system based on a 6 gram Gumstix Overo Firestorm computer, an infrared camera and three ground based high power infrared LED's. It has great potential for educational applications because no fixed and calibrated camera installation around the flight area is necessary so it can easily be deployed and removed. The use of various filter algorithms makes it very reliable and the images are processed with high speed. The achieved accuracy is about 2% and the range was tested up to 10 m.**

***Index Terms*—Computer vision, Gumstix, infrared, high power LED, robotics education, UAV, MAV, resection**

I. Introduction and Related work

In recent years small mobile ground based or aerial robots like line followers, sumo and soccer robots or quadrocopters have become a very popular means to teach technical topics with project based learning at universities. The Department of Embedded Systems at the University of Applied Sciences Technikum Wien founded its Micro Air Vehicle Team in 2007. The team developed different kinds of inertial, ultrasonic and GPS based navigation systems, flight control computers and aerial robots [1] [2] and won in 2010 and 2012 two third places at the RobotChallenge, the largest European robotics competition.

The type of contest we attended in 2012 was the new "Air Race". In the semi autonomous class the aircraft had to fly for 10 minutes as much figure eight patterns around two poles as possible with computer controlled attitude and height. Only the flight path was controlled manually. For the next year's contest it was decided to participate at the autonomous Air Race where, for the same task, no human interaction is allowed [3]. So it was necessary to find an adequate solution for positioning in an indoor space with a size of about 8x12x4 m which could be used at the contest but also in different large class rooms and labs at our university. After a first glance at different available technologies like ultra wide band radio, ultrasonic or vision based indoor positioning systems it was decided to use a simple vision based solution with some kind of easily detectable markers. This approach opens the way to a subsequent operation in natural environments without using explicit markers whereas the other solutions always rely on some kind of infrastructure in the flight area.

A simple and widely used approach to vision based indoor positioning is to use a downward oriented camera on the Micro Air Vehicle (MAV) and some kind of easily detectable markers on the floor. This setup was used for instance by the ETHZ Pixhawk team at the IMAV 2010 competition [4] or by all three winner teams of the autonomous Air Race at the RobotChallenge 2013. The drawback of this approach is that with limited ground clearance in indoor environments and limited camera aperture angle a lot of markers are necessary to cover larger flight areas. The precise deployment and the removal of the markers is time consuming and in the narrow time slots of a competition maybe not possible. The organizers of the RobotChallenge therefore provide a dashed figure eight line for their participants.

Another interesting solution was introduced by Kara in [5]. To determine the position, the MAV sends three slightly outward oriented laser beams down to the ground. The dots on the ground are captured by a fixed camera and a computer vision algorithm calculates the position which then is sent back to the MAV.

The Vicon MX motion capture system is a popular high end commercial solution offered by the UK based company Vicon. Universities like the Swiss ETHZ [6] [7] or the MIT Aerospace Controls Laboratory (RAVEN) use these systems for their development of autonomous aerial robots. The basic system consists of eight high speed, high resolution infrared cameras mounted on the ceiling of the flight area, a high speed network to connect the cameras and a central computer to merge the information from the cameras to attitude and position data for the observed MAV's. The camera lenses are surrounded by a ring of IR LED's illuminating the scene in front of the camera. On the MAV's four balls covered with a retro reflective coating reflect the light directly back to the cameras. The light passes through a daylight blocking filter and produces black pictures with bright white dots for the detected balls in the field of view. By combining the information from several cameras, position and attitude of several aircrafts can be computed with a rate of 200 Hz, a delay of 30 ms and centimetre precision.

II. System Concept

The computer vision system we had in mind on our survey should provide good value for money and it should be easily portable because at our university we can not afford a large classroom with fixed installations dedicated to only one project. The Vicon solution needs a lot of permanently installed and calibrated cameras and costs several ten thousands of Dollars. Also the solution with the laser pointers on the MAV needs several cameras if used in a larger flight area and if the MAV rolls and yaws. The solution with markers on the floor and only one camera on the MAV is much more cost effective but has also the drawback of the necessary marker deployment and removal.

Our approach is now to use one small CMOS camera on the MAV that is directed forward and three position lights on a tripod in front of the MAV. Figure 1 shows the concept with the tripod in front of a single rotor hovering MAV.

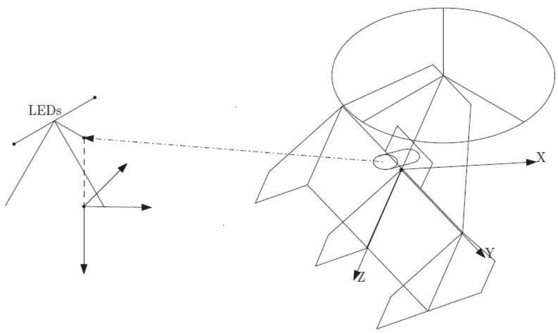


Figure 1: Setup with a computer vision equipped MAV and a tripod with position lights in front of it.

The tripod (see figure 3) holds three arms directed to the left, the right and to the front in an exact horizontal plane. Each arm is 50 cm long and holds a wide angle high power infrared LED at the end. If the infrared camera is exactly in front of the tripod centre it sees three equally spaced lights in line. If the camera is moved to the right, the middle light also moves towards the right light and vice versa. If the camera is moved above the LED plane the middle LED is below the connection line of the left and right LED and also vice versa. Because in figure 3 the middle LED is near the right LED and below the connection line of the left and right LED it is clear that the camera is left above the tripod. The distance to the tripod can simply be deduced by the distance of the left and right LED on the camera picture. The more the camera is moved away the smaller the LED triple gets.

With this concept it is possible to measure the exact position of the camera relative to the position lights in a wide area in front of the tripod. The MAV has only to take care to look always towards the position lights but for all kinds of hovering aircrafts this should be no problem. If necessary the flight area can be equipped with several sets of position lights so that the MAV is freer in its movements. And this additional position lights will definitely be cheaper than additional cameras.

III. Key Components

For the infrared LED's OSRAM SFH4235 are used. They are high efficiency and high power IR LED's with a wide half angle of +/- 60 degrees. The maximum continuous current is 1 A at 3 V which produces an optical flow of typically 950 mW at 860 nm. Figure 2 shows the SFH4235 connected to the aluminum square tube arm for sufficient cooling.

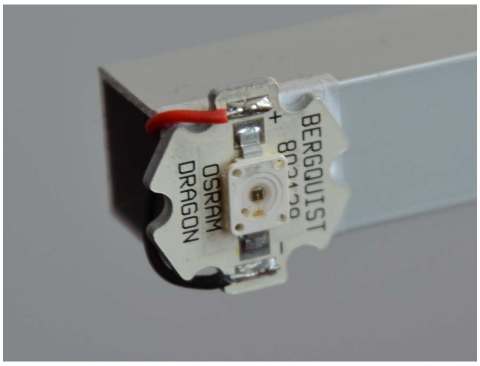


Figure 2: OSRAM SFH4235 high power infrared emitter.

Figure 3 shows the tripod on which the LED arms are mounted. The LED arms and the tripod legs can be folded so that the equipment can be stored space saving. If needed it is setup in minutes and can cover even very large classrooms.

Figure 3: Tripod with IR-LED's.

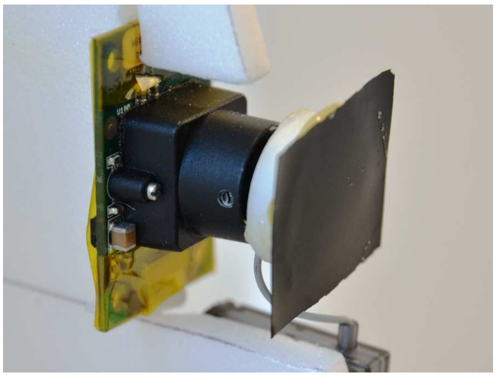


Figure 4: CMOS camera with daylight blocking filter.

The CMOS camera used is a Gumstix Caspa FS for full spectrum imaging priced at 75 USD and weighing 23 grams. It features a 1/3-Inch Wide-VGA CMOS Digital Image Sensor and comes with a 3,6mm lens, aperture 2,0 and the angle of view is 94.91˚ / 68.46˚. The sensor is an Aptina MT9V032 and it is capable of taking 60 pictures per second at a resolution of 752 x 480. Figure 4 shows the camera with an additional daylight blocking filter made of unexposed Fujichrome color reversal film that increases the contrast of the pictures significantly. The Fujichrome film is glued on a 3D printed lens adapter.

The heart of the computer vision system is an Gumstix Overo Firestorm Com computer on module. It has a Texas Instruments DM3730 SOC that is typically used for mobile devices. It offers camera interfaces, a powerful 1 GHz ARM® Cortex™-A8 Core as a general purpose processor and a Video DSP. The whole Gumstix computer weighs only 6 grams and is priced at 189 USD.

Figure 5: Gumstix Overo Firestorm computer on module.

Figure 6 shows the complete avionic system of our MAV. The camera with the Gumstix computer is on the upper left side, the middle board holds the 9-DOF Inertial Measurement Unit and the board on the lower right is a ublox LEA-6T GPS receiver with a helix antenna. The bottom board below the three plug on boards holds the Cortex-M3 avionik processor that controls the servos and motors of the aircraft, processes the IMU and vision data and handles the communication.

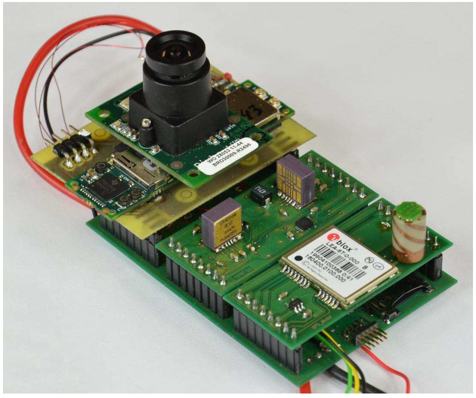


Figure 6: Avionik system with Computer Vision, Inertial Navigation System and GPS receiver.

For better handling and connectivity the Gumstix Tobi expansion board is used for software development and education (figure 7). However due to its weight and size it is not suitable for the MAV where we use our own circuit board. The expansion board has connectors for HDMI, 10/100baseT Ethernet, USB and a 40-pin header with GPIO, PWM and A/D lines. For protection and better handling all electronic parts and the camera are mounted in an ABS-casing.

Figure 7: Development setup for the Gumstix computer with Tobi expansion board and Caspa camera.

Because all parts used in our system are low cost of the shelf parts the total cost for the computer vision system is less than 350 $.

Item	**Price[$]**
Gumstix Overo	189
Tobi Base Board	69
Caspa Camera	75
Flex ribbon cable	5
ABS-Casing	5
Total	343

Table 1: Costs of embedded camera system.

IV. OPERATING SYSTEM, SOFTWARE AND TOOLS

The operating system running on the Gumstix is an embedded Linux distribution called Ångström. It is loaded from the SD-card. The custom built image has already all required tools and libraries installed. The boot parameters passed from u-boot to the kernel are also stored on the SD-card in the boot.scr file. This means that a student receives a Gumstix computer, a Tobi baseboard and a memory card with the image and is up and running in five minutes. The operating system uses preconfigured static IP address and the user can connect using SSH over a direct Ethernet connection. This makes the system very flexible. In the process of developing programs for a UAV a computer with Ubuntu is used and the video output is streamed over the network to the connected PC. This setting offers a convenient work environment with a minimum effort, once the image is configured and compiled.

Due to the processing power of the processor, all applications are written and compiled directly on the target. For the development of the programs that are executed on the Gumstix Geany is used. It is a mall and lightweight integrated development environment (IDE) that is easy to use and runs on the target. This makes it the system of choice for compiling and executing the code.

For image processing an open source library called Open Source Computer Vision (OpenCV) is used. OpenCV is a free-to-use library for academic use. The library offers a wide range of algorithms for video and image processing. Today more than 2500 optimized algorithms are published with the OpenCV library. Our signal processing is based heavily on OpenCV. It also supports different programming languages and provides interfaces in C, C++, Java and Python. The library is widely used and perfect for use in education. Documentations and examples are easy to find [8].

The implementation of a highly reliable computer vision system is one of the main goals of this project. We achieve this by increasing the contrast of the LED's to their ambience to a very high level and by applying several filters in the image processing. To suppress unwanted daylight we use a daylight suppression filter made of an unexposed Fujichrome color reversal film to pass only infrared light to the camera lens. The infrared LED's, that are tracked, are also extremely bright and have therefore a high contrast to their environment. Figure 8 demonstrates an image with a long exposure time and the filter. The environment can still be seen in dark shades, the LED's are heavily overexposed.

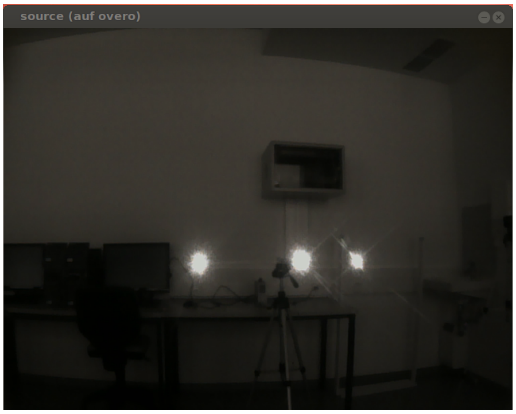


Figure 8: Image with long exposure time and filter.

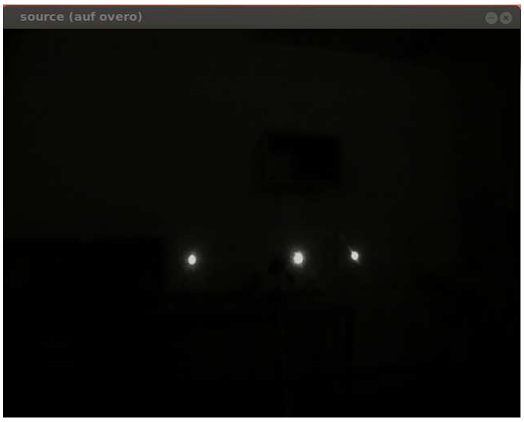


Figure 9: Image with short exposure time and filter.

By further manual reduction of the exposure time we got the picture in figure 9. With this setting also disturbances by light bulbs, fluorescent lamps or sunlight are suppressed very well.

As can be seen it is easy to track the objects now. The short exposure time also influences the size of the dots. They are smaller when the exposure time is shorter and the detected position in the picture is more precise. Another great advantage of very short exposure times is the robustness against blur because of camera movement. All our pictures are crispy sharp now, even if the MAV flies fast turns.

Many CMOS cameras include infrared blocking filters. For our project a camera without such a filter is necessary. We use the Gumstix Caspa camera. It is available in two versions, the Caspa VL (visible light) and the Caspa FS (full spectrum) which is used here in conjunction with the Fujifilm daylight suppression filter.

V. Image Processing

The OpenCV library provides a function for loading the picture from a memory of a lower layer. The first step of the signal processing chain is to transform the loaded picture to a grey-scale image. Setting a threshold and convert the picture into black and white reduces the information of the picture. One main character of this method is that the contrast of the object is very high in comparison to the background. Selecting a threshold value is done by a histogram analysis. A function of OpenCV finds contours in a black and white picture. Circles are placed over these contours, where the characteristics like radius or centre coordinates are known. The plausibility check is necessary to find the three correct objects that are searched. To find the correct position of these points a restitution of the coordinates is necessary. Finally the position determination can be calculated by using a 3-point resection. Before sending the information to the aerial vehicle the position is checked if the value is probable and possible. An overview of the image process is pictured in figure 10.

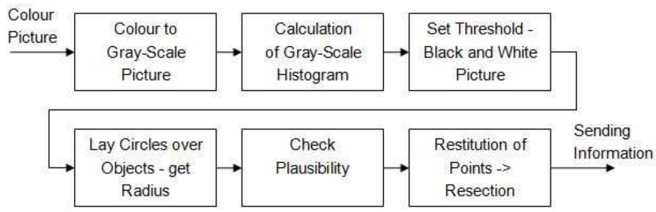


Figure 10: Overview of the image processing.

Histograms are often used to analyse images. A histogram for a grey-scale picture shows the brightness values on the x-axis. The numbers of pixels that have this value are shown on the y-axis. The sum of the y-values is the number of pixels of the image. This characteristic can be used for setting the threshold value. The infrared LED's are shown as white objects at the grey-scale image. The rest of the image is dark or nearly black. Figure 11 shows how a typical histogram in our setup looks like. Left there is the value 0 (black) and right is 255 (white). The upper histogram shows the complete distribution of the pixel values. On the left side the large number of black pixels can be seen and on the right the view white ones. As it can be seen in the figure, the number of dark pixels is very high. The lower histogram shows the distribution of the pixels zoomed into the white pixel area. In this area a peak can be detected, which is caused by the infrared LED's. The threshold is set at a value shortly below this peak.

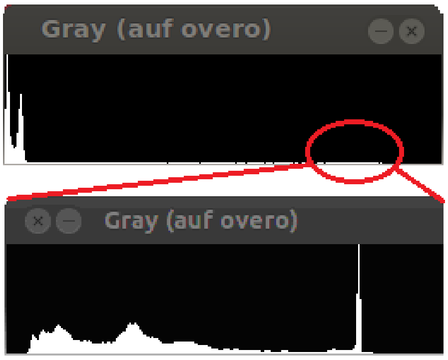


Figure 11: Full histogram and histogram detail for bright pixels.

If an error exists, because of sun, lights or reflections, it is necessary to check the plausibility of the detected contours. Filters and check algorithms raise the reliability to find the correct objects. An easy, but very useful filter is to check the radius of the objects. If a radius is too large or small the objects can be dropped. An attribute of the objects is the similarity of radius of the correct objects. The tracked objects get sorted by their radius. The three correct objects should be located near each other in this array. An algorithm calculates the average radius of the actual three objects. After that the maximal difference between the average radius and the radius of each object is calculated. This operation is done for every combination in the array. If five objects are tracked, the operation has to be done three times. When the operation is finished for every combination, the combination with the smallest difference is selected. This algorithm has a short run time of only 7 ms and supports a higher reliability of the computer vision.

High performance and low delays are important for computer vision systems. Many autonomous aerial vehicles like quadrocopters rely on information from computer vision. This information has to be up-to-date and should be available with a delay that is as short as possible. A significant contribution to the image processing time is caused by the correction of the lens distortion. One step to a short delay is to reduce the time for distortion correction. A standard algorithm corrects the distortion of a complete image. In our system the white LED spots are detected and measured in the distorted picture and only the centre coordinates are undistorted. OpenCV provides a method for distortion correction of individual pixels that is called cvUndistortPoints. The usage of this method cut down the time for correcting the distortion more than 60%.

For the calculation of the camera position, also called the three point pose estimation problem, the resection method is used. There are a number of direct and iterative ways for the solution with the first published already in the 19th century [9]. They are used in different applications like camera calibration, robot navigation and computer vision. The basic principle of all these different methods is that there are at least three points (objects) on the picture whose positions in space are known. The 3D coordinates of the three points and the 2D coordinates of their projections are used to calculate the position of the camera. The iterative method used here was developed as part of a master's thesis [10] at our department. In the project of the MAV team the infrared lights are the reference points (A, B and C) for the aerial vehicle (N) (figure 12).

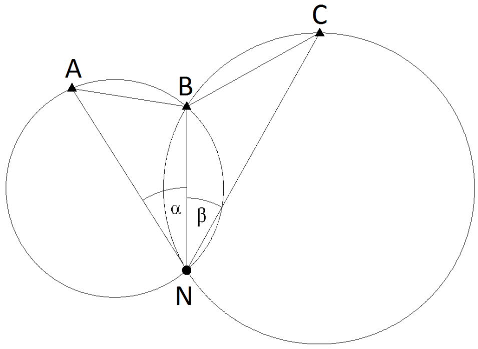


Figure 12: Resection in 2D with the known points A, B and C.

In a first step the three angles α, β and γ between A, B and C, seen from N by the camera, are calculated out of the picture coordinates A'B'C' of the three points. The picture coordinates are undistorted using the camera calibration data (OpenCV chessboard camera calibration) before that.

The side lengths of the projected triangle A'B'C' are calculated by formula 1. This is done in 2D with the positions of A'B'C' in the projection surface of the camera.

$$\bar{s} = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \qquad (1)$$

The distance from the camera center N to the projection surface is the focus length f. So the Euclidean distance from N to, for instance, A' can be derived by formula 2.

$$\overline{NA'} = \sqrt{A'_x * A'_x + A'_y * A'_y + f * f} \qquad (2)$$

With all lengths between N, A', B' and C' now given, the angles α, β and γ can be derived with the law of cosines (formula 3) like in formula 4.

$$c^2 = a^2 + b^2 - 2ab * \cos\gamma \qquad (3)$$

$$\gamma = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{ab}\right) \qquad (4)$$

A remaining problem with α, β and γ is, that the angles are the same when B is above the plane ANC or below it. For that reason a set of alternative angles φ_1, φ_2 and φ_3 is used by defining a base point X' between A' and C' so that the connection from A' to C' is normal to the connection from X' to B'. The three alternative angles are then defined between A' and B' (φ_1), A' and X' (φ_2), and between X' and B' (φ_3). The sign of φ_3 is now holding the information if B is above or below the plane ANC.

The second step in our resection method is to iteratively assume a camera position n that gives the same angles φ_1, φ_2 and φ_3 as they were measured with the camera picture at camera position N. It is easy to derive the three angles because all positions (n, A, B, C) are known. The starting point n_0 is set some meters in front of the infrared LED's ABC.

The approximation technique that is used is the inexact Newton´s method. Formula 5 gives an example for the one dimensional case. The term $(f'(x_n))^{-1}$ is the reciprocal value of the first derivative of the function.

$$x_{n+1} = x_n - \frac{1}{f'(x_n)} f(x_n) \qquad (5)$$

Because we have a three dimensional problem the Jacobi-matrix is used instead of the first derivative to calculate the iterations (formula 6). At the beginning of the approximation it can happen that the inverse of the derivative gives too large values. Because of that, the factor p (0 < p <= 1) reduces this value in the first four iterations of the algorithm.

$$x_{n+1} = x_n - p * (J(x_n))^{-1} f(x_n) \qquad (6)$$

After 10 iterations a position is estimated that is nearly correct. In figure 13 a simulation of our resection method is demonstrated. The red dots symbolize the infrared LED's, which are the reference points. The position of them is known. The green dots are the iterative steps to the correct position of the camera which is shown as a black dot. The start position for the iterative algorithm in this case is [0 0 0]. The last iteration steps cannot be seen because they are inside the black camera dot.

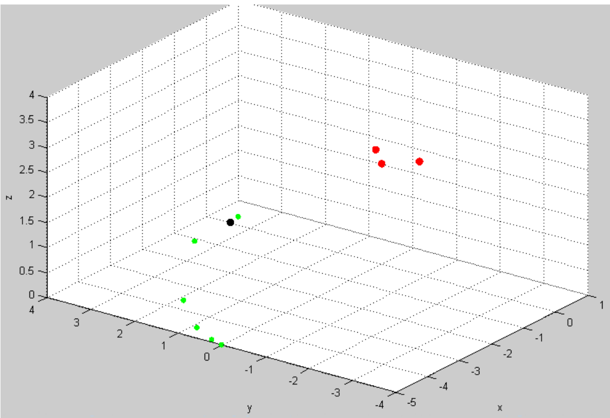


Figure 13: Iterative camera position estimation.

VI. Results

The computer vision system of the MAV project provides a low budget solution with a good performance. With a frame rate of 12 pictures per second and a delay of only 130 ms the computer vision system covers a wide range of applications. In closed rooms the reliability was nearly 100 %. Of course a difficult environment can influence the reliability significantly. For indoor use it provides sufficient performance characteristics. In figure 14 a difficult case for the computer vision system is shown. The left image shows the original image made by the camera with a short exposure time and the filter. This picture shows the three IR LED's in front of a window. The adjacent building is in direct bright sunlight at noon. The image in the middle represents the contours of the objects that were identified. The right image shows the correct results. The plausibility check detects the correct objects and the position information is ready for sending to the navigation computer.

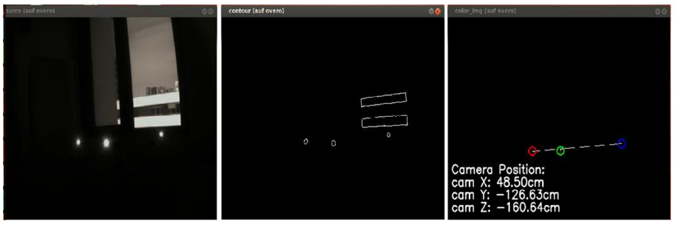


Figure 14: Camera view and picture analyses.

The achievable range of the system is limited with the exposure time. The distance between aerial vehicle and LED's was successfully tested at 10 m. This distance can be expanded to more than 10 m by choosing a longer exposure time but with reduced detection reliability. In this project a high reliability is more important than a long range.

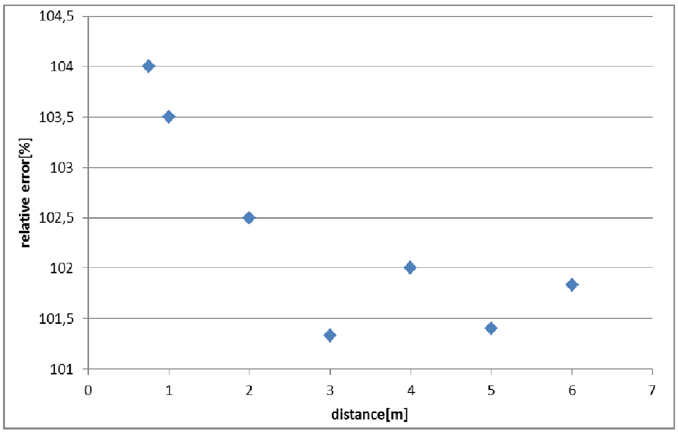


Figure 15: Precision of computer vision range finder.

The accuracy was tested at different distances. The results in figure 15 show, that the imprecision is very small. Also at small distances the failure is not more than 2-3%.

VII. Conclusion

Although the computer vision system was originally developed to work on board a MAV it has also great potential for other projects with educational focus. Its reliability and precession combined with its small dimensions and low weight makes it the ideal choice for aerial vehicles. The used hardware and software makes it very easy to write programs once an OS image is created with the required tools and software. The system has also a high potential for education usage. The startup is well supported and also the OpenCV library is widespread. OpenCV knowledge can also be used for many other projects in the area of image processing.

REFERENCES

[1] G. Freudenberger, "Sensor Fusion für ein MEMS-basiertes Trägheits-navigationssystem", Master’s thesis, University of Applied Sciences Technikum Wien, 2012.

[2] N. Vavra, "Guidance, Navigation and Control of a Micro Air Vehicle", Master’s thesis, University of Applied Sciences Technikum Wien, 2012.

[3] RobotChallenge Organization Team, "RobotChallenge 2013 Air Race Rules", http://www.robotchallenge.org/competition/

[4] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, "PIXHAWK: A System for Autonomous Flight using Onboard Computer Vision", IEEE International Conference on Robotics and Automation, 2011.

[5] M. Kara, S. Patra, A. Lanzon, "Designing Simple Indoor Navigation System for UAVs", in 19th Mediterranean Conference on Control and Automation.

[6] G. Ducard, "Autonomous quadrotor flight using a vision system and accommodating frames misalignment", in Industrial Embedded Systems, 2009, SIES '09, IEEE International Symposium.

[7] S. Lupashin, "The Flying Machine Arena as of 2010", in Robotics and Automation (ICRA), 2011.

[8] I. Culjak, "A brief introduction to OpenCV", in MIPRO, Proceedings of the 35th International Convention, 2012.

[9] R. Haralick, C. Lee, K. Ottenberg, M. Nölle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", International Journal of Computer Vision, 13, 3, p. 331-356, 1994, Kluwer Academic Publishers.

[10] R. Zarits, "Kamerabasierte Positionsbestimmung und Speicherung der Bilddaten auf einem Micro Air Vehicle", Master’s thesis, University of Applied Sciences Technikum Wien, 2012.

Overview of Technologies for Building Robots in the Classroom

Cesar Vandevelde
Dept. of Industrial System & Product design
Ghent University
Kortrijk, Belgium
cesar.vandevelde@howest.be

Jelle Saldien
Dept. of Industrial System & Product design
Ghent University
Kortrijk, Belgium
jelle.saldien@howest.be

Maria-Cristina Ciocci
Dept. of Industrial System & Product design
Ghent University
Kortrijk, Belgium
maria-cristina.ciocci@howest.be

Bram Vanderborght
Dept. of Mechanical Engineering
VUB
Brussels, Belgium
bram.vanderborght@vub.ac.be

***Abstract*—This paper aims to give an overview of technologies that can be used to implement robotics within an educational context. We discuss complete robotics systems as well as projects that implement only certain elements of a robotics system, such as electronics, hardware, or software. We believe that Maker Movement and DIY trends offers many new opportunities for teaching and feel that they will become much more prominent in the future. Products and projects discussed in this paper are: Mindstorms, Vex, Arduino, Dwengo, Raspberry Pi, MakeBlock, OpenBeam, BitBeam, Scratch, Blockly and ArduBlock.**

Keywords—Do-It-Yourself; Education; Open Source; Open Hardware; Project Based Learning; Rapid Prototyping; Robotics; STEM

I. Introduction

The demand for engineers in Europe has almost tripled over the past six years, while the number of engineers who graduate from universities and colleges in that period decreased drastically. Different engineers associations, such as the VIK (Flemish Association of Engineers) assert that the drop in number of students is connected to an image problem of STEM-related knowledge domains (Science, Technology, Engineering and Mathematics) as perceived by pupils and students. Action plans focused on promoting access to the engineering profession are set up, following the highlights of the rapid developments in this field. In addition to this, new pedagogical approaches are also experimented with, e.g. [1-3].

Studies suggest that certain teaching strategies may foster the STEM participation and achievement and give some evidence that using scientific equipment and hands-on activities are related to higher science and mathematics achievement [4]. As a consequence project based education and problem-based learning are becoming the innovative approaches to engineering education and fundamental science education [5, 6]. Also, it seems that there is a precise relationship between new technologies and new pedagogical methods; even though this relationship is complex and not only instrumental, the simplest explanation is that new technologies are the instruments to realize new pedagogies. According to the model of micro worlds of Papert [7], there is a strong link between mental acquisition of knowledge and actual manipulation of the objects of knowledge. Simply put: one learns by doing. Nowadays, this pedagogical prospective meets a new frontier with the commercialization of programmable toys (e.g. Lego Mindstorms), the upcoming of affordable DIY electronics (e.g. Arduino), and the rise of FabLabs equipped with 3D-printers and laser cutters. These systems make it possible to design and construct real robots whose working is determined by a computer program. From the moment that robotics entered our houses and started influencing our everyday life it has become important for everyone to have at least a basic understanding of such technologies. To introduce robotics in schools, there is a methodological choice in the contraposition of top-down teaching and bottom-up learning.

Building robots is a popular project choice for the implementation of problem-based learning (PBL) in classrooms. The reason why it is such a popular choice can be explained by the multidisciplinary nature of the topic: robotics requires many different scientific, technical and technological skills, such as physics, electronics, mathematics and programming. It is an ideal subject because so many different courses can be linked to it [8]. Additionally, robots themselves capture the imagination of children and teenagers, providing inspiration and motivation [8].

The PBL approach in general and the use of robotics in education in particular have a number of other differences with more traditional ways of teaching. Whereas math problems typically have one, and only one correct answer, PBL emphasizes that most real world problems have many different solutions. With PBL, students learn to deal with these real world problems using creative problem solving, an important real-life skill. In addition to technical skills, the PBL approach also allows the students to learn important social skills, such as communication, leadership, planning and cooperation [9].

The relatively new spectrum of technologies that can be used to implement robotics within a standard educational curriculum can become very overwhelming for educators and their administrators who are trying to decide what is possible and what the costs will be to begin such programs. It is important that costs can be afforded without any special purpose grant. This does not mean that robotics is out of reach. Several options exist that can be leveraged to achieve very effective results.

There are two ways to implement robotics in an educational context, either by starting from an existing robotics kit or by building the entire robot from scratch. Building a robot from scratch is typically much more difficult, thus using a kit is a more popular choice, especially in projects involving younger students. These robot kits provide everything needed to make a functional robot, such as building elements, motors, sensors, instructions, a programmable microcontroller, and the software to program the robot. While this provides a great starting point, this solution is typically more expensive and less flexible than a fully customized robot.

In recent years, making a robot from scratch has become much easier, in part due to the many different products and platforms that implement some of the elements that are required to build a robot. They can be categorized into 3 large groups: software, electronics and hardware projects. Historically, software has been the easiest to share as Open Source because collaboration can be done easily over the internet, because development tools are readily available and because there is virtually no cost associated with copying or modifying software. Online platforms such as GitHub greatly facilitate this [10]. However, in recent years the electronics and hardware projects have taken a jump forward in a phenomenon sometimes referred to as the Maker Movement [11]. The Maker Movement is a trend that can be described as a high-tech extension of the DIY and Arts & Crafts subcultures. It is characterized by the use of CNC machines such as laser cutters, 3D printers and CNC mills for the development and replication of Open Source hardware. These CNC machines are often low-cost devices that were developed by open source communities [12, 13].

II. Challenges and Pitfalls

While PBL using robotics offers a promising alternative to the traditional teaching methods, implementing this on a large scale in education does pose several challenges. Mataric et al. [14] describe 5 big challenges:

- "Lack of teacher time"
- "Lack of teacher training"
- "Lack of age-suitable academic materials"
- "Lack of ready-for-use lesson materials"
- "Lack of a range of affordable robotics platforms"

Besides these challenges, we also detected gender issues in the context of robotics in education. Presently, robotics and other technological hobbies are usually associated with boys. Girls are often subtly discouraged and told to pursue other interests. As a result, women are underrepresented in STEM-related fields. Studies have shown that while girls will not always focus on the same aspects of building robots as boys, they show just as much interest in the topic [8]. Consequently, robotics - if properly approached - can serve as a way to increase the number of women in technical and technological fields [15].

III. Complete Robotics Kits

Historically, robotics in education is usually implemented using premade robotics kits that include everything needed to build a functional robot. While this is a great way to get up and running quickly, it does include several disadvantages.

- All-in-one kits are generally more expensive.
- It's hard to interface these kits with other components, such as standardized components, components made by third parties or off-the-shelf sensors and actuators.
- Finding or buying replacement parts is not always easy.
- Not all components are used or needed in a robot, so you are paying for components you do not need.

That being said, these systems are a great starting point as they provide a set of compatible building blocks and electronics, software for easy (graphical) programming, instructions and a community network.

A. Lego Mindstorms

Fig. 1. Lego Mindstorms NXT [16].

Mindstorms [17, 18] is a product line by Lego that provides the necessary tools for creating simple robots using Lego bricks. Mindstorms is built around a programmable microcomputer brick that can control up to 3 motors and read up to 4 sensors. The motors and sensors can be hooked up to the control unit using snap connectors, so no special skills are required to assemble a functional set of robot electronics. The programmable brick itself can be programmed using a graphical programming language, named NXT-G, which is bundled with the sets. Alternatively, the brick can be programmed using one of the many available third-party applications, which provide support for languages such as C++ [19] or Java [20]. Lego Technic style bricks are used to build the structural parts of the robot. Because of this, building blocks from other Lego sets can be easily incorporated, expanding the potential level of intricacy of the robots.

The ease of use, the size of the Mindstorms community and the familiarity of Lego bricks make this a popular choice as a platform for robotics in education, especially when working with younger children. The popularity of Mindstorms translates to a plethora of resources available for educators, such as the many books, robot contests, online communities and workshops built around the Mindstorms ecosystem.

One of the main disadvantages of Lego Mindstorms is the cost, at a price of over €330 for the educational base set [21], providing enough robot sets for an entire class can quickly become an expensive affaire. Larger schools can alleviate this by buying enough sets for one group and then pass them around between the different class groups, but this is not always possible, especially for small schools or organizations. Another problem we've encountered frequently is that the programmable brick is limited to a maximum of 3 motors and 4 sensors. While this is typically enough to build a multitude of different robots, sooner or later students want to add another motor or sensor to their robot, only to discover they've run into the physical limit of what Mindstorms robots can do. A third point of criticism is that it's generally much harder to integrate third-party robotics components within a Mindstorms robot. Third-party sensors that can be interfaced with the Mindstorms programmable brick, such as those sold by MindSensors [22], do exist, but they typically rely on specialized circuitry for compatibility. Similarly, incorporating non-Lego hardware components into your Mindstorms robot is not always easy and typically involves modifying or otherwise damaging the Lego components.

B. Vex

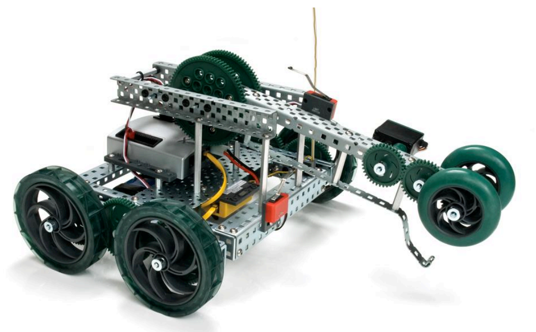


Fig. 2. Protobot Robot Kit [23].

The Vex Robotics system [24] is similar to Lego Mindstorms in that it is a platform that supplies all the necessary elements needed to build a functional robot: structural components, electronics, software and instructions. The main difference is that while Mindstorms tends to be more toy-like, suited for younger children, Vex instead opts for a more serious approach to robotics, targeting older students and adults. This is evident in the way the kit works; instead of plastic, perforated metal beams are used as structural elements and connections are made using nuts and screws, instead of snap and friction connections. Vex offers 2 different microcontroller options for use with its products, the PIC microcontroller and the Cortex microcontroller, both of which can be programmed using written code, as opposed to the graphical programming language used in Mindstorms. Vex allows for much choice in its product line, whereas Mindstorms aims to provide everything needed in one box, Vex allows for much more granular choice by splitting everything up in separate kits.

The Vex platform is more in line with "real" engineering practices than Lego Mindstorms and it offers a large degree of flexibility. This flexibility does make it more complex to use, meaning that this platform is better suited for older students and adults. At around €356 for a programmable starter kit [25], the cost of Vex seems similar to Mindstorms, however this not include programming software (which costs another €70) or the various expansion kits needed to complete a specific kind of robot. Mindstorms, in contrast, does include programming software and the components supplied in the base kit will typically be enough for many different types of robot projects. The Vex system is well documented, offering detailed instruction manuals, CAD models of the different parts, video tutorials and teacher materials. In our opinion, the main downsides of this platform are the high cost of the products and the degree of complexity, which makes it less suited for young children.

IV. Electronics

One of the core requirements for a functional robot is an electronics system that can read sensors, process information and control outputs. Many options of electronics are available when building a robot from scratch, making it possible to tune the electronics to the specific needs of your robot.

A. Arduino

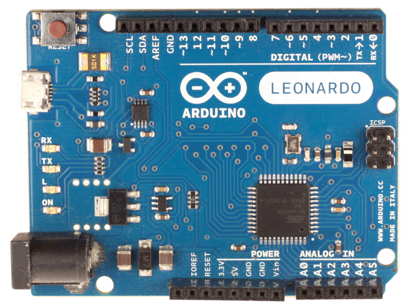


Fig. 3. Arduino Leonardo [26].

First developed at the Ivrea Institute for Interaction Design in 2005, the Arduino platform is the combination of a microcontroller board, a set of C++ libraries and an Integrated Development Environment (IDE) aimed at making microcontrollers accessible to artists, designers and hobbyists [27, 28]. The board itself is based around the AVR series of microcontrollers, made by Atmel. It provides all the necessary circuitry needed to make the microcontroller functional and breaks out the microcontroller's pins to easily accessible headers. On the software side, a multiplatform IDE bundles a code editor, a compiler, a linker and a programming utility into one preconfigured package. By preloading a boot loader onto the Arduino boards, it is possible to program the boards using a standard USB cable, without the need for an external hardware programmer. Writing code for Arduino boards is done in C++, but special libraries are provided which abstract the intricacies of programming microcontrollers into easy to use functions and classes. The combination of a low cost (€18 for an Arduino Leonardo board [29]) and the ease of use have made Arduino a very popular platform, especially among hobbyists.

The Arduino boards are not specifically designed for use in robotics, but this functionality can be added through the use of daughter boards, called shields. These shields can be attached on top of an Arduino board and provide the board with extra functionality, such as a display or circuitry to control DC motors directly. Many of the Arduino and Arduino compatible boards use the same physical pin layout; because of this the shield form factor has become a de facto standard. Many different shields exist, providing a plethora of possible functionality, made by either the Arduino organization or, more often, by third party vendors.

The availability of the large number of boards and shields, the ease of use, especially compared to other microcontrollers, and the low cost have all contributed to the creation of an Arduino ecosystem. Consequently, there is a large body of documentation and support available, in the form of books, tutorials and online communities. Arduino makes building a robot from scratch easier, in a low cost and flexible way. This low cost and flexibility does have an impact on the ease of use. While Arduino makes the use of microcontrollers easier, building robots using Arduino still requires a good working knowledge of both electronics and programming. For this reason, building robots with Arduino is significantly more complex than using an all-in-one solution, such as Mindstorms. Another problem that educators face when choosing Arduino is that the sheer amount of boards and shields can be overwhelming, so that the starting point is not always clear.

B. Dwengo

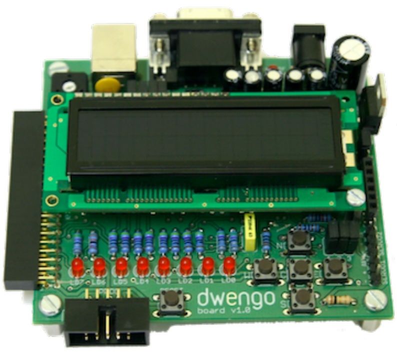


Fig. 4. Dwengo board [30].

The Dwengo project aims to provide an easy to use platform for getting started with microcontrollers, electronics and programming [31]. Originally started in 2009 at Ghent University as a microcontroller experimentation board for internal use, it was developed further when its potential benefits for education became clear. This board is built around the PIC series of microcontrollers, made by MicroChip. Unlike Arduino boards, which typically aim to provide a very low cost bare-bones board, the Dwengo board includes several features to facilitate building robots. The board includes a 16x2 character LCD, input buttons, 2 servo connectors and a 2 channel motor driver. In addition to the electronics board, the Dwengo project also provides a set of C libraries to facilitate programming, a set of tutorials which teach how to use board's features in a step-by-step manner and an experimental web application to program the board using a drag & drop interface. Like with the Arduino, building robots with Dwengo is definitely more complex than using a complete system such as Lego Mindstorms, but this in turn allows for more flexibility and gives the students better insight concerning how and why things work. While Dwengo has some large benefits over Arduino, such as the built in peripherals for making robots, Dwengo does not have the benefit the same large ecosystem.

C. Raspberry Pi

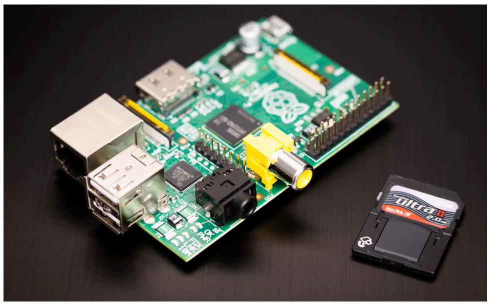


Fig. 5. Raspberry Pi [32].

The Raspberry Pi is a low cost single board computer, developed to promote the teaching of programming and computer science in education [33, 34]. The boards can be bought for $25 (model A) or $35 (model B) and provide the hardware required for a simple Linux system. In addition to 2 USB ports, an HDMI port, an SD card reader and an Ethernet port, the board also contains a pin header that gives access to low-level peripherals. The low cost combined with the access to these low-level peripherals make the board popular with hobbyists. One important thing to note is the difference between a microcontroller board, such as an Arduino, and a single-board computer, such as a Raspberry Pi. While a single-board computer is generally much more powerful, the overhead caused by the operating system (OS) makes it less suitable for applications that require precise timing, a task at which a dedicated microcontroller excels. On the other hand, the high-level nature of a Raspberry Pi makes it possible to program using high-level languages, to interface with more complex peripherals, such as webcams, and to connect to the internet.

The Raspberry Pi lacks many of the features that are required to build a robot, such as the ability to control DC motors. Luckily, this functionality can be added using daughter boards, much like Arduino shields. While the Raspberry Pi may not be able to provide the same level of real-time control as a dedicated microcontroller, it does offer many advantages compared to more traditional solutions. The board is designed to run Linux; this makes it possible to make a robot using more powerful programming languages. Another advantage is the ability to easily interface with USB devices, such as webcams and Wi-Fi dongles, which allows for advanced robots with internet connectivity and image recognition capabilities.

V. HARDWARE

Hobbyist robot makers tend to rely on a wide variety of techniques when it comes to building the physical embodiment of their robots. Some repurpose old toy components, some make their robot out of cardboard, glue and duct tape, some even build their own metal chassis using advanced CNC machines. A few projects aim to facilitate this process by providing a standardized building system.

A. MakeBlock

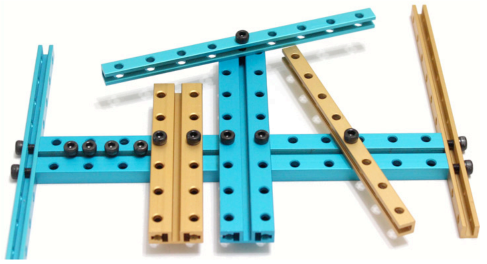


Fig. 6. MakeBlock [35].

MakeBlock [36] is a commercial building system specifically aimed at building robots. The MakeBlock system is built around several different types of aluminium beams which are joined together using standard machine screws. These screw connections can be made using either the beam's threaded slot, the grid of evenly spaced holes or the tapped holes at the end of a beam. In addition to these beams, several accessories are available, such as DC motor mounts, servo mounts, angle brackets and joining plates.

The MakeBlock system is a great way to make a very rigid robot and the threaded slot makes it easy to connect third-party components to the frame. The system uses the same hole spacing as Lego Technic bricks, further improving compatibility. In our opinion, the main downsides of MakeBlock are the relatively high costs and the limited availability.

B. OpenBeam

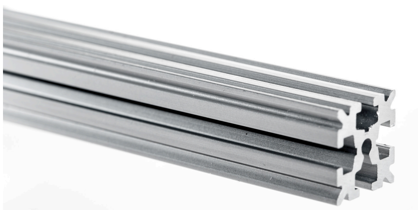


Fig. 7. OpenBeam [37].

OpenBeam [38] is a small scale version of the well known industrial T-slot aluminium profile systems. OpenBeam was conceived as an Open Hardware project and was realized though the Kickstarter crowdsourcing platform [39]. While regular T-slot profiles typically use their own proprietary nuts and bolts, OpenBeam was specifically designed to make use of standard M3 nuts and bolts, which are generally much cheaper and more readily available. These beams can be combined using angle brackets to build three-dimensional structures. Because OpenBeam uses standard hardware and because any arbitrary hole spacing can be used, it is easy to connect third-party components to the system. OpenBeam was not made specifically with robotics in mind, but because other components can be connected so easily, it does make it a viable way of building robots. Still, the OpenBeam system focuses on providing strong static connections, but offers much less in terms of building moving mechanisms.

C. BitBeam

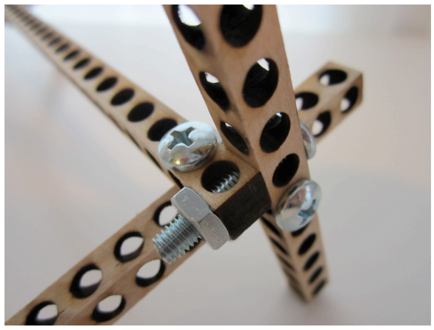


Fig. 8. BitBeam [40].

BitBeam [41] is a miniaturized version of GridBeam, a building system that uses 1.5 inch square beams with regularly spaced holes as a construction material to build large objects such as furniture [42, 43]. The specific geometry of the beams allows for a technique called a trijoint, shown in figure 8, in which 3 beams are joined in a corner using only 3 bolts and nuts. The BitBeam variant of the GridBeam system uses 8 mm square beams with holes spaced at 8 mm intervals, making it much more suited for building small scale robots. The 8 mm distance was not chosen arbitrarily, the hole spacing matches that of Lego Technic bricks, making integration with Lego bricks trivial.

The big advantage of BitBeam over other systems is its low cost and the fact that it can be made from a variety of materials. The beams shown in the examples are made by laser cutting holes in 5/16 inch square wooden beams, but compatible beams can also be made using a 3D printer or even by manually drilling holes using a drill press. In our experience, wooden beams are not as durable as similar components made from plastic or metal, especially because the beams are severely weakened by the 2 directions of holes. We have tried making BitBeams out of 8 mm acrylic, and while they are much more durable, we ran into the problem that longer beams started to warp significantly due to the heat caused by the laser cutting process.

VI. PROGRAMMING

Often, robots are programmed using a low-level, textual programming language, such as C. These textual languages can be quite daunting for people with no prior experience. Not only is it hard to translate human language concepts to algorithms a computer can understand, one has to take care that the syntax of the program is correct. Graphical programming languages can alleviate the latter problem, while also providing an interface that is more appealing to children.

A. Scratch

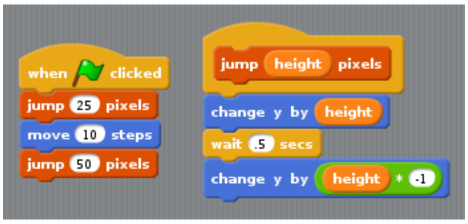


Fig. 9. Scratch [44].

Scratch, developed at MIT Media Lab in 2006, is a graphical programming language that aims to teach children the principles programming through the creation of simple games and interactive movies [45, 46]. Scratch features a display area, onto which different sprites can be placed, and a programming area, onto which puzzle-like programming blocks can be placed. These puzzle blocks can represent simple commands (e.g. "move 10 steps"), or more complicated control structures, such as loop statements (e.g. "repeat … until") or conditional statements (e.g. "if … then …"). The blocks can be snapped together to create a logical sequence of actions, akin to the sequence of statements one would find in traditional code. Because of the different shapes of the different types of puzzle-like blocks, they can only be combined in a way that makes sense, making syntax errors impossible.

While Scratch is very much oriented at computer-centered use, there are some options for making Scratch interact with the outside world, such as Enchanting [47] and PicoBoard [48]. Enchanting is a variant of the Scratch application that can compile Scratch programs into programs that can be run on a Lego Mindstorms NXT intelligent brick. PicoBoard is a board featuring a light sensor, a sound sensor, a button, a slider and 4 extra ports to which external sensors can be connected. While there are some options to make Scratch interact with the outside world, these options are limited and Scratch is better suited for computer use only. In our opinion, Scratch still provides a great way of learning the fundamentals of programming in a colorful, attractive environment.

B. Blockly

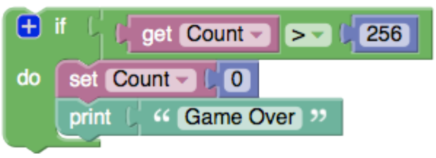


Fig. 10. Blockly [49].

Google Blockly [50], a programming language influenced by the aforementioned Scratch, is different from other graphical programming languages in that it's not intended for direct use. Instead, Blockly can be seen as a set of libraries that greatly facilitates the development of a Scratch-like programming language. Blockly is written in Javascript and is intended to run inside a web browser environment. Using the Blockly Application Programming Interface (API), one can easily define its own set of blocks in order to create a fully customized graphical programming language. One of the defining features of Blockly is that it can automatically translate a Blockly program to readable, written code. Language generators for Javascript and Python already exist, but custom generators can also be made using the API. Blockly cannot be used to program robots directly (and is not meant to do so), but it does provide a very convenient way to design a language that can be used for that purpose.

C. ArduBlock

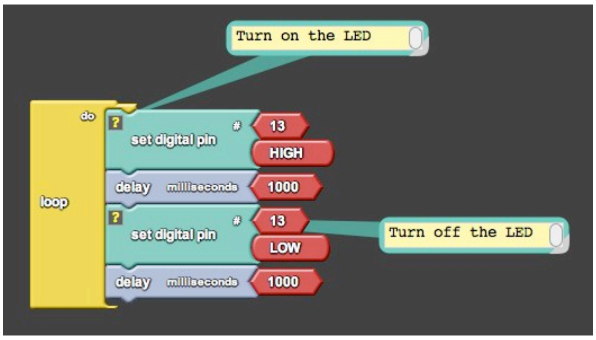


Fig. 11. Raspberry Pi [51].

ArduBlock [52] is a plugin application for the Arduino IDE. It provides an integrated tool that makes it possible to write Arduino programs using the same style of graphical blocks as Scratch and Blockly. In addition to blocks that are literal translations of the functions in the Arduino library, it also provides some predefined blocks for working with third-party Arduino components, such as a relay or a joystick. When programming an Arduino board using ArduBlock, the graphical program is translated to regular Arduino code, not unlike Blockly's language generators. This facilitates the transition between using graphical blocks for programming and using written C++ code, which is very helpful for novice programmers. While ArduBlocks is a great introductory tool for the Arduino platform, it is our opinion that there is still much room for improvement. It does not yet have the same level of attention to visual details as Blockly or Scratch, not all Arduino functions or features are available and some editing functionality (such as deleting blocks) works counter intuitively. Still, it is a great tool for use in education, especially when students have already been introduced to a similar language, such as Scratch.

VII. DIY AS A NEW WAY OF TEACHING

A DIY methodology promises to transform education from student observation and listening to active engagement through ingenious interactive hands-on lessons guided by instructors and augmented by examples and equipment that can be fabricated only when needed allowing for personalizable and customizable classes and individual learning within common frameworks. The feasibility of a robotics-enhanced problem-based curriculum depends on the access to versatile robotics tools and well-organized tutorials. We screened the available solutions and we carried out a series of interviews with educators to pin point the needs in the field when implementing a new curriculum. For this first round of informative interviews we focused on Flanders (Belgium) and a few primary and secondary school teachers in Italy and Switzerland.

One situation we have encountered frequently is that educators choose an all-in-one robotics platform because they do not know of any alternatives or because they cannot find a clear starting point for alternative platforms. This choice is often further motivated because their regional colleagues tend to use the same platform, so there is a certain form of a support network. In our experience, classes that use a complete robotics platform, such as Mindstorms, tend to outnumber classes that build their own robots from scratch by a large margin. And while robots built with an all-in-one kit may perform better, students that build their own robot either completely from scratch, or by combining elements from the different systems as described above, tend to gain a much deeper understanding of technology, engineering and creative problem solving. Class groups that use a complete robotics platform also tend to stay within that platform, adding a third-party or a home-made component does not happen frequently. Robots built using a complete platform tend to have more functionality and are usually easier to build, while building robots from scratch tends to require more experience but also give the students more insight.

We believe that the DIY techniques of the Maker Movement provide a good way to make building robots from scratch much easier, while also bridging the gap between all-in-one platforms and DIY robots. The Maker Movement makes Rapid Prototyping technologies, such as laser cutting, CNC milling and 3D printing much more accessible to the general public. Whereas the use of 3D printing used to be limited to large organizations and businesses, this movement has created comparable machines, such as the many types of RepRap machines, which can be made for under €1000 [53]. In addition to the option of building your own rapid prototyping machine, which many not always be financially or practically possible for schools, other options have also become available, such as the use of online rapid prototyping services or the use of machines at a local FabLab. FabLabs are publically accessible local workspaces that offer access to these rapid prototyping machines. These FabLabs, if they are locally available, can be a great benefit to robotics projects in schools [54]. They offer a suitable space to work in, access to machines to manufacture parts with, and a community of like-minded people who can share their knowledge and experience.

This DIY way of thinking has already gained some foothold within the context of small-scale robotics. Some of the projects we described above, such as Arduino, OpenBeam, BitBeam, Scratch, Blockly and ArduBlock are Open Source and/or Open Hardware. However, this phenomenon is not limited to these projects. Thingiverse [55], an online repository for design files of physical objects, lists many different types of DIY robot projects, ranging from very simple parts (e.g. a mounting plate to connect an Arduino to Lego bricks [56]) to small wheeled robots (e.g. MiniSkybot [57]) to very complex robots (e.g. humanoid robot InMoov [58]). Another great example of this DIY way of thinking being applied in education is Arvind Gupta’s Toys from Trash project [59, 60], which is a website that lists a plethora of small scientific experiments, all of which can be made with very cheap materials.

VIII. Conclusion

In this paper, we have presented a summary of products and projects that can be used as tools for enabling robotics projects in education. The categories we discussed are complete, all-in-one robotics platforms, electronics, hardware, and software. Generally speaking, there are 2 ways to build a robot: either by using a complete robotics platform, or by constructing a robot from scratch. Complete systems are easier to use, allow for quicker results and are better suited for young students. The downside is that they are generally more expensive and less flexible. Building a robot from scratch, in contrast, is much harder and is more suited for older students, but gives much better insight in the technology, is more flexible and can be much cheaper. In recent years, building a robot from scratch has become much easier due to numerous projects and products that implement certain aspects of a robot, such as hardware, software or electronics. These product and projects can often be linked to the recent DIY and Maker Movement trends. These trends are characterized by the use of CNC machines and the collaboration over the internet to create physical hardware projects. We believe that the DIY and Maker subculture can have a valuable impact on education, as it not only encourages young people’s interest in STEM-related fields, it also fosters creativity and technological fluency. All of these skills will undoubtedly be vital in the society of tomorrow.

References

[1] W. Sunthonkanokpong, "Future Global Visions of Engineering Education," *Procedia Engineering,* vol. 8, pp. 160-164, 2011.

[2] J. Perkins, "Education in process systems engineering: past, present and future," *Computers & chemical engineering,* vol. 26, pp. 283-293, 2002.

[3] M. Molzahn, "Chemical Engineering Education in Europe," *Chemical Engineering Research and Design,* vol. 82, pp. 1525-1532, 2008.

[4] P. B. Campbell, E. Jolly, L. Hoey, and L. K. Perlman, "Upping the Numbers: Using Research-Based Decision Making To Increase Diversity in the Quantitative Disciplines," 2002.

[5] I. Ríos, A. Cazorla, J. M. Díaz-Puente, and J. L. Yagüe, "Project–based learning in engineering higher education: two decades of teaching competences in real environments," *Procedia-Social and Behavioral Sciences,* vol. 2, pp. 1368-1378, 2010.

[6] O. Ates and A. Eryilmaz, "Factors affecting performance of tutors during problem-based learning implementations," *Procedia-Social and Behavioral Sciences,* vol. 2, pp. 2325-2329, 2010.

[7] S. Papert, *Mindstorms: Children, computers, and powerful ideas*: Basic Books, Inc., 1980.

[8] J. Johnson, "Children, robotics, and education," *Artificial Life and Robotics,* vol. 7, pp. 16-21, 2003.

[9] S. Bell, "Project-based learning for the 21st century: Skills for the future," *The Clearing House,* vol. 83, pp. 39-43, 2010.

[10] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in github: transparency and collaboration in an open software repository," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 2012, pp. 1277-1286.

[11] D. Dougherty, "The Maker Movement," *Innovations: Technology, Governance, Globalization,* vol. 7, pp. 11-14, 2012.

[12] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, *et al.*, "RepRap–the replicating rapid prototyper," *Robotica,* vol. 29, pp. 177-191, 2011.

[13] E. Ford. (2013). *ShapeOko*. Available: http://www.shapeoko.com/

[14] M. J. Mataric, N. Koenig, and D. Feil-Seifer, "Materials for enabling hands-on robotics and STEM education," 2007.

[15] S. Hartmann, H. Wiesner, and A. Wiesner-Steiner, "Robotics and gender: The use of robotics for the empowerment of girls in the classroom," in *Gender Designs IT*, ed: Springer, 2007, pp. 175-188.

[16] Lego, "The NXT Technology," http://cache.lego.com/upload/contentTemplating/Mindstorms2History/images/picB2F61C4781F076001DA7E229CE695D81.jpg, Ed., ed, 2013.

[17] Lego. (2013). *LEGO.com MINDSTORMS: Home*. Available: http://mindstorms.lego.com

[18] G. J. Ferrer, "Using Lego Mindstorms NXT in the classroom," *Journal of Computing Sciences in Colleges,* vol. 23, pp. 153-153, 2008.

[19] T. Chikamasa. (2013). *nxtOSEK: Index*. Available: http://lejos-osek.sourceforge.net/

[20] J. Solórzano. (2013). *LeJOS, Java for Lego Mindstorms / NXJ*. Available: http://lejos.sourceforge.net/nxj.php

[21] Rato. (2013). *LEGO Education*. Available: http://www.mindstormsnxt.be/

[22] Mindsensors. (2013). *Mindsensors*. Available: http://www.mindsensors.com/

[23] VEX, "VEX Protobot Robot Kit," http://www.vexrobotics.com/media/catalog/product/cache/103/image/5e06319eda06f020e43594a9c230972d/P/r/Protobot.jpg, Ed., ed, 2013.

[24] InnovationFirstInternational. (2013). *VEX - VEX Robotics*. Available: http://www.vexrobotics.com/vex

[25] InnovationFirstInternational. (2013). *Programming Control Starter Kit - VEX Robotics*. Available: http://www.vexrobotics.com/276-2750.html

[26] Arduino, "ArduinoBoardLeonardo," http://arduino.cc/en/uploads/Main/ArduinoLeonardoFront_2.jpg, Ed., ed, 2013.

[27] Arduino. (2013). *Arduino*. Available: http://arduino.cc

[28] D. Mellis, M. Banzi, D. Cuartielles, and T. Igoe, "Arduino: An open electronic prototyping platform," in *Proc. CHI* vol. 2007, ed, 2007.

[29] Arduino. (2013). *Arduino Leonardo (+headers) [A000057] - €18.00 : Arduino Store - community and electronics*. Available: http://store.arduino.cc/eu/index.php?main_page=product_info&cPath=11_12&products_id=226

[30] Dwengo, "Overview functionality of the Dwengo board," http://www.dwengo.org/sites/default/files/overzicht_dwengo-board_display_0.png, Ed., ed, 2013.

[31] Dwengo. (2013). *Dwengo | Gets you started with microcontrollers*. Available: http://www.dwengo.org/

[32] RaspberryPi, "Raspberry Pi Model B," http://www.raspberrypi.org/wp-content/uploads/2011/07/7513051848_9a6ef2feb8_o-1024x682.jpg, Ed., ed, 2013.

[33] RaspberryPi. (2013). *Raspberry Pi | An ARM GNU/Linux box for $25. Take a byte!* Available: http://www.raspberrypi.org/

[34] G. Mitchell, "The Raspberry Pi single-board computer will revolutionise computer science teaching [For & Against]," *Engineering & Technology,* vol. 7, pp. 26-26, 2012.

[35] MakeBlock, "connection," http://www.makeblock.cc/wp-content/uploads/2012/11/connection1.jpg, Ed., ed, 2013.

[36] ShenzhenHuluRoboticTechnology. (2012). *Makeblock - Aluminum Robot Kit*. Available: http://makeblock.cc/

[37] T. Tam, "OpenBeam 1515 x 1000mm, clear anodized," http://www.openbeamusa.com/images/products/229.jpg, Ed., ed, 2013.

[38] T. Tam. (2013). *OpenBeamUSA.com - Home*. Available: http://www.openbeamusa.com/

[39] T. Tam. (2012). *OpenBeam - An open source miniature construction system by Terence Tam — Kickstarter*. Available: http://www.kickstarter.com/projects/ttstam/openbeam-an-open-source-miniature-construction-sys

[40] J. Huggins, "bitPad - Tri-joint," http://www.flickr.com/photos/68386867@N05/6384780527/, Ed., ed, 2011.

[41] J. Huggin. (2013). *Bitbeam | The Open Source Building Toy*. Available: http://bitbeam.org/

[42] P. Jergenson. (2013). *Gridbeam - open source building system for a post carbon future*. Available: http://www.gridbeam.com/

[43] P. Jergenson, R. Jergenson, and W. Keppel, *How to Build With Grid Beam*: New Society Publishers, 2008.

[44] Scratch, "Scratch 2.0: Create your own block," http://1.bp.blogspot.com/-s0Z7HmigCkY/TY6VV5c1IkI/AAAAAAAAAB8/SqyulusFK1o/s1600/jump-pixels-height.png, Ed., ed, 2011.
[45] MIT. (2012, September 10). *Scratch | Home | imagine, program, share*. Available: http://scratch.mit.edu/
[46] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan*, et al.*, "Scratch: programming for all," *Communications of the ACM,* vol. 52, pp. 60-67, 2009.
[47] C. Blackmore. (2012). *Enchanting*. Available: http://enchanting.robotclub.ab.ca/tiki-index.php
[48] PlayfulInvention. (2013). *PicoBoard - Sensor Board that works with MIT's Scratch*. Available: http://www.picocricket.com/picoboard.html
[49] N. Fraser, "Blockly Sample," http://blockly.googlecode.com/svn/wiki/sample.png, Ed., ed, 2013.
[50] N. Fraser. (2013). *blockly - A visual programming editor*. Available: http://code.google.com/p/blockly/
[51] D. Li, "Getting Started with Arduino in ArduBlock: Example 01," http://blog.ardublock.com/wp-content/uploads/2012/02/untitled1.jpg, Ed., ed, 2012.
[52] D. Li. (2012). *ArduBlock*. Available: ardublock.com
[53] A. Bowyer. (2012). *RepRap - RepRap Wiki*. Available: http://reprap.org/wiki/Main_Page
[54] P. Blikstein, "Digital Fabrication and 'Making'in Education: The Democratization of Invention."
[55] MakerbotIndustries. (2012). *Thingiverse - Digital Designs for Physical Objects*. Available: http://www.thingiverse.com/
[56] badBrick. (2013). *Brick Mount Base Plate for Arduino UNO R3*. Available: http://www.thingiverse.com/thing:62139
[57] J. Gonzalez-Gomez. (2011). *MiniSkybot Robot V1.0*. Available: http://www.thingiverse.com/thing:7989
[58] G. Langevin. (2013). *Head for Robot InMoov*. Available: http://www.thingiverse.com/thing:67676
[59] A. Gupta. (2013). *Toys from Trash*. Available: http://www.arvindguptatoys.com/toys.html
[60] A. Gupta, "Aha! activities," *Bhopal: Eklavya Publication,* 2007.

DIDACTIC AUTOMATED ASSEMBLY SYSTEM

Witold Pawłowski
Institute of Machine Tools and Production Engineering,
Lodz University of Technology, Poland

Michał Krępski
Institute of Machine Tools and Production Engineering,
Lodz University of Technology, Poland

Sławomir Gabara
SMC Industrial Automation Polska Sp. z o.o.

***Abstract*-The article presents project and didactic assumptions as well as the construction and functioning of the automated system for the assembly of component containing a case, a shaft, a bearing and a cap. In the construction of the stand there have been applied pneumatic drives by SMC company, PLC controller and the PROFIBUS communication. In order to simulate the anomalous situations occurring naturally, the switch box has been used. The article focuses on the significance and role of the modern solutions within the didactic process.**

Keywords-pneumatic drive and control; valve manifold; PLC; didactic process; assembly system

I. INTRODUCTION

Pneumatic parts have been mostly used in the construction of automated assembly systems. They are especially applicable for transportation and manipulating the details, their positioning, combining them in units as well as for assessing their proper shape, size or location [2], [3], [4]. Combining the pneumatic drives units with electronic PLC controllers is an effective method of automation of the process of assembly. Within the didactic process of a technical university with a faculty of mechanical engineering, it is particularly vital to run training on the use of modern devices, drives and controllers, which the future graduates will be able to use and apply in industrial settings. Therefore, didactic stands should contain elements of construction solutions used in industry. The complexity of these stands should be adjusted to the specific character of the didactic process. However, the procedures while designing and operating such a stand must produce useful operation stereotypes applied in the industrial practice. This is the prerequisite of the contemporary, modern didactic process associated with mastering engineering skills. It is also crucial that the already trained stereotypical way to proceed does not trigger creativity or activity in the sphere of prospect modifications of the existing state of technology. Hence the role of the modern didactic stands whose examples are described as follows in this article.

Characteristic features of the modern didactic systems include:

- the appliance of up-to-date and widely used in industrial practice concerning drives and controllers,
- the opportunity of combining elements according to the pre-planned operating cycle by an individual,
- the opportunity of simulating the most probable failures (uncommon operation of the stand).

These conditions have provoked further designing and constructing of the didactic assembly stand for components with the use of pneumatic drives, PLC controllers and PROFIBUS network communication between the valve manifolds and the PLC controller [1].

II. CONSTRUCTION AND FUNCTIONING OF THE ASSEMBLY SYSTEM

Fig.1 shows an exemplary assembly system containing four details: a case, a shaft, a bearing and a cap. Fig.2 presents a scheme of the pneumatic system unit which has been constructed out of the elements provided by the SMC company [5], [6].

The stand comprises the following operating units:

- case assembly,
- control and transport case,
- bearing assembly,
- control and assembly of the shaft,
- cap assembly

and other units:

- air preparation kit,
- PLC controller,
- valve manifold,
- control panel unit and switch box (of simulated failures).

The basic cycle of the work of the pneumatic stand comprises 27 steps. In case of fault detection of the malfunctioning case (the lack of a bearing hole or a too shallow hole) there are four additional steps, after which the cycle repeats from step 1 again.

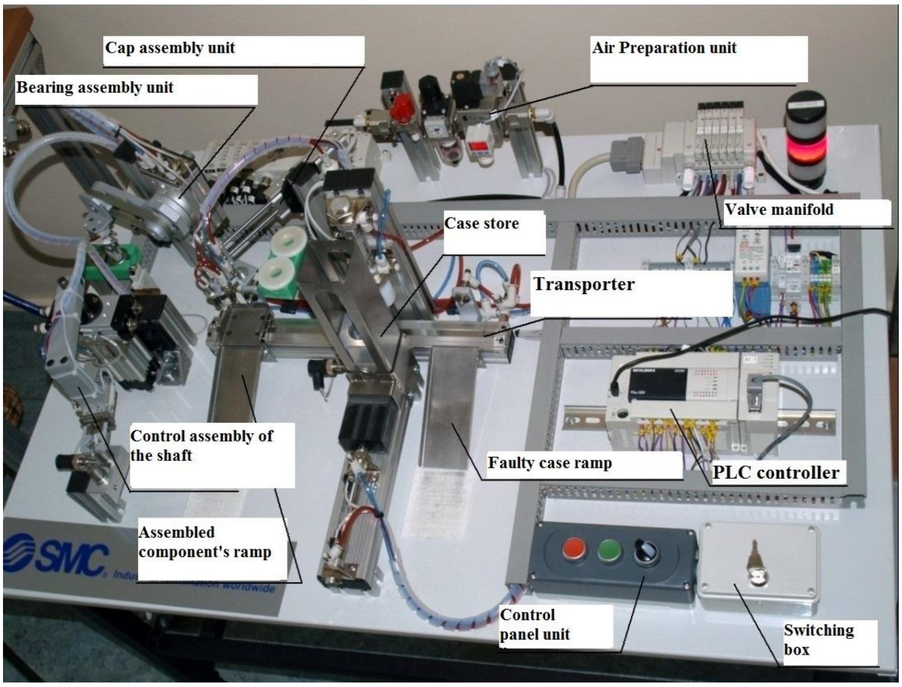


Fig. 1. The view of the assembly system

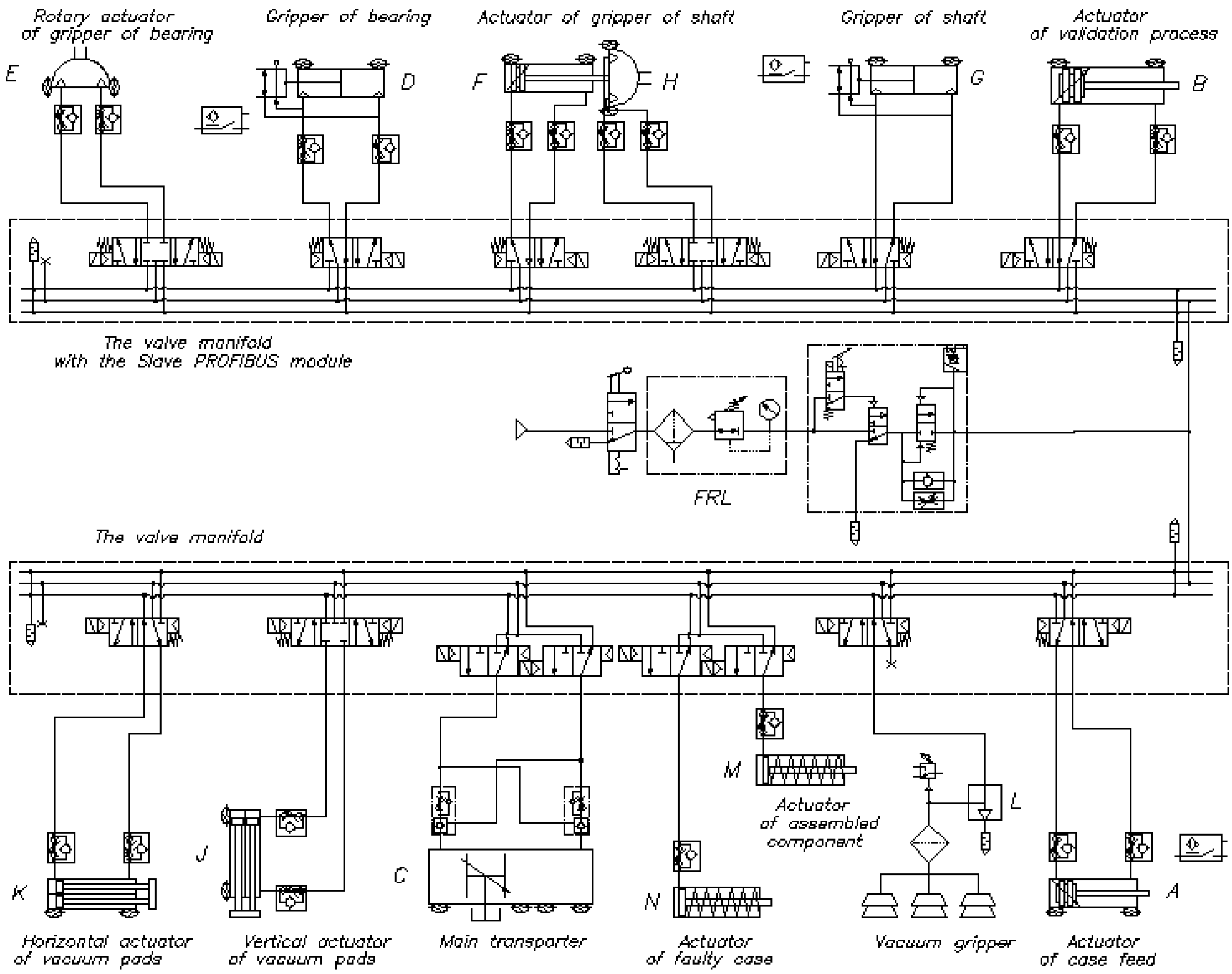


Fig. 2. The scheme of the pneumatic assembly system

A. The case assembly unit

The main drive element is the double-acting rod actuator A (Fig.2) which is equipped with two reed switches of the piston location.

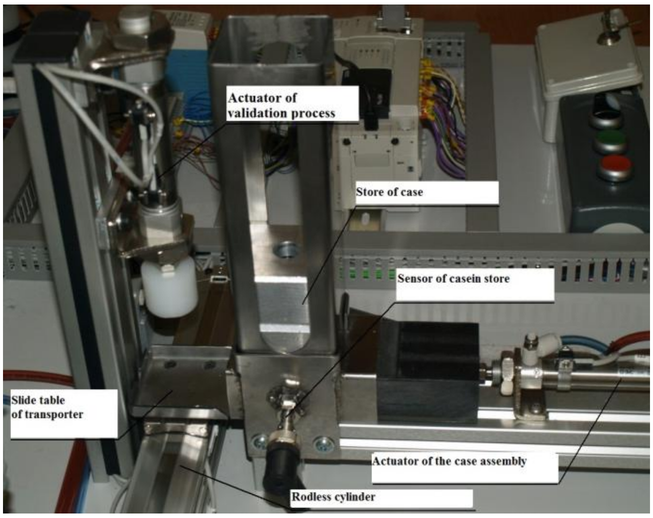


Fig. 3. The case assembly unit

An inductive sensor has been attached to the case store its role being to confirm the very existence of case in the store (Fig.3).

The cycle of work of the unit is as follows:

- actuator A of the case assembly pushes the detail from the store onto the slide table mounted on the transporter (mechanically jointed rodless cylinder – C – Fig.2),
- the retreat of the cylinder A causes shifting in the case store one step forward.

B. The control and transport case unit

The validation process of the location of the case is run by introducing a test through the assembly hole in the case (Fig.4). The test is performed by the double-acting actuator (Fig.2) which is equipped with two reed switches of the piston location. The lack of the signal of the bottom switch in a given amount of time, after the slide table has already been loaded with the case, means a faulty (upturned with the assembly hole down) location of the case.

Fig. 4. Validation process of the case location

After the assembly has been completed the unit is thrust further back from the slide table by the single-acting actuator M.

C. Feed and assembly bearing unit

The air gripper D (Fig.2) equipped with fingers for gripping the bearing by the inside ring transmits the bearing from the bearing magazine into the case (Fig.5).

The main drive element is the rotary actuator E (Fig.2) with the angle of the rotation 180°, equipped with a rolling bearing table together with hydraulic shock absorbers. Through the hole in the table there runs a stationary axis on which there is a wheel (also stationary) of the toothed belt transmission mounted. The rotary actuator table is mounted with an arm with the other wheel. The ratio of the transmission is 1:1, which assures the constant position of the gripper along the vertical axis of the bearing and the hole of the case.

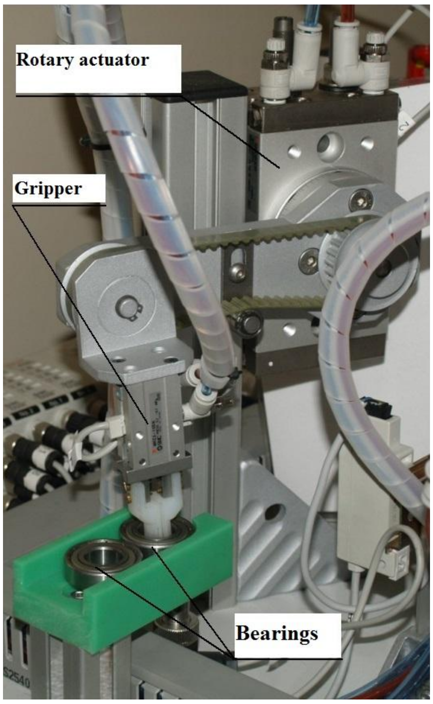


Fig. 5. Feed and assembly bearing unit

The cycle of the unit work is as follows:

- the gripper takes the bearing (Fig.5),
- the rotary actuator turns the arm with the gripper with the bearing over the case,
- the gripper places the bearing into the case.

D. *Control and assembly shaft unit*

The drive element of the unit consists of both linear and rotary actuator (Fig.6) which is made of a linear rod actuator F and a rotary cylinder H (rack and pinion style). At the end of the hexagonal rod of the rotary actuator there is an arm with a gripper G mounted. Once the rod has been taken from the magazine, the gripper raises up and rotates a quarter of a turn (90°) taking the position over the case. Next, it goes down and mounts in the bearing. The gripper is equipped with a special trimmer auto switch which validates the diameter of the shaft pivot. An acute sensitivity of the trimmer allows to detect a difference less than 1 mm from the assumed proper shaft pivot. The reason why the trimmer is used is to avoid the situation in which a manipulator tries to force the faulty mounting of a shaft larger than typical. The validation during the assembly stage is quick and allows to secure the drive elements and mounted parts from damage. The cycle of work of the control and assembly shaft unit comprises the following stages:

- the linear actuator lowers the gripper onto the shaft level,
- the shaft gripper closes fingers and if the diameter of the detail is correct the process is continued (if the diameter is too small it aborts the process and waits for the signal from the operator to start again),
- linear actuator raises the arm with the shaft up,
- the rotary actuator turns the arm with the shaft over the mounted unit,
- the linear actuator lowers the arm with the shaft gripper,
- the gripper opens fingers and places the shaft into the hole of the bearing,
- the manipulator comes back to the starting point, e.g. the linear actuator raises the gripper up, the rotary actuator turns the arm over the shaft feed.

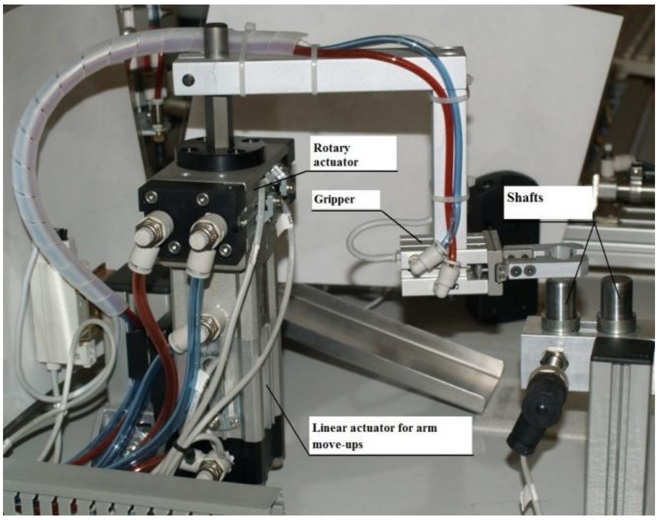


Fig. 6. Control and assembly shaft unit

E. *Cap assembly unit*

The drive of the cap assembly unit comprises two double-piston actuators equipped with reed auto switches (Fig.7).

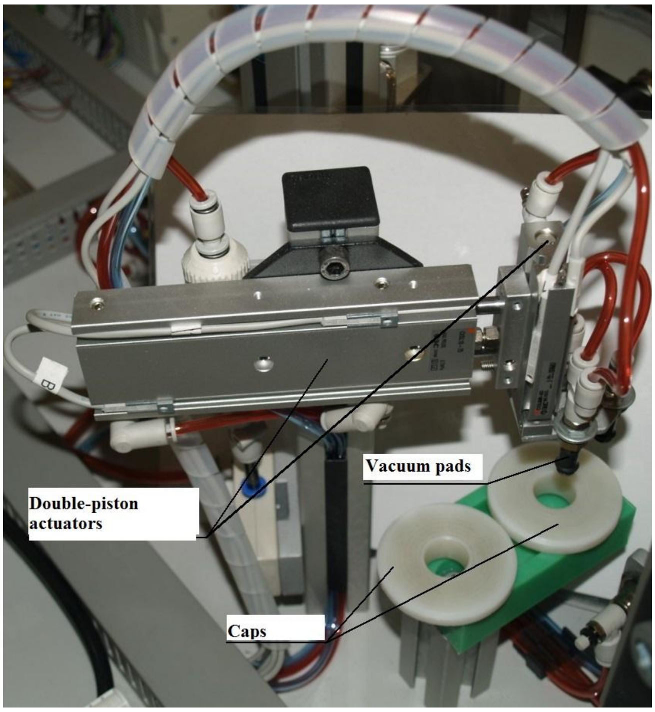


Fig. 7. Cap assembly unit

The rod endings of the other actuator are mounted with a unit of three bellows vacuum pads for moving the cap. Apart from the vacuum pads in the vacuum system there is a filter, a vacuum switch and an ejector. The validation of the cap gripping is run with the use of a vacuum switch which doesn't generate a controlling signal if level of vacuum is too low.

The cycle of the unit is as follows:

- the vertical linear actuator moves downwards,
- vacuum pads grip the cap,
- the vertical linear actuator moves upwards,
- the horizontal linear actuator moves ahead and places the cap over the case,
- the vertical linear actuator moves downwards with the cap,
- the vacuum pads loosen the cap,
- the vertical linear actuator moves upwards,
- the horizontal linear actuator moves back.

The cap unit returns to the starting position and thus the process is completed. The already assembled component is thrust from the slide table onto the ramp by a single-acting actuator, and the slide table comes back to the starting position in front of the case magazine.

F. *Air preparation equipment unit*

A crucial element of the system is the air preparation unit (Fig.8) including the following parts by SMC company:

- a 3-port hand VHS valve functioning as a shutoff valve with an unauthorized start-ups protection,
- an AW filter regulator with a manual exhaust,
- a soft start-up EAV valve equipped with a pressure sensor ISE30A.

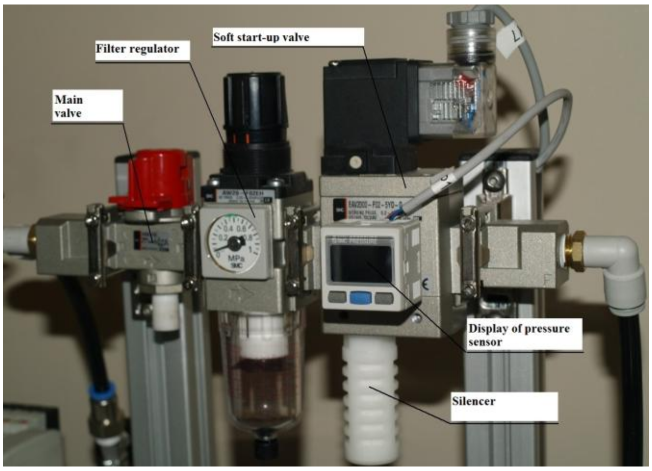


Fig. 8. Air preparation equipment unit

G. *PLC controller*

In the controlling system of the assembly stand the PLC FX3U-32M controller by MITSUBISHI has been implemented (Fig.9).

The controller gives the opportunity to connect 18 input signals and 18 output signals. In the already existing system there are 34 input signals and 26 output signals altogether. The problem of the lack of a sufficient number of inputs and outputs in the controller has been solved through the development of the PROFIBUS industrial network master module which allows up to 126 SLAVE devices being connected, each of them having 32 inputs and 32 outputs within the network span up to 1,000 m. Applying the industrial network module elasticizes the controller system giving the opportunity of connecting other devices alongside its development.

Fig. 9. PLC FX3U-32M controller by MITSUBISHI

H. *The valve manifolds*

The controller is connected to two valve manifolds. One of the manifolds is directly connected to the controller (Fig.10), whereas the other one by means of the PROFIBUS communication network module (Fig.11).

Fig. 10. The valve manifold connected to the controller

Fig. 11. The valve manifold connected to the Slave PROFIBUS module

In the next stage of the stand development, automated shafts, bearings and caps feeds are planned to be installed. In consequence, it will enlarge the number of input signals from reed switches and output signals to valves. In order to control these signals another Slave PROFIBUS module should be installed in the particular stand without any necessary changes to be done to the PLC controller.

I. Control panel and switch box

The control panel (Fig.12) has been equipped with START and STOP pushbuttons as well as with the switch of the type of work.

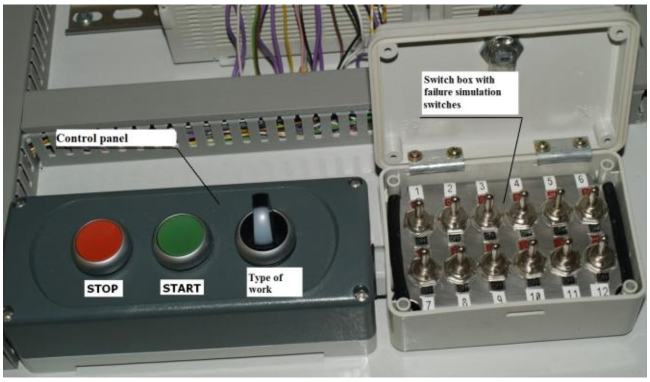


Fig. 12. Control panel and switch box

Next to the control panel, there is a switch box. The inside contents of the box is unavailable to the students. The instructor/trainer can choose to introduce the failure simulation (from 1 to 12) in the functioning of the system. Some of the simulated failures are due to the lack of input signals, others are due to cut-offs in the output signals of the PLC controller. The aim of the students is to diagnose the factual state of the functioning of the system, namely the failure.

III. Summary

The designed and constructed laboratory assembly system with the use of the PLC controller and the PROFIBUS communication network is applied within the didactic process at the Faculty of Mechanical Engineering of the Lodz University of Technology. This is a stand allowing to run trainings on designing of the cycle of devices, programming of the PLC controllers with the use of the network communication between the valve manifold and the controller together with diagnosing the functioning along with the failure detection. It is one of the most advanced didactic stands of the Pneumatic Drive and Control Laboratory and is equipped with the contemporary drive and control technology used in industrial systems.

Drive and control solutions applied in the stand can be used for presenting and analyzing basic kinematic systems of manipulators as well as their mechanical construction and control. Students can also observe, analyze and practice programming of the PLC along with Profibus industrial network in order to control the complex operation system of mechanical assembly. Additionally, the instructor can simulate failures of the system. The activity of detecting the failures and their reasons induces and stimulates the analytical attitude of the students and verifies the practical ability of the students to solve actual problems of the production or assembly lines in reality. To solve the problems properly the thorough knowledge and well-trained abilities to analyze the whole system (electric, electronic, pneumatic and control subsystems) are required. The number of tasks on different levels of difficulty for students to be solved is almost limitless. Due to the limited place around the stand the most effective students group should not exceed 5 persons.

The presented system constitutes the first stage of its construction. At the moment, it is undergoing the second stage of its development in which stores of bearings, shafts and caps feeds are being created. After they have been mounted and connected to the pneumatic system and PLC controller system, the full automation of the stand will be achieved without the necessity of hand feeding of the details.

References

[1] D. Stawiarski, „Urządzenia pneumatyczne w obrabiarkach i przyrządach," WNT, Warszawa, 1975.

[2] W. Szenajch, „Przyrządy i uchwyty pneumatyczne," WNT – Warszawa, 1975.

[3] W. Szenajch, „Pneumatyka i hydraulika maszyn technologicznych," Skrypt Politechniki Warszawskiej, Warszawa, 1983.

[4] S. Gabara, „Projekt automatyzacji procesu montażowego wyrobu zbudowanego z czterech podzespołów," the work supervised by M. Krępski, Archiwum IOiTBM PŁ, 2011.

[5] SMC, „Podręcznik firmy SMC: Sprężone powietrze i jego zastosowanie," 2011.

[6] SMC, „Katalog elementów pneumatycznych firmy SMC," 2012.

CVARC: an educational project for a gentle introduction to autonomous robots' control

Pavel Abduramanov, Maxim Kropotov, Anton Ryabykh and Yuri Okulovsky
Institute for Mathematics and Computer Sciences
Ural Federal University, Yekaterinburg, Russia, Turgeneva 4,
Email: yuri.okulovsky@gmail.com

***Abstract*—In this paper, we present CVARC – an educational initiative for gentle introduction of computers scientists in robotics field. CVARC is a system for conducting online competitions on virtual robots' control. It represents robotics in a way that is friendly for computer scientists and helps them to have a preliminary experience in robotics before joining the real-robotics projects. We hope that CVARC will be able to bring more young specialists on control algorithms, computer vision and other computer sciences areas into robotics, and maybe therefore to boost the robotics in general. We describe the architecture of CVARC, the case study of competitions that were conducted in our University, and the upcoming activities we are currently planning.**

INTRODUCTION

Historically, the robotics originated at the turn of mechanical and electrical engineering. To build a robot meant to create a mechanical system, animate it with motors and electronics circuits, and then add a simple deterministic behavior by programming some commands in the controller. With the robotics evolving, more and more intelligent and interactive algorithms are required. Nowadays control systems are so complex that the creation of almost any robot requires a third specialist — the software engineer, a specialist in development, debugging and testing of the software.

The role of the software developers in some cases even ceased to be an auxiliary one. For example, the current progress of mechanical and electrical engineering enables the industrial production of flying machines, like AR.Drone 2.0 Parrot [1]. Such machines could observe the streets and maintain security, or deliver the urgent mail. But in order to guarantee the robustness, safety and efficiency of such services, very complex control system should be developed — so complex, that today it is more fruitful to use manned or remotely controlled machines.

In order to overcome this problem, we should educate more software engineers, who are specialized in the robotics' tasks. The situation resembles another epoque of software engineering: when SAP ERP (a software for planning the resources of enterprises) were introduced, it created a demand for software engineers who can program it and understand economics and management. The same applies here: the progress of robotics creates standard platforms that can be used in many applications, so now we need software developers who would write the corresponding algorithms. And, like SAP developers need to know about economics, the robotics developers should know a lot of specific knowledge.

However, in Russia, the software engineering (SE) students are often alienated from robotics. Due to the history of computer sciences and software engineering, SE departments are usually administratively divided from electrical or mechanical engineering departments. SE students have troubles exercising in robotics by themselves, due to several reasons. First, building robots is expensive and software engineers often don't have necessary funds. Second, it requires lots of equipment, which is way too uncommon for SE departments. Software engineers also lack many vital skills in mechanics and electrical engineering. Finally, in real robotics very basic problems, like precise driving along a straight line or building a reliable robot that won't unexpectedly shut down, are not at all trivial, and SE students often just get stuck on them. They cannot advance to "really interesting" problems, and because of that are soon demotivated. SE students could join their colleagues from other departments and help them to write the software for the robots that are already built. Unfortunately, such perfect division of labour is often hard to achieve. The first problem is that these colleagues often expect from SE students some competence in robotics' control, which in turn requires the experience with a real robot, and so the vicious circle closes. The second is that "hardware" engineers are usually very sceptical about using technical vision, intelligent strategy planning, or other complex algorithms. This skepticism is partially justified, because SE students do not know how complex it is, to implement such algorithms efficiently and robustly, but again - where could they learn it without trying? However, in some cases the complexity of such tasks is exaggerated by "hardware" specialists, because the tasks are extremely hard *for them*, due to the poor skills in the programming languages, or to the inability to find and integrate the complete third-side solutions.

So, there is a problem of how to introduce robotics to software engineers. Such introduction will not only allow them studying and then working in an interesting and perspective field. It will also show them a various applications of things they are learning in classes. Robotics demands a profound knowledge in mathematics, including the areas which are usually considered "abstract" and "theoretical" by SE students. It improves the motivation to study such areas. Moreover, projects in robotics control are usually quite large and complex, involve several participants, and therefore improve the skills of project and team management. The bugs in control systems usually yield very funny situations, and therefore it is more enjoyable to implement testing techniques, like unit testing. Therefore, the robotics control project generally fit the SE curriculum.

In this article, we describe our initiative to introduce software engineers to robotics, which took a form of online competitions. Many SE students willingly participate in such extracurricular activities: there exist ACM [2] to develop skills in extreme programming and knowledge in complex algorithms, or competitions on computer security [3]. There are, of course, lots of virtual robotics competitions: virtual league of soccer for Aldebaran Nao [4], control of flying machine [6], various competitions on technical vision [5]. However, most of them require a very high competence from students, which is hard to achieve on a first or second year. But exactly these years are vital to help students to choose their future specialization. That is why we have decided to create a virtual robotics competitions with simpler rules, which emulates indoor competitions for mobile robots, like Eurobot [7].

We understand that virtual competitions is not a full-fledged alternative to real robotics, and we do not pursue the aim to create one. We want the software engineers to understand the common problems they probably would face in the robotics, and to learn the common means to solve them. Also, we want them to understand what can and cannot be achieved, because when SE student hears about robotics, he or she often starts thinking about a nearly human-like artificial intelligence, and is then either scared off, or drown in fruitless attempts.

The aim of such competitions dictates some differences between them and real robotics competitions. In virtual competitions, we may not consider the problems of building real robot, designing its actuators, making it go precisely or at least predictable. Therefore, we may spend more time on technical vision, strategy planning, competitive behavior and other "high-level" problems. Sometimes it is considered as a shift of attention from "the real" robotics problem. It our opinion, it just widens the horizon of what is possible in robotics, and so, when SE students join their colleagues in real robotics, they will be able to create much more complex control system, and therefore boost the overall level of robotics.

We have created a software for carrying out such competitions, CVARC — Competitions on Virtual Autonomous Robots' Control. The most important features of CVARC are as follows. First, almost no starting knowledge is required for the participant to join the competitions. He or she has to know only of how to open a TCP/IP connection and send a string though it. Any programming language can be used, there is no restriction on the operating systems, and so on. The second feature is the extreme easiness to create a new competitions: we paid special attention to make this process as simple as possible. To create a competition, one should download the framework, create a code on any .NET language with the locations of objects, and count score. After that, an Internet server for competitions, as well as the downloadable tutorial for trying the rules with manually controlled robot, will be created.

It is also important to stress, what CVARC *is not*. It is not the physical or graphical engine, like Unity [8]. We use existed solutions to perform physics interactions and draw scenes, though we wrap these solutions in order to gain a more simple access to them. We decided not to use the engines that are developed for computer 3D games, because they have lots of features that are useless in our case, and are enormously complex. CVARC is also not a robot simulator, like [9] or [10]. We do not emulate low-level data from controllers or devices, and do not try to offer a possibility to build various robotics from parts. Instead, we concentrate on things that are important for competitions: defining rules, building bots to provide a controllable opposition to participants, and so on.

CVARC is implemented in C# and .NET Framework. Therefore, only .NET languages can be used for the creation of new competitions. CVARC also requires Microsoft Windows to run the server, because it uses DirectX.

The main concepts of CVARC are presented in the section 1. Section 2 explains how to create your own competitions. Section 3 presents a case-study of the virtual robotics competitions, which we held in the Ural Federal University. In Section 4, we describe the upcoming online competitions, which are conducted gradually and therefore are a simple and enjoyable way to get started in the autonomous robots control.

I. Main concepts

A. Bodies and their architecture

The world of a competition consists of various bodies: the robots, the obstacles, the objects to handle, and so on. To construct the world, we have to define these bodies, their properties and locations; to count score or emulate measurements from sensors, we should analyze the current state of bodies; to perform the actions like gripping and releasing we need to change the state. Bodies are therefore fundamental, and it is especially important to implement them so that they are easy to understand and to use.

Our implementation is based on model-view-controller pattern [12], which was designed as a pattern for business applications, but fits surprisingly well to our causes. The model is a set of logical properties of our world. Bodies' locations, shapes, colors, etc. are defined in model. The controller is an entity that changes model in an appropriate way, obeying the business logic. In our case, the business logic is the laws of physics, and therefore the controller is a physical engine. A view is an entity that presents the model to the user, and in our case, the view is a graphics engine.

B. Model

The model is an hierarchical structure of bodies. A model of a body is an object of one of the `Body` descendant classes: `Box`, `Ball` or `Cylinder`. The `Body` class defines properties like `Shape`, `Location`, `Density`, `Size` and many others.

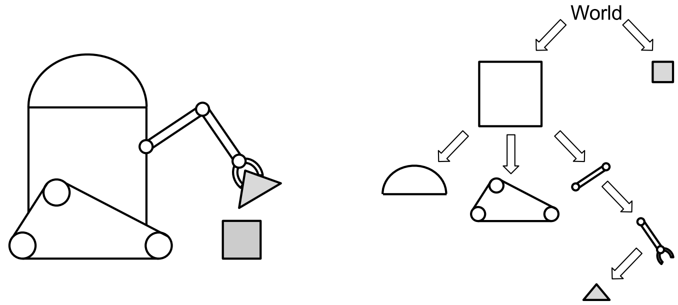


Fig. 1. The decompositions of the scene into the hierarchy

Additionally, the inherited classes `Box` or `Ball` adds shape-specific properties: e.g., six properties to define the color or textures of the sides for a box. All these properties are the ordinal C# properties.

The second fundamental class is `BodyCollection`, which is a collection of bodies. Any `Body` object contains such collection, which lists the bodies, attached to their owner. The attachment is always rigid, and the `Location` of the nested body determines its relative location inside the host.

The root entity of the bodies' hierarchy is `World`, which is a `BodyCollection` descendant. The `World` contains top-level bodies, which can be physically moved with all the nested bodies attached to them.

Consider the Figure 1 with the example of the bodies hierarchical structure. The robot consists of several bodies, and we choose one of them to be "top-level". Other parts of the robot are attached to it. Some parts are further decomposed, like the manipulator with the gripped object is. The object in front of the robot is added directly to the World, because it is not connected to the robot. In order to place the gripped object on top of the object ahead, we should remove the triangle part from the robot's descendants, and place it into the `BodyCollection` of the free object.

C. Controller: physical engine

Physical engine is a software that provides an approximate simulation of certain properties of physical interactions between objects. For CVARC, we need only a rigid body dynamics with collisions handling. Lots of libraries with such features are available, and the best solution in our case was to include one or several of them in the project.

We had chosen two free, open-source libraries with C# implementation: 3D engine BEPU [14], and 2D engine Farseer [13]. In our opinion, 2D simulation is enough for the robotics competitions, since in most of them neither robots nor playable objects do not interact in the air. The model, however, remains three-dimentional, and so it is possible in any case to introduce a flying or dangling objects that do not interact with the environment. Nevertheless, we implemented both these engines in CVARC for the following reasons. First, 3D physics may be required in the future. Second, it was challenging to make these engines interchangeable, so CVARC would be expandable.

Our aim is to make the model the only entity, which is operated by the developers of rules and bots. The hierarchy of the model is clear and natural, the properties of the model are meaningful and understandable, and we want hide the physics engine behind the model, in order to simplify the developers' work. No direct interaction with the engine is allowed in CVARC. Instead, the engine is connect to the model through the decorator, which plays the controller role.

We used late binding between the model and the the physical engine. The model knows nothing about physics, there is no reference to the physical engine in it. However, the model can report about its changes. The `Body` class implements `INotifyPropertyChanged` interface, and therefore any time when the property is changed, an event is risen. Similarly, `BodyCollection` is an `ObservableCollection`, and reports about its changes also.

The model and the engine are to be created separately, and then the controller binds the model as follows:

- The controller reflects the structure and properties of the bodies in the model by creating instances of the data structures, provided by the physical engine library. This reflection will be different for BEPU and Farseer engines.
- The controller subscribes to `BodyCollection`'s events, and since the collection is observable, the controller knows about insertions and deletions of bodies. When receiving such event, it updates its reflection to keep the correspondence.
- Each body implements `INotifyPropertyChanged` interface, and therefore raises an event whenever a property of the body is changed. The controller subscribes to that event too, and updates the property of the corresponding objects in the reflection. Therefore, the mirrored model is always coherent with the original model.
- When the time comes to perform an iteration, we call the method in the controller, which starts emulation, and then copy the locations of the bodies back to the original model from the reflected one.

The advantages of the late binding in our case are as follows:

- The model remains simple. It contains only a well-understandable properties, such as density, size and location. Moreover, it contains only the properties we expect to be used, while in physical engines lots of properties are stored in bodies, there are many obscure methods to be run and conditions to be met, and because of that it is hard to start working with physical engine. In our implementation, the engine is hidden behind the user-friendly model.
- Since no references to the physical engine are located in the model, the controller can be changed from BEPU to Farseer or to any other supported engine. To support an engine, one should only implement an interface and reflect the model as shown above.

The main disadvantage of the approach is the programmer's inability to use all the functionality of the physical engine. In order to use a property, the property must be represented in the model. However, it can be done relatively easy. Also, we assume that no complex physics is required for virtual robotics competitions. Remember, that CVARC is not an emulator, should not be used to debug the real-world robots, and therefore, in fact, we only need the scene to look plausible. Not so many properties are required for that, and the disadvantage is compensated by the CVARC's easiness of use.

D. View: drawing bodies

The view in CVARC is the most simple part: it just runs over the bodies' hierarchy and draws the bodies at the drawing context. Bodies contain properties to describe how they should be drawn. For example, boxes contain six groups of properties for each side, and these groups allow setting colors or images or textures for the sides. Similarly, cylinders contain three such groups, and a ball can be colored with the single color or texture.

The current view uses DirectX to draw scenes, and that makes the CVARC server requiring Microsoft Windows to operate. We should stress that this is the only part of the system which is not cross-platform. We do not consider it as a great limitation, because it only applies to the server. However, views are as interchangeable as the controllers are, so OpenGL solution can be implemented if required.

Initially, we also had developed Windows Forms solution to draw scenes, which is cross-platform. This view allows drawing the top-view of the scene. The aim was to allow at least some clients to be run at *nix systems for the clients, so they could view the match at the downloaded viewer. However, we had later discovered the WebGL library, which was used to draw the logs at the web browser, and therefore Windows Forms is not needed now, being only a demonstration of the views' interchangeability. The WebGL is not a full-fledged view, it cannot be used to draw the scene at the real-time, and therefore is described in the section 2.

E. Control and feedback

Robot is a wrapper over a selected body in the competitions' world, and this wrapper drives the body by setting its speed. We currently support only differential wheeled robot mechanics, which seem to be most populair for indoor robotics. However, addition of a car mechanics can easily be done if necessary. To manipulate objects in the world, robot changes the bodies in the hierarchy: for example, gripping an object is just putting the object into robot's `BodyCollection`.

Getting feedback is a more complex task, because all three components of the MVC pattern play part in it. For example, the easiest way to get the image from the robot's onboard camera is to urge the DirectX view to create an image from the current robot's location, which is stored in the model. Collision detection naturally comes from physical engine.

However, most of the measurements come from the model. We stress again, that we do not try to reproduce exact low-level output from different devices. For example, to locate the robot, we now use `NavigatorData` type which is a tuple of robot's current basis in 3-dimentional space and the time of when the measurement is taken. For our target audience, it is way easier to manipulate such data than to restore the robot's location from encoders states, or measurements from an accelerometer and a gyroscope. Of course, in real robotics, interacting with sensors electronics and getting the right `NavigatorData` out of sensors' measurements is not an easy task. Many techniques are used to restore in from real sensors' data, and to clear this data from distortions of any kinds. But this seems to be another "boring" task for the computer scientists, which distracts them from the things they are interested in, like definitions of strategies or image processing. This is possible, however, to go one step deeper and provide the distance covered by virtual encoders, accelerations from accelerometers and rotation speeds from gyroscopes.

The model also provides the proximity sensor, which detects the distance to interesting objects (primarily, the opponents' robot), and the depth map, emulating the Kinect [15] sensor's output.

II. How to make your own online competitions

While creating the competitions, one should develop a world as a .NET assembly, which contains the code for the following actions:

- Definition of the locations and descriptions of all bodies that present at the table at the beginning of the round;
- Creation of the robots as wrappers over some bodies;
- Implementation of the robot's interface to enable manipulation with objects;
- Definition of the rules of how scores are counted, based on objects' positions.

This assembly should be referenced in several projects in order to create a functional software, which is explained in the following sections.

The general approach for world's creation is simple: we should just create Bodies and build their hierarchy. Therefore at first we cover the problem of what to do with this world, and the paragraph 2.A explains the process of granting access to the competitions, when the world is created. The paragraph 2.B explains some peculiarities about counting scores and building bots.

A. Granting access to the world

1) Tutorial: Tutorial is an individual application, which allows the user controlling the robot with the keyboard, and seeing the top-view image, the image from the robot's camera, the depth map, the measurements from the sensors and the scores. The tutorial application is to be used by the competitions' developers to debug their rules, as well as by the participants to get used to the rules, to understand which impact their actions would have. To create a tutorial, one need to write down virtually a dozen lines of code: create the world, defined in world's assembly, then map the keyboard to the world-specific actions like gripping or releasing, and after that run the `TutorialApplication` class, defined in `CVARC.dll`.

2) Web access: The web-access to the rules is provided by an Internet-server application, which can be created by the same easy way, as tutorial application does. After the Internet server is deployed, the participants can write programs to work with it. These programs should establish the connection with a given TCP/IP address, and exchange XML packages with it. The example of such exchange (with minor deletions) is shown in the Listing 1.

Listing 1 The example of interaction between the client and the server.

```
Client:
-------
<Hello>
<Participant>John Smith</Participant>
...
</Hello>


Server:
-------
<State>
<Location><X>10</X>...</Location>
<Image>...</Image>
<DepthMap>...</DepthMap>

Client:
-------
<Command>
<Forward>100</Forward>
</Command>

Server:
-------
<Result>
<Link>http://air-labs.ru/view/idXXX</Link>
<Log>...</Log>
</Result>
```

The interaction starts with a hello package, where the participant introduces some detail about himself and the round. The server initializes the world and sends back the measurements of robot's sensors at the round's start. After the analysis, the player returns the command to the robot. In the Listing 1, the command `Forward` is used to move robot 100 cm forward. Alternatives are `Backward`, `Turn` and `Action`. The world "freezes" after sending the package to the participant, and therefore there is no latency in commands. However the total allowed time for the player to think is limited in case of an error in the program, or of the server's misuse. When the round is over, a feedback carrying log is sent to the client.

In spite XML is rarely used in real robotics, we choose this language for the data transmission, because it is a common tool if the software world, and the parsers for this language are available in almost all programming languages. The binary data is encoded in base64 encoding, which is also a typical solution to wrap the binary data in XML. The images are sent in PNG format, which reduces the bandwidth in comparison with BMP format. There are also readers for PNG format for many programming languages.

The web-server is downloadable, so the participants can debug the programs without Internet access. In downloadable versions, no time limit is set, so breakpoints can be used to debug the program. Also, we do not deliver the bots with the downloadable server in order to encourage the participants to try the solutions with the real server at least sometimes.

3) Logs and replay: Logs are being collected during the round, and are sent back to the user when the round ends. Log-file is a list of locations and appearance of all the bodies, presenting at the table during the round. Logs are collected by subscribing to `INotifyPropertyChanged` and `ObservableCollection` events, and therefore are completely separated from the model. Everything the system does with the model is stored in log, but there is no need to do some special actions in order to achieve it.

Logs can be replayed with the original DirectX view. However, it may be inconvenient for those who use *nix operating systems. Because of that, we implemented the log reader with WebGL [16]. WebGL is a tool that allows drawing 3D scenes in a web-browser. To play the log with WebGL, we need to generate WebGL instructions from the flow of bodies' states. This feature allows participants avoiding downloading and installing any software in case they do not wish to do it. They just write down the program that interacts with a remote server and receives an identifier that is linked to the log, which is stored on the server. This log can be viewed in an internet-browser.

Logs are also can be used to create more spectacular representation of the competitions. In robotics competitions, the environment usually consists of the simple shapes of bright yet monotonous colors, because image processing is enormously hard otherwise. But for an outside observer, it is not enjoyable to watch cubes and balls being moved around. In order to improve this impression, the logs can be changed by attaching 3D models for bodies. For example, instead of robot being a blue cube, it may appear as the 3D model with wheels, cameras, etc. It will give an ability to represent the match in a more vivid form, while keeping the actual scene easy for image processing algorithms.

B. Defining the world

1) Counting scores: Scores can come from the following sources:

1) Locations of the bodies. In this case, the aim of the competitions is to deliver playable objects in some specific areas, so scores are assigned when objects are placed in the areas, and revoked when they are removed.
2) Movement of the bodies. If the competitions emulate actions like throwing the objects in baskets, the event

should be set for the corresponding body. When body is at some location, the event is raised and the body is removed from the world (i.e. falls into the basket), and the scores are assigned.

3) Actions of robots. If the task is to push the button or perform the specific action, the trigger is set to the robot's command. When robot receives the command and is in the proper location, the scores are assigned.
4) Collisions. The collisions and ramming are often prohibited in the robotic competitions, and so when a collision is detected, the system should set a penalty.

To collect the scores from all these various sources, we use a singleton pattern for the scores accounting system, and events, attached to the bodies.

2) Creating bots: Bots are the built-in "artificial participants", which opposes the participants in the predictable way. The bots do not see the world through the same interface as the participants do, they see al the bodies' hierarchy, and therefore are simpler to be written. Therefore, the competitions' developer are able to fast create of the decent opponents, which the participants can try they strengths against. Some bots are generic, and can be used in arbitrary contests:

- The bot that does nothing, just moves randomly.
- The bot that tries to stand directly before the opponent, therefore blocking its way. It gives the possibility to test collision detection and replanning subsystems.
- The bot that rams into the opponent and pushes it off its way. It gives the possibility to check the robustness of the control.

3) Building worlds with XAML: The experimental feature of the CVARC is the possibility to partially create worlds with XAML. Consider the code at the Listing 2. The code creates the world and fills its bodies' collection with a single Box, which has one of its sides painted in red.

Listing 2 An example of definition of a body with XAML

```
<World>
  <Box X="100" Y="100" Z="25"
       Width="50" Height="50" Lengh="50">
    <Box.Left>
      <ColoredSurface Color="Red"/>
    </Box.Left>
  </Box>
</World>
```

This feature is experimental. The world can be defined in such fashion (in fact, XAML allows creation of the arbitrary class), and it is the more convenient way that direct C# encoding of the object, at least for the static objects. However, we still did not implement the most crucial feature of drawing the world in the Visual Studio XAML editor. It is important because we want to minimize the tools that are required (or recommended) to work with CVARC in order to keep it simple and easy-to-use. So the addition of our own XAML editor conflicts with this policy. But the requirements for drawing bodies in Visual Studio include the strict attachment to Windows Presentation Framework [17], which will not be multiplatform in any near perspective, and introduction of the new View in CVARC. That makes us really considering if the benefits worth the costs and drawbacks. In the future, we plan either to decline this branch of research, or develop the custom editor.

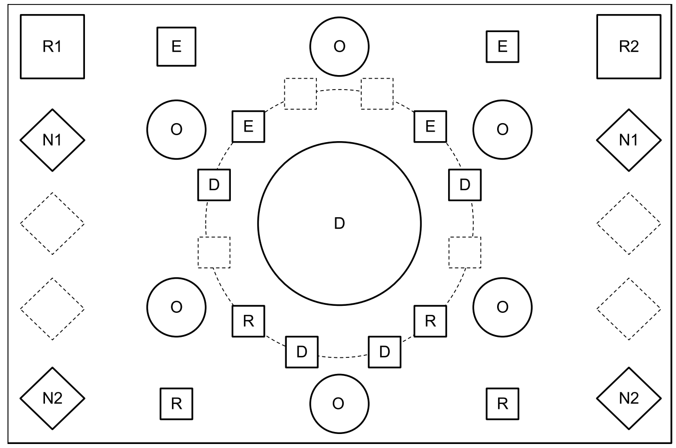


Fig. 2. The configuration of the field in Suuraz Ypud'!

III. Case study: Suuraz Ypud'!

Suuraz Ypud'! (*the dwarven power*, from a fantasy novel "The War of the Mage" by Nick Perumov) are the competitions which we conducted in the April of 2013. They was dated to the Day of the Mathematician and Mechanician, which is an annual festival on the Computer Science department of the Ural Federal University. In this section, we describe rules and bots for this competitions, as well as their outcome.

A. Rules

The legend of the competitions is that three artifact came from the sky, cleaved in parts, fell into the cave with a dragon inside and was later discovered by two hostile dwarven clans. Both clans built robots to retrieve the artifacts, and the participants need to program the robots. The map of the competitions is represented in Figure 2.

At the begin of the round robots are located in corners (R1 and R2). The center of the cave is occupied by the sleeping dragon (D) and obstacles (O). Each part of the artifact is a cube: green emeralds (E), red rubies (R) and barely visible, almost transparent diamonds (D). Some positions are fixed, some are chosen randomly from predefined locations around dragon. The robot can grip a piece of an artifact and carry it to one of the cyan regions, which represent nets (N1 and N2), lowered to the cave. The location of the net N1 is fixed, while N2 can be randomly placed in one of the three predefined locations. If a robot picks up a next piece of the same artifact, they agglutinate and hereby brings more scores. The piece of the artifact, which is different from the one is being carried, cannot be picked. The artifacts which are collected by the opponent can be stolen. The aim is to win more scores than the opponent did. Scores are assigned for collected objects: the diamonds values more that emeralds and rubies, and agglutinated pieces values more that the sum of scores for components.

Our primary concern for the rule design was to offer a gradual complexity of the task. To earn some scores, one need to send to the server a firm sequence of commands: go to the left, pick the object, return to the network and release the object. Taking the red piece is harder, because one need to locate the second network. It can be easily done by image processing, because the network cannot be moved and its color is very distinguishing. Picking the second piece of the red or green artifact is harder, because one need to consider strategies of doing so. Finally, locating the transparent pieces require analysis of the depth map, and therefore is harder, and competitive games like stealing the opponent's treasure while preventing the opponent from doing the same, are extremely hard.

B. *Outcome and Overthinking*

The competitions were held at April 26, we had six registered teams and only three participants that were able to present a working solutions. The rounds between them can be viewed at http://air-labs.ru/index.php/cvarc/suurazYpud online, without downloads or installations. Note that spectacular representation was not implemented, hence the appearence is rather plain.

We believe that reasons for the small count of participants are mainly organisational. Rules and downloadable software were published only one month before the competitions' date, and apparently that was just not enough. The rules are also quite complicated, and although it is possible to earn some points very fast, the students considered such solutions as initially imperfect and did not send them. There is a sharp contrast with real-robotics competition, when the working robot than is capable of doing at least something, is already considered as a great attainment worth trying to compete.

Another distinction from the real-robotics competitions is very aggressive strategies. Two winning participants not only collected their own treasures, but always stealed or at least pushed out of the net the items, which was collected by the opponent. Again, such strategies are unusual for real-robot competitions, but was surprisingly profitable in ours.

We understand that such small-scaled competitions cannot provide a reliable source of information about how rules for virtual competitions should be designed: should the stealing be banned, should the complexity be reduced, etc. However, this competition proves the applicability of CVARC. Some minor errors were discovered during the preparations (for example, it was possible to grip not only the playable items, but also the opponent, and then carry it), but the system itself is overall stable and functional, and can further be used for similar events with minor revisions and refactoring.

IV. Upcoming competitions

We will start the new competitions in the mid October 2013 with the completely new rules. This time, we publish the rules gradually. In case of Suuraz Ypud', this graduality could be as follows:

- Week 1: Tutorial is published. No "O" obstacles, no diamonds, also the nets and all the pieces are located in known locations. The aim is to introduce the world to the participants.
- Week 2: Server is published, so participants can write and debug programs at their local computers.
- Week 3: Web-server with several bots is available.
- Week 4: An intermediate contest.
- Weeks 5-6: Addition of obstacles. Nets and pieces are now located randomly, so technical vision is required.
- Week 7: An intermediate contest.
- Weeks 8-9: Addition of diamonds, so depth map's analysis is required.
- Week 10: A final contest.

The whole project can be considered as an online active course, during which the participants learn about autonomous robots' control. Of course, the forum will be provided, so the participants could discuss their ideas and algorithms, or bring the links to the internet sources with the lectures about robotics, such as mass online learning systems like Coursera. The idea of the project is to create a polygon where students can test their ideas as the gentle introduction in the autonomous robots control. Please subscribe to the news of the project at http://air-labs.ru/index.php/user/registration.

Note that such gradual competitions can be developed for the narrower areas of robotics: for image processing, for introducing the peculiarities of depth maps' analysis, for approaches to the movements' corrections, and so on. In that case, such competitions can be used as an auxiliary software in the corresponding course to carry out laboratory activities.

The competitions in the traditional form with the instant publishing of the rules and 3-4 weeks for the development will then be held in the April, 2014. We are currently considering some new possibilities that the online competitions bring. For example, we want to add active playable elements, which would, for example, run away from robots and try not to be catched. This is near to impossible in real robotics due to enormous organizational work of standardizing or delivering this playable elements to all participants, but is relatively easy in virtual polygons.

Conclusion and future works

In this paper, we presented CVARC, a framework for building the competitions on virtual autonomous robots' control. We also presented the case study of the competitions we conducted with CVARC, and the plans for the future competitions. The source code of CVARC and Suuras Ypud'! competitions is available to download at http://air-labs.ru/index.php/cvarc/download. We will be grateful for suggestions and commentaries that would make the product better.

We also invite the collaborators to join the CVARC community as participants, the developers of competitions or of the framework.

Acknowledgments

The work is supported by RFBR grant 12-01-31168.

References

[1] **Parrot ARDrone 2.0**. *http://ardrone2.parrot.com/*
[2] **The ACM-ICPC International Collegiate Programming Contest**. *http://icpc.baylor.edu/*
[3] **RuCTF Challenges**. *http://ructf.org/index.en.html*
[4] **RoboCup official web-page**. *http://robocup.org*
[5] **ImageCLEF/Robot Vision Challenge**. *imageclef.org/2012/robot*
[6] **International Aerial Robotics Competition**. *http://iarc.angel-strike.com/*
[7] **Eurobot official web-page**. *http://www.eurobot.org/eng/*
[8] Will Goldstone **Unity 3.x Game Development Essentials**. *PACKT Publishing, 2011*
[9] Johns, K. and Taylor, T. **Professional Microsoft Robotics Developer Studio**. *Wrox, 2008*
[10] **Webots 7: fast prototyping and simu;ation of mobile robots**. *http://www.cyberbotics.com/overview*
[11] A. Troelsen **Pro C# 2010 and the .NET 4 Platform**. *Apress, 2010*
[12] Krausner, G. E. and Pope, S. T. **A description of the model-view-controller user interface paradigm in the smalltalk-80 system.**. *Technical report, ParcPlace Systems, Inc., 1988*
[13] **Farseer Physics Engine**. *http://farseerphysics.codeplex.com/*
[14] **BEPUphysics**. *http://bepuphysics.codeplex.com/*
[15] Webb, J. and Ashley, J. **Beginning Kinect Programming with the Microsoft Kinect SDK**. *Apress, 2012*
[16] Parisi, T. **WebGL: Up and Running. Building 3D Graphics for the Web**. *O'Reilly, 2012*
[17] Nathan, A. **Windows Presentation Foundation Unleashed**. *Sams Publishing, 2006*

Why teach robotics using ROS

Stefano Michieletto, Stefano Ghidoni, Enrico Pagello, Michele Moro, and Emanuele Menegatti
Intelligent Autonomous Systems Lab (IAS-Lab)
Department of Information Engineering (DEI)
Faculty of Engineering, The University of Padua
Via Ognissanti 72, I-35129 Padova, Italy
{stefano.michieletto, stefano.ghidoni, enrico.pagello, michele.moro, emanuele.menegatti}@dei.unipd.it

***Abstract*—This paper focuses on the key role played by the adoption of a framework in teaching robotics with a computer science approach in the master in Computer Engineeering. The framework adopted is the Robot Operating System (ROS), which is becoming a standard *de facto* inside the robotics community. The educational activities proposed in this paper are based on a constructionist approach. The Mindstorms NXT robot kit is adopted to trigger the learning challenge. The ROS framework is exploited to drive the students programming methodology during the laboratory activities and to allow students to exercise with the major computer programming paradigms and the best programming practices. The major robotics topics students are involved with are: acquiring data from sensors, connecting sensors to the robot, and navigate the robot to reach the final goal. The positive effects given by this approach are highlighted in this paper by comparing the work recently produced by students with the work produced in the previous years in which ROS was not yet adopted and many different software tools and languages were used. The results of a questionnaire are reported showing that we achieved the didactical objectives we expected as instructors.**

***Index Terms*—Educational robotics, ROS, LEGO NXT robot, teaching robotics.**

I. Introduction

Robotics is a multidisciplinary field which involves researchers from many different research areas, from mechanics to control theory, from electronics to computer science. Thus, robotic competences are taught at different levels and from different points of view through undergraduate and graduate courses. In this paper, we focus on a course on *Autonomous Robotics* offered in the master curriculum of Computer Engineering at the University of Padova.

In our case this is the only course strictly related to Robotics: the lack of previous experience, apart from a general basic knowledge, makes it not easy for the students to tackle the complexity which is behind the building of autonomous robots. As reported also by [20], incremental experiences are essential for this purpose but, even providing a well-planned sequence of experiences, we realized that the code developed by students suffered of one of the major plague of robotics: i.e. little (or almost no) code reuse. Even though reusing code is difficult and possibly involves debugging and integration effort, it is nevertheless an important aspect of software development that should be learned. The absence of code reuse in the old course implementation, was caused by the fact that we used different software tools and programming languages (and this is often the case for robotics courses, see for instance [23], [24]). This is a situation similar to what happens in the robotics community. However, the solution comes from software environments able to offer to different programs the possibility to communicate one with each other sharing a common interface: in a few words, a software framework.

The choice of exploiting a software framework offers several advantages in a high number of real-world robotics applications, and therefore in such scenarios it is often a compulsory choice. But when educational purposes are concerned, some further motivations should justify its adoption: the advantages provided by a framework cannot be fully exploited in this scenario, since time usually allocated to laboratiorial activities is short, and the robotic platform exploited is often quite simple. A very common robot for teaching activities is the Mindstorms NXT: this option is relatively cheap and simple and for these reasons rather frequently adopted in university courses [12][7][4]. Moreover, it should be noted that the NXT platform comes with the NXC language that targets all the specific capabilities of the robot: this means we are comparing a small, easy to learn but hard to spend, procedural programming language targeted to the robotics platform employed for the experiments, with a large, general and complex framework with high potential which is however tricky to understand, for which the NXT is just one of the many platform that can be handled.

The purpose of this paper is to show that integrating a general purpose framework into a university course has a positive didactic impact. Students were able to use a framework to complete the same experiments developed using NXC in the previous years, and the chosen framework (ROS) showed good performance also when used to handle a simple robotic platform. The success of this choice relies on exploiting the framework for the laboratorial experiences without substantial overhead for the students. This way students develop their knowledge in robotics by using tools that are easier to apply to real-world scenarios in which they will be asked to work after graduating.

Given the decision of employing a framework for teaching robotics, an important aspect is to choose which one is best suited, since in the last decade a lot of robotics frameworks have been developed. This is the case of URBI [1], OROCOS [3], YARP [6], Microsoft Robotics Studio [8] and Piaget [9]; however, no one has obtained the proper consensus to become a standard *de facto*. Recently, the scene has changed thanks to the introduction of the Robot Operating System (ROS) [13]. ROS is a framework for robotics with the addition of some operating system functionalities. A great variety of tools are integrated in order to allow easy debug operations and analyze the communication between processes. One of the main advantages that ROS offers among other similar products is the large community that supports, uses and extends the software.

The choice of employing ROS for teaching robotics is important to let the students have experience of a complete and modern software framework for robotics. Moreover, since ROS is likely to become the most popular choice in the future, supporting an increasing number of robotic platforms, its knowledge will enable students to easily handle other robots in the future. Many universities are adopting ROS to teach robotics, including South Carolina, Washington, Brown, Stanford, KU Leuven, Sherbrooke, Tokyo, Sapienza and Leibniz University.

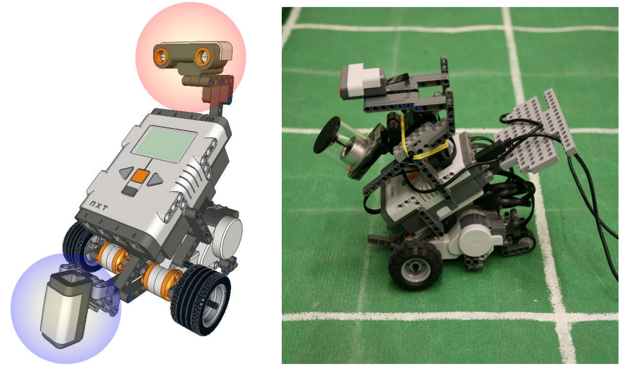
The paper is organized as follows: in section II the robotic course organization will be described, together with the expertise that students should gain with it; in section III the experience with laboratory activities will be summarized, and a comparison between before and after the introduction of ROS will be provided. In section IV the didactic impact of the lab experiences will be evaluated, based on the analysis of the homeworks produced by students, as well as on their answers to a questionnaire. Finally, in section V some final remarks on the choice of employing ROS for teaching robotics will be drawn.

II. Robotics Course for Master in Computer Engineering

The robotics course is based on a mixed approach merging theoretical lectures in class and practical experiences in the laboratory. Lectures aim at building a strong background on robotics fundamentals, perception systems, computer vision, and navigation, while laboratory sessions are meant to let students get acquainted with software tools and algorithms exploited in robotics.

The platform we chose is the Mindstorms NXT 2.0. Several robot kits are available for educational purposes [25], but we believe the NXT offers the right balance of complexity versus modularity [17] (in Italian). NXT is composed by a microcomputer, three motors, and a suite of sensors, including touch sensors, sonars, light sensors, microphones, compass and accelerometers. A set of LEGO parts also comes in the box, letting the user build structures for holding sensors in the preferred position, as shown in figure 1: in (a), a sketch of the model employed in the laboratory experience is shown, equipped with a light sensor (blue bubble) and a sonar (red bubble); in (b), a robot holding an omnidirectional camera.

The strong point of this package is that the LEGO kit provides high flexibility, so that it is possible to build robots with a wide variety of shapes, and choose among a number of different sensor placements. The basic configuration shown in figure 1 (a) (usually called "Tribot") is one of the simplest to control, and has therefore been selected for first laboratory experiences; it is a differential drive robot platform with two driven wheels in the front and a castor wheel on the back. Motion control is rather simple in this case, but some complexity is added by the fact that motors themselves are not very precise, and the way they are controlled (e.g. acceleration curves) has an impact on the trajectories the robot can follow.



(a) (b)

Fig. 1. The NXT in the Tribot configuration (a) with the light sensor (blue) and the ultrasonic sensor (red). In (b), a robot holding an omnidirectional camera placed on a green carpet is shown.

III. Robotic Experiences

In this chapter the teaching experience before and after the introduction of a software framework will be outlined.

A. Past experiences

In the previous years, we taught students the open-source C-like programming language NXC (Not eXactly C) and we used the IDE called BricxCC. With this approach, programs are composed of one or more tasks, and the syntax is easy to read and understand for people with little programming background.

However, it should be noted that the NXC is dedicated to program the NXT platform and even if it is very popular, its knowledge is not exploitable outside this context. Moreover, the NXC language is limited to the sensors included in the NXT package, and it is hard to deal with additional sensors, like a webcam attached to the robot. In such cases some additional software and libraries should be run outside NXC to manage the new sensors.

Robotic frameworks are going towards module-based architectures, often supporting distributed processing, and capable of exploiting network resources for sharing data. Of course, exploitation of such aspects is beyond the scope of the robotics course, however, by employing such new frameworks even for developing the first laboratory experiences with simple robots, it is possible to introduce concepts that will be reused by students when they will face more complex scenarios. Adopting ROS instead of relying on simpler languages, as NXC, presents of course an overhead [10] and a steeper learning curve, but from our experience it was clear that the overhead was limited: the number of hours dedicated to laboratory sessions was the same in the course adopting ROS as in the one in which NXC was employed.

B. Recent experience

Updating the Autonomous Robotics MSc course, a single framework has been adopted in order to make students familiar with programming practices which could be useful in a future job.

As previously discussed, a number of frameworks for robotics exhist, some of which are specifically tailored for didactic purposes, as it is the case of Tekkotsu [18], that cares about real-time aspects of robot programming, and Pyro (Python Robotics) [19]. Almost all of such frameworks are able to support the basic experiences that are proposed during the course, so the main aspects considered choosing the framework to be adopted were: i) the possibility to exploit the same framework also outside the context of the course, i.e., the generality of the framework; ii) the number of supported robots suited for lab experiences (in particular the NXT); iii) the community and the documentation, that represent a valuable help for students. So for example, by adopting Tekkotsu there is a strong constraint on the types of robots that are supported, probably caused by the fact that it is a very recent project. The ROS framework is instead very strong on this point, and has recently become even stronger thanks to the ROS industrial project [5], that is meant to improve the support for industrial robots, making it a proper working tool for professional engineers in the robotics field.

The effectiveness of ROS in teaching is demonstrated by the rather large number of robotics course that have adopted it, including Brown University (USA), Cornell University (USA), University of Birmingham (UK) and obviously Stanford University (USA). The panorama is quite wide, since the robots employed among the courses are quite different, and the tasks assigned to students depend on this: for example, experiences with inverse kinematics are proposed with the PR2 robotic platform. Anyway, a common base about motion planning and basic computer vision can be found in the majority of the courses.

The introduction of ROS did not require to change the laboratory experiences objectives developed in previous years. Such experiences focus on the quantitative aspects typical of engineering and to create a constructivism/constructionism and educational robotics architecture [11]. All didactical goals of the course were kept. In addition, we could add other objectives to the course related to the computer science curriculum: students have the opportunity to write code in a widely used programming language (preferably C++, commonly used in the robotics field) supporting Object-Oriented Programming (OOP) concepts. Among all available frameworks, ROS has been chosen since it supports OOP, and also because its community is very active, and represents a valuable help. A large variety of tutorials are available from which students can easily learn.

In the following, the set of laboratory experiences proposed in the course will be described. They involve the classic challenges of robotics: robot control, navigation, and perception through sensory information. This way students can gain experience on different aspects, and build a robot that has a certain degree of intelligence. Multiple sensors have been employed: ultrasonic proximity sensor, color sensor, and an omnidirectional camera: this way it is possible to increase the complexity of the sensing tasks in the different laboratory experiences.

Experience 1: Obstacle avoidance

In the first experience, students have to plan a robot path in order to avoid two obstacles, represented by cones. The robot has to go around the first cone and stop 3 cm behind the second one for 10 seconds, and finally come back, as shown in Figure 2.

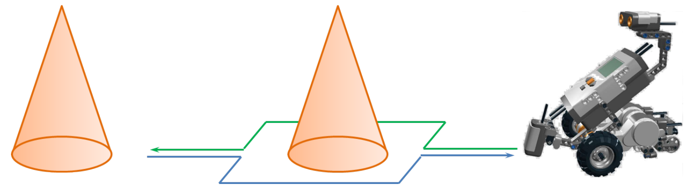


Fig. 2. Robot behavior for the first experience

Robotics objectives: The first objective is to make students familiar with robots, and their motion and sensors. They have to deal with noisy sonar sensor, motor inaccuracy and odometry imprecision. During this experience, students work with basic ROS modules: they start practicing with ROS master (`roscore` or `.launch` files), then they explore nodes and topics functionalities (`rostopic` and `rxgraph`). Once students are familiar with these basic concepts, they can evaluate robot characteristics by teleoperating it using the keyboard. A simple visualizer (`rviz`) is also available, which eases the result visualization. In order to do this we developed a basic package to provide students the NXT model, the robot controller to translate velocity command into joint rotations and the teleoperation program. Finally, students create a new package and develop their own module, which communicate with the others following the publisher/subscriber mechanism, which is exploited also for reading sensors, e.g. acquiring range data, and for controlling robot movements. The experience involves robotics topics like interpreting uncertain sensor data, navigation and control, and motion planning.

Computer science objectives: The experience is meant to make students review the concepts of data structure and class. They can understand how data are handled in the framework by looking at prebuilt ROS messages, and they are also asked to analyze the package structure in order to know how they depend one from each other; this way they will be able to develop their own packages in a similar way. Students will also analyze the callback construct which is covered through the publisher/subscriber communication method. In this first experience a simple problem is proposed, so that students can try to solve it in a fast procedural way as they usually do. Nevertheless, an object oriented approach is promoted to build independent entities to control the robot.

Experience 2: Path planning

The goal of the second experience is robot navigation into a map composed of *N*x*M* cells, that are represented using a green carpet with a white grid, over which the robot moves. Students have to extend the algorithm created in the first experience to make the robot go from a *Start* cell to a *Goal* cell while avoiding obstacles, and understanding its own motion inside the map. To guide the robot motion a color sensor is exploited in order to identify the squares borders: as long as the robot moves, the sensor indicates whether it is going over the green carpet or a white stripe. The representation of the robot movements is simplified, since it can only go to the north, south, east, west squares, that is, it cannot move diagonally; this means the activities for navigating across the map are

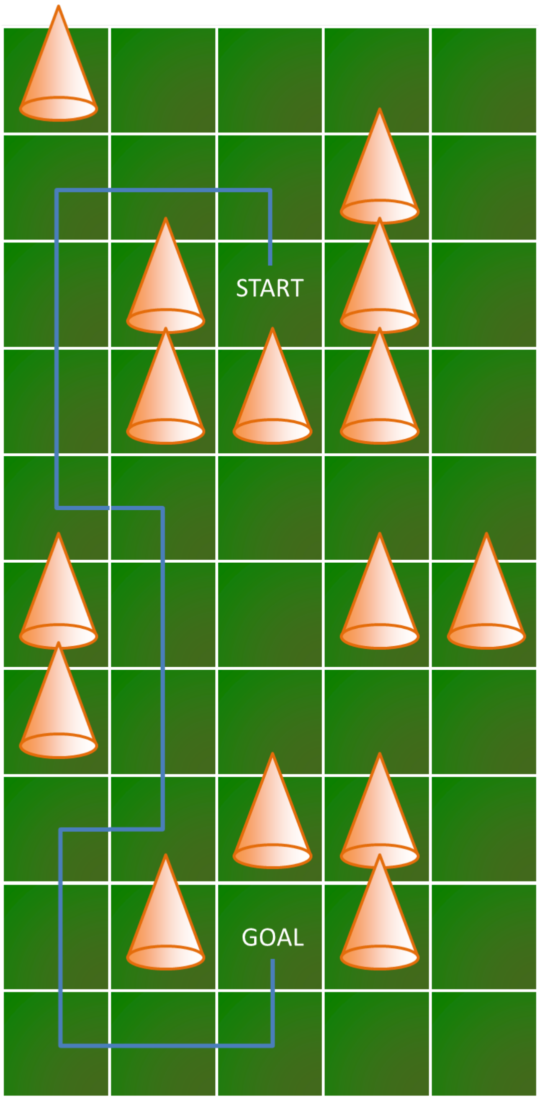


Fig. 3. Map example

basically forward translations by one cell and rotations. Since employed motors are not extremely precise, the orientation of the robot is affected by inaccuracies that students need to deal with: once the robot crosses a white stripe, it performs left and right rotations in order to understand its direction with respect to the map grid.

The experience is divided into two scenarios: in the first one, obstacles in the map are fixed and students know where they are, whereas in the second one they can be moved, hence the correct route cannot be known in advance. Figure 3 shows an example of map and a possible path to reach the goal.

Once the robot has reached the goal, it is asked to push a touch sensor, which is in turn connected with ROS. This tells the system the experiment was successful.

Robotics objectives: In this experience students have to develop a path planning algorithm choosing among those presented in the course. They deal with the intensity sensor data and exploit communication through ROS modules. Understanding modules is very important working with ROS. Students are pushed to split their code into a reasonable set of modules that should communicate among each other. Using sensory information the robot has to recognize the motion from one cell to another, navigate in the map and avoid obstacles in order to reach the *Goal*. Since the map is not known in the second scenario, an internal representation of the checked cells and robot position should be developed. The purpose of the last part of the experience is to learn how to exploit communication between two different devices (the robot and the touch sensor) using ROS. The main robotics topics faced by students in this experience are localizing and mapping, and multiple-robot coordination.

Computer science objectives: A more complex experience highlights the importance of a well-structured code. Students need to model the robot and the map as two different entities, and to generate a message flow that starts from the former and goes into the latter. The change from a static map to a dynamic one also helps students to reconsider their choices in software design in order to better identify which data belongs to which structure. Finally, the communication between the robot moving on the map and the one handling the touch sensor promote the abstraction of the robot class. The first one should implement a very smart behavior in order to move and manage several sensors, while the second one has to take care of the touch sensor only. Both elements are the same entity (a robot), so they comes from the same generic class, but each one is a different specific class.

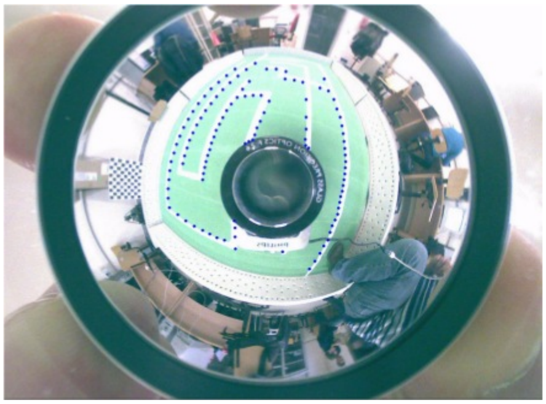
Experience 3: Perception using computer vision

In this lab experience a new sensor, based on computer vision, is introduced. The robot is given the same task of *Experience 2*, that is, reaching a goal on a map with moving obstacles, but this time the sensory suite is different: in place of the infrared sensor, an omnidirectional camera is exploited.

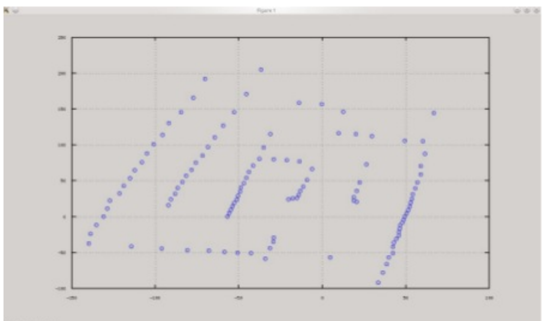
As a first step, students are asked to build their own omnidirectional sensor and calibrate it with the Ocam Calib tool [14]. They should then find the optimal placing on the robot, which involves two main parameters: the height from the ground, and the orientation of the axis of the mirror, which can be vertical or slightly tilted. At this point the real laboratory experience, divided into two parts, starts.

As a first task, students are asked to implement a vision-based module that behaves like the infrared sensor of the previous experience, thus providing the same output. This can be achieved by selecting a small Region Of Interest (ROI) in the image, in which objects at a certain distance will appear, and observing its content.

The second task is more focused on robotic vision, since students are asked to develop a ROS module capable of analyzing all white stripes visible on the carpet. In particular, the lines which are closer should be detected to evaluate their distances to the robot, in order to gather its position inside the cell and its orientation with respect to the grid; in figure 4 it is possible to see the result when the white stripes composing a labyrinth are analyzed. The resulting module is able to overcome the data provided by the infrared sensor: thanks to the vision sensor, it is possible to detect stripes at a rather high distance, and to recover the orientation of the robot with



(a) Lego NXT starting configuration.



(b) Lego NXT omnidirectional configuration.

Fig. 4. Result of the line detection algorithms when a labyrinth is analyzed. (a), and a bird's eye view reconstructed from such image (b).

respect to the grid by looking at a single image, instead of performing rotations while passing over a line.

Robotics objectives: In previous experiences students have already developed some hands-on robots knowledge, therefore guidelines do not specify details on how to place the vision sensor, nor how to develop vision algorithms. Students are asked to explore different solutions, and find the best-suited ones: for example, by properly tilting the omnidirectional camera it is possible to extend the detection range, but this leads to a slightly more complex processing. Facing the trade-off between complexity and accuracy is an important aspect at this stage. Another important objective is to face computer vision with omnidirectional images, which is widely used in robotics. Students will learn how to build an omnidirectional sensor, calibrate it, and exploit the huge amount of information included in each single image. Finally, experience with ROS modules and how to manage communication among them will be developed: students will create a module for acquiring images, and use its output to feed another module implementing image processing algorithms. This experience combines the image acquisition and processing topics together with localizing and mapping topics inherited from the preceding experience.

Computer science objectives: One of main goals of this experience is the creation of a module that can replace an existing sensor module and then improve it with new features. This implies code modularity, and also encourage the use of inheritance and design patterns in order to make the work easier when classes are well-designed. Using an environment similar to the one introduced in the second experience helps students to focus on the computer vision activity; however, students that chose a good software design in the previous experience will be able to easily reuse it in this third one.

IV. Evaluation of didactic impact

In order to obtain a feedback on the effectiveness of the adoption of ROS, we compared the results of the group of students that attended the Autonomous Robotics course two years ago before we introduced ROS and the group of last year, that exploited the ROS framework. The comparison exploits on one hand some objective considerations about problem-solving methods and code developing and on the other hand the subjective students' opinions about laboratory experiences, satisfaction about the course, and future expectations.

A. Comparison with the past

The use of ROS as a common framework pushes students to develop algorithms in a structured environment. This was not the case in the previous years, since NXC is still a procedural programming language. Developing code in NXC limits students to work in a procedural paradigm. NXC provides functions or data structures, but students do not actually use them, because they feel they can code faster if they write their algorithms in a single file. The analysis of the code of last year reveled students favored cut-and-paste to think to a general structure for the code; for the same reason, they preferred to use global variables to passing parameters, and created ad-hoc solutions to problem generalization. While using ROS, this year's students are pushed to organize their software into modules, reuse their data structures and classes, exploit class inheritance. They will also experience the power of message sharing mechanism, which is an important service offered by robotics frameworks.

The proposed experiences are designed so that students can have an advantage if they are able to correctly formalize the problem and choose a proper software design for implementing solutions, since in this way each experience can be built on top of the previous ones. In Table I the results of the analysis of the software produced by this year's students is reported. The source code (about 1000 lines for each experience) of sixteen students (grouped by two) has been analyzed looking for the following OOP concepts:

- features coded in functions or methods to be applied in different situations (**code reuse**);
- similar characteristics and values related to each other in order to build a single structure (**structured data**);
- modeling of real entities into objects that represents them and their features (**classes**).

Data reported in Table I represent the percentage of students groups implementing each OOP concept at least once in their source code. As it can be seen, code reuse and adoption of structured data strongly increased after the first experience.

B. Students satisfaction

Students were asked to fill an anonymous questionnaire summarized in Table II. The answer to each question is

TABLE I. DATA FROM COURSE STUDENTS AFTER THE INTRODUCTION OF A COMMON FRAMEWORK.

	1^{st} exp.	2^{nd} exp.	3^{rd} exp.
Code reuse	75%	100%	100%
Structured data	38%	88%	100%
Classes	63%	88%	88%

represented by a choice among four states: *Very much* (green), *Enough* (blue), *A little* (red), *Not at all* (yellow). Some of the statements are quite similar, in order to emphasize small differences that could be hidden by a multiple choice answer.

The questionnaire was meant to test several aspects of the laboratory activity, like:

- exploitation of students' background in terms of programming skills and software engineering;
- effort spent in developing the experiences;
- closeness with future job activities.

Answers to the questionnaire highlight that programming capabilities students were sufficient, even though they would like to had more programming experience (Q2); moreover, such capabilities improved during the practical experience in the laboratory (Q3). From the software engineering point of view, it turned out that the ROS framework forced students to adopt a modular approach for their software, which eased its reuse (Q4). Students appreciated team work, and are convinced that it is very difficult to achieve important results by working alone (Q7-Q8). Students showed a moderate confidence on the fact that expertise coming from lab experiences could be reused in a future job (Q10,Q16): answers to these questions were based on the current working experience that a certain number of students already had, while the others answered based on what already graduated colleagues told them. Students seemed to greatly appreciate the hands-on approach of the course, and would agree on increasing the number of courses adopting this approach (Q9, Q13), even though this means an increase in work load (Q11-Q12). Finally, students also appreciated the way experiences gradually increase in complexity (Q14-Q15).

Overall, the questionnaire demonstrates that choices made while designing the course had a good impact over several aspects, including code production, putting into practice theoretical concepts studied during lectures, and working with a structured framework like ROS.

V. CONCLUSION

This paper presented a series of experiences targeted to MSc students attending the Autonomous Robotics course. Experiences already defined for the course in the previous years, based on the constructivist approach, are now developed inside a robotic framework, that forces students to get in touch with advanced software structure, and take advantage of the services it offers. The introduction of ROS as a framework pushes students to use OOP concepts thanks to the highly structured environment they have to work with. The overhead given by understanding and learning how to use a new framework, besides its intrinsic added value, is compensated by the later ease to develop code for the subsequent laboratory experiences, to integrate new sensors, and to interact with different devices. Finally, being able to handle complex softwares like ROS is a strong reward for students, which realize they have learnt how to deal with real (and complex) robotic frameworks.

The course includes a set of laboratory experiences that represent an important feedback for students' achievements. Experiences test the capability of controlling a robot (experience 1), sensing the environment with simple sensors and modifying the robot's behavior accordingly (experience 2) and handling more complex sensors that need high-level processing (experience 3). This can be seen as a small but complete set of abilities students should gain to deal with robots, and the positive outcome of such experiences is the ultimate proof of their achievements. For this reason, the way experiences cover a number of subjects, and their increasing complexity level has been stressed in the paper.

The analysis of a report for each laboratory experience and of the developed code made it possible to verify students' comprehension of robotics basics, their use of complex syntactic constructs and their problem-solving capabilities. Finally, students' satisfaction was tested by means of a questionnaire. The results highlight a good response both regarding how students' expectations were met, as well as improvements in robotics and programming skills. This has also been assessed by testing the robots moving on the map, and observing how they deal with a dynamic environment. Since all students were able to correctly complete all the experiences, even though going through a number of difficulties, it is possible to conclude that the proposed set of experiments is correctly planned.

The laboratory experiences are currently limited by both the robots used (more advanced ones would require stronger investiments) and by the time that can be dedicated to experiences. If such limitations could be overcome, an interesting extension would be to face challenges posed by humanoid robots, starting from the gait. More advanced tasks, like grasping, are too complex to be practically solved in the context of this course. The introduction of the approach presented in this paper into other courses than Autonomous Robotics is also planned.

REFERENCES

[1] J. Baillie. URBI: Towards a Universal Robotic Low-Level Programming Language. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3219–3224. IEEE, 2005.

[2] J. C. Baillie. Urbi 2: Introduction to concurrent real-time programming. In *Proceedings of the Third International Workshop on Software Development and Integration in Robotics*. IEEE ICRA, May 2008.

[3] H. Bruyninckx, P. Soetens, and B. Koninckx. The real-time motion control core of the Orocos project. In *IEEE International Conference on Robotics and Automation*, pages 2766–2771, 2003.

[4] A. Cruz-Martin. J.A. Fernandez-Madrigal, C. Galindo, J. Gonzalez-Jimenez, C. Stockmans-Daou, and J.L. Blanco-Claraco. A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses. *Computers & Education*, 59(3):974–988, Nov. 2012.

[5] S. Edwards, and C. Lewis. Ros-industrial:applying the robot operating system (ros) to industrial applications. *IEEE International Conference on Robotics and Automation*, ECHORD Workshop, St Paul, Minnesota, May 2012.

[6] P. Fitzpatrick, G. Metta, and L. Natale. Towards long-lived robot genes. *Robot. Auton. Syst.*, 56(1):29–45, Jan. 2008.

[7] J.M. Gomez-de-Gabriel, A. Mandow, J. Fernandez-Lozano, and A.J. Garcia-Cerezo. Using LEGO NXT Mobile Robots With LabVIEW for Undergraduate Courses on Mechatronics. *IEEE Trans. on Education*, 54(1):41–47, Feb. 2011.

[8] J. Jackson. Microsoft robotics studio: A technical introduction. *IEEE Robotics & Automation Magazine*, 14(4):82–87, Dec. 2007.

[9] J.D. Dessimoz, P.F. Gauthey, and H. Omori, Piaget Environment for the Development and Intelligent Control of Mobile, Cooperative Agents and Industrial Robots. *International Symposium for Robotics (ISR)*, Aug. 2012.

[10] T. Koletschka, and A. Hofmann. Technikum Wien's entry in the Robotour'11 competition. In *Proceedings of the 2nd International Conference on Robotics in Education (Rie2011)*, pages 151–156, Wien, Austria, September 2011.

[11] E. Menegatti, and M. Moro. Educational robotics from high-school to master of science. In *SIMPAR Workshop on Teaching Robotics, Teaching with Robotics*, pages 484–493, Darmstadt, Germany, November 2010.

[12] P. Ranganathan, R. Schultz, and M. Mardani. Use Of Lego NXT Mindstorms Brick In Engineering Education. In *Proceedings of the 2008 ASEE North Midwest Section Conference Educating the Engineer of 2020*, Platteville, Wisconsin, USA, October 2008.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[14] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[15] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots, 2001. In *Artificial Intelligence*, Volume 128, Issues 1?2, May 2001, Pages 99-141.

[16] J. Arlegui, E. Menegatti, M. Moro, A. Pina. Robotics, Computer Science curricula and Interdisciplinary activities In *Proceedings of the TERECoP Workshop "Teaching with robotics: didactic approaches and experiences* pp.10-21 November 3-4, 2008 Venice (Italy).

[17] M. Moro, E. Menegatti, F. Sella, M. Perona. Imparare con la Robotica - Applicazioni di problem solving. Edizioni Erickson Trento (Italy), pp.1-196, 2011.

[18] E. Tira-Thompson, D.S. Touretzky. The Tekkotsu robotics development environment. In *Proc. IEEE ICRA 2011* pp. 6084-6089 May, 2011.

[19] D. Blank, D. Kumar, L. Meeden, H. Yanco. Pyro: A python-based versatile programming environment for teaching robotics. In *Journal on Educational Resources in Computing (JERIC)*, Volume 4, Number 3, p.3, 2004.

[20] A. Nourdine. Teaching fundamentals of robotics to computer scientists. In *Computer Applications in Engineering Education* Volume 19, Issue 3, pages 615–620, September 2011.

[21] Robot Operating System. *www.ros.org*

[22] Terecop project: Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods. *www.terecop.eu*

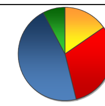
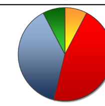
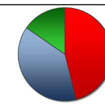
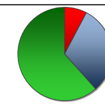
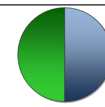
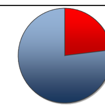
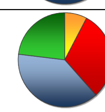
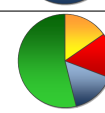
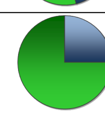
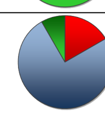
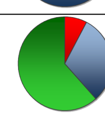
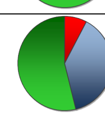
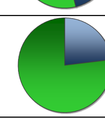
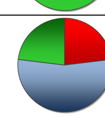
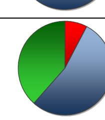
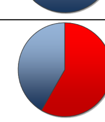
[23] G. Cielniak, N. Bellotto, T. Duckett Integrating Vision and Robotics into the Computer Science Curriculum In *Workshop on Teaching Robotics, Teaching with Robotics*, pages 484–493, Riva del Garda, Italy, April 2012.

[24] A. Albers, M. Frietsch, V. Bartenbach, G. Robens, N. Burkhardt. A New Robotics Laboratory for Interdisciplinary Mechatronic Education In *SIMPAR Workshop on Teaching Robotics, Teaching with Robotics*, pages 456–464, Darmstadt, Germany, November 2010.

[25] M. Ruzzenente, M. Koo, K. Nielsen, L. Grespan, and P. Fiorini, A review of robotics kits for tertiary education In *Workshop on Teaching Robotics, Teaching with Robotics*, pages 44–50, Riva del Garda, Italy, April 2012.

TABLE II. RESULTS OF THE QUESTIONNAIRE.

Legend: Not at all; A little; Enough; Very much

Q1.	During the experiences of the robotics course I exploited knowledge acquired in other courses that I had not put into practice.	
Q2.	Programming skills I developed in previous courses were sufficient for working on the ROS framework.	
Q3.	I gained new programming experience because I had to work with a complex framework as ROS is.	
Q4.	In order to efficiently work with ROS, I have been forced to divide my software into modules, which in turn made it easier to reuse it.	
Q5.	In my future job I will be asked to work with modular software structures similar to ROS.	
Q6.	By working with ROS I have got in touch with a well structured software, that has been a source of inspiration for new ideas on how to develop my software.	
Q7.	By working in groups I improved my ability to split tasks among people.	
Q8.	By working in groups we were able to reach results I would not have reached alone.	
Q9.	By working with real robots we needed to deal with practical problems that would have not shown up in a simulated environment.	
Q10.	Thanks to the lab experiences I developed expertise I will exploit in my future job.	
Q11.	Lab experiences require a lot of time to be completed, but are the only way I can develop expertise that are useful for my future job.	
Q12.	Lab experiences required an excessive work load, and should correspond to a higher number of credits.	
Q13.	I would like that a larger number of courses would be based on a mixed approach including both theory and laboratory experiences, in order to put into practice what we learn during lectures.	
Q14.	The complexity of lab experiences increases gradually.	
Q15.	What I learnt in each experience has been useful for the following ones.	
Q16.	In my future job, I will be asked to work on topics that are similar to the ones faced in the laboratory.	

Introducing modern robotics with ROS and Arduino

Igor Zubrycki, Grzegorz Granosik
Lodz University of Technology
tel +48 42 6312554
Email: igor.zubrycki@dokt.p.lodz.pl, granosik@p.lodz.pl

***Abstract*—This paper describes our experience with introducing modern robotics through Robot Operating System. ROS framework allows rapid robot prototyping and gives access to many state-of-the-art robotic solutions. It is however, software oriented and requires its users to understand well software development ideas and methods. While teaching undergraduate students ROS, we came up with some solutions how to introduce it to people without deep computer science background. The paper presents our *Mymodel robot* application that simplifies modeling of the robots using URDF format and some Arduino based programs.**

I. Introduction

The robotics curriculum must contain the laboratory stage. This is absolutely necessary to familiarize students with real robots, their control systems and software. However, an interesting approach is to proceed this stage by modeling and simulation. For several years we have been using two convenient applications to teach students how to model and simulate robots and the whole robotic stands, namely: the combination of Robotics Toolbox (for Matlab) with RoboWorks, and the EasyRob software [**?**]. These programs provide tools to build graphical models of robots, to manipulate them, and analyze kinematics and dynamics. Recently, much more powerful solution appeared that can support both simulation and real control stages of robotics curriculum.

ROS (Robot Operating System) is an unified and robust framework for robot modelling, control and visualisation [**?**]. It is a more and more popular tool for rapid prototyping of robot software as it provides an easy way to integrate, test and reuse algorithms constructed by robotic community around the world. And it is an open source, too. However, because of its capabilities and scope, ROS has a fairly steep learning curve [**?**]. This problem is more distinct if the user has only little computer science background, what is the case for the bachelor program in Automatics and Robotics at the Lodz University of Technology. We believe though, that the benefits of using ROS are vast and worth our work of finding skilful methods, easy to use tools and appropriate knowledge, to involve even less ”computer science type” students to use this modern robotic tool. In this paper we will describe methods and tools, that worked best in our case. Arduino is the hardware platform we have employed in this quest.

II. Motivation

ROS is a tool used by robotic teams worldwide when designing large robotics systems. The main reasons for its popularity, that also led us to the introduction of ROS for our students, are as follows [**?**]:

1) ability to rapid prototype. There is a multitude of tools and libraries that were created around ROS. It is possible to connect and ”pipeline” these tools literally in a few hours. Because of that, relatively small teams and beginning students do not need to ”reinvent the wheel” and can create entire robotic applications.
2) modern software architecture. ROS is a modern software architecture that allows to connect easily different applications and devices. Users can build systems where most of processes work in parallel and on different machines without building multithreading or networking procedures by themselves.
3) ”Thin” ideology. Programs to be used in ROS do not need to be highly integrated. There only has to be a small executable running that exposes program’s functionality to ROS or extracts some information from it. This opens the way to reuse all specific tools that were created outside ROS.
4) Ease of debugging, visualisation and logging. ROS has a number of tools that enable users to check and save system’s state. Users can see system’s state graph (*rxgraph* tool), plot different variables online with *rxplot*, or visualise whole robot and its sensors readings by using *Rviz*. All data can be easily archived and replayed by using rosbag.
5) ROS is well documented and supported. Beginners can find tutorials online, there is ROS book [**?**] and lively ROS forum [**?**].
6) Free and Open Source. Most of ROS packages is open source and has straightforward licences [**?**]. This simplifies development and allows to engage wide range of contributors both from within and outside academia. Open source gives also partial guarantee that long time development will be possible – even if library creator stops development, we will be able to improve and compile it by ourselves.

However, the introduction of such sophisticated and modern framework to our students involved some difficulties, these are the most important we have faced in our work:

1) transition from simple single threaded systems. Our students have different computer science skills but the experience of most of them is limited to single threaded applications – designed in the MATLAB environment or on devices such as Atmega microcontrollers. To use ROS effectively students have to understand its structure and philosophy – mainly the need to treat each functionality as a single part – that will become *ROS Node*, and which will communicate to others only by using *ROS Messages*.
2) ROS is an open framework and most of its power

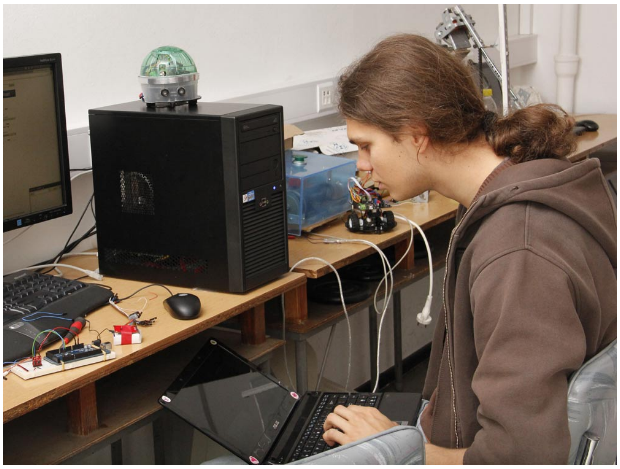


Fig. 1. Picture from one of our ROS introduction workshops

comes from the possibility to use other teams' solutions. Unfortunately, these solutions are usually "proofs of concept" and using them becomes an integration task, which is difficult for the people with little experience with the software development.

3) ROS is a Linux tool and to use it effectively, users have to have experience in this environment. Basic parts of ROS are available as Ubuntu Packages but more advanced and less popular ROS tools are only available as sources on the Github or other internet revision control services. Users need to know basics about version control, compiling, makefiles, etc.
4) ROS framework is rapidly developing, there are many versions that have different APIs, different tools and functionalities. Tutorials that work in one ROS version, sometimes do not work or even mislead in other – what is utterly frustrating for the beginners.

III. Target group

As the Robot Control Department we provide robotic related courses for students from different faculties of our university. We teach basic robotics, robot control – that are mainly about industrial robotics as well as more advanced subjects such as mobile robotics, vision systems or rehabilitation and service robotics. We understand, that a large part of these courses can be based on ROS, what would allow students to work with single framework or even on one project throughout different courses.

To derive and test solutions we have conducted series of workshops for the second year of the bachelor course in Automatics and Robotics (Fig. 1). This group had already learned some basic engineering subjects, programming (C++, MATLAB) and had several courses on the electrical engineering. Unfortunately, these students had very little experience with subjects from computer science curriculum – software development, object oriented programming, etc. Therefore, training them to use ROS turned out as a challenge.

IV. Solutions

To enable our students working with ROS we came up with a number of ideas. At the beginning we planned to base our teaching mainly on the internet tutorials [?] and a textbook [?]. Students were supposed to build small applications that would contribute in our bigger projects. Meetings were to be spent on discussing matters related to robotics.

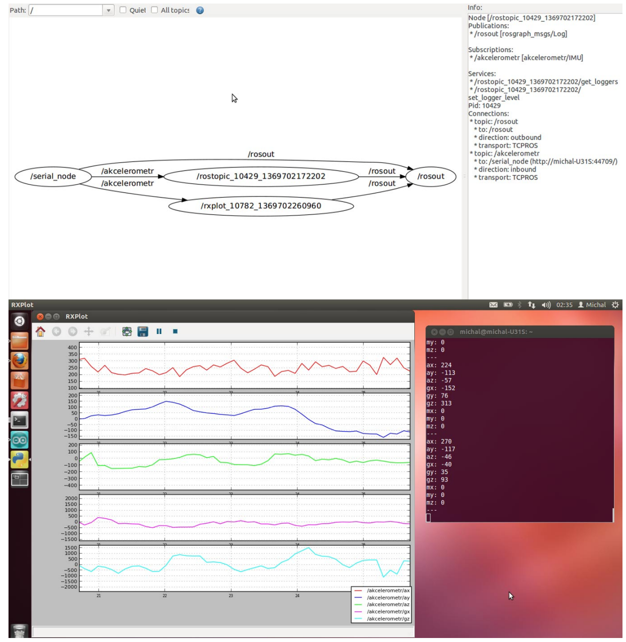


Fig. 2. ROS Computation Graph created by our student Michal Kielan to read IMU sensor readings and resulting graphs

Unfortunately, because of the weaknesses of ROS, described in Section II, our students, especially the less computer adept were unable to start practical work, even after completing all beginners tutorials. Also discussions with them proved that they do not understand the ROS nomenclature and basics of work.

We understood, that our students would learn ROS faster if we used their existing skills and divided their work into practical modules, through which they would be guided.

A. Arduino and rosserial

We have found that students weren't able to grasp basic ROS ideas of using Publish/Subscribe and Services based only on the tutorials. What helped us enormously was introduction of *rosserial* package and *rosserial_arduino* [?].

The *rosserial* is a protocol that allows to connect different hardware to ROS system using serial connection. The type of activity (Subscriber/ Publisher or Service) and a message format are defined directly on the device, with the use of *rosserial* libraries. There is also a python node on the host computer that forwards messages and makes nodes created on the device visible to ROS. As long as devices use *rosserial* format to communicate using serial connection, they can have any functionality. It provides an excellent way to communicate with simple real time devices such as microcontrollers or sensor arrays. We have found that *rosserial_arduino* – the extension of *rosserial* for Arduino platform [?] – can significantly help in integration of custom hardware with ROS and is also a convenient way to teach our students ROS. There are several reasons for that:

Fig. 3. Micromouse, a small differential drive robot based on a modified Arduino board, used on our workshops and controlled through bluetooth by ROS

1) Arduino has excellent documentation, IDE and community support. We introduced it independently with our robotic platform – robo mouse, small differential drive robot based on modified arduino board.
2) Using real sensors or actuators, connected to Arduino, helped our students see and understand ROS functionality, and benefit from it.
3) Students were able to do "real work" and to use ROS in their own projects – this was an enormous motivation.

To test this approach we have used small mobile robots (see Fig. 3) equipped with differentially driven wheels with encoders, IR distance sensors, sonars, RC receiver, BlueTooth modules, and Romeo controller, which is Arduino-compatible and integrated with H-bridges.

Students realized several projects: sonar reading, chaotic movements of the robots with obstacle avoidance, remote controlled mouse, web-controlled mouse. Additionally, we have demonstrated other projects: smartphone-controlled mouse, sensor glove readings.

B. Working in groups

We have also found that working in groups – pairs or trios – made learning more effective. Students shared their experiences from working independently and explained to each other how different functionalities of ROS work.

To make teamwork easier we have set up a forum and encouraged students to use Git-based source code management (which we taught them by using internet teaching games).

Teamwork motivated students – it was harder for them to explain delays or lack of progress to their peers than to us. It also involved their different skills – each of the students could work on the part of the project he felt best in. This somehow reflects the ROS philosophy.

C. Robot modeling and debugging

Students from our test group have already passed Introductory robotics course and have gained some theory on industrial robots, forward and inverse kinematics, and manipulator dynamics. This knowledge could be easily illustrated by ROS set to work with simulation tools such as Gazebo. It has a built-in transformation system that can dynamically construct a transformation tree – known from kinematics. In order to further prepare the robot's visualization or simulation the URDF model is required. URDF – unified robot description format is XML based structure that describes important robot properties [?]:

- shape and dimensions (*link* element's *origin* element properties and *collision* element properties)
- visual properties (*link* element's *visual* element properties)
- inertial properties (*link* element's mass and inertia properties)
- robot joint characteristics and limits (*joint* element's properties)
- placement and properties of sensors (*sensor* element's properties)

Example of URDF description and resulting tree are shown in Fig. 4.

```
<robot name="KAWASAKIRS03N">
  <link name="base">
    <visual>
      <geometry>
        <box size="2 2 0.1"/>
      </geometry>
      <material name="black">
        <color rgba="0 0 0 0"/>
      </material>
    </visual>
    <collision>
      <geometry>
        <box size="2 2 0.1"/>
      </geometry>
    </collision>
  </link>
  <link name="zl00">
    <visual>
      <geometry>
        <box size="1.4 1.7 0.5"/>
      </geometry>
      <material name="AlmostWhite">
        <color rgba="0 0 0.9 0"/>
      </material>
      <origin rpy="0 0 0"/>
    </visual>
    <collision>
      <geometry>
        <box size="1.4 1.7 0.5"/>
      </geometry>
    </collision>
  </link>
  <joint name="JT1" type="revolute">
    <parent link="base"/>
    <child link="zl00"/>
    <axis xyz="0 0 1"/>
    <limit lower="-2.7925" upper="2.7925" effort="1" velocity="1"/>
    <origin xyz="0 0 1.1"/>
  </joint>
</robot>
```

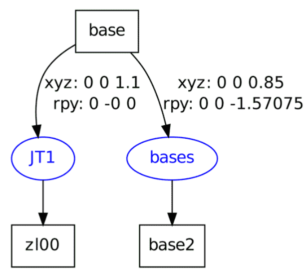


Fig. 4. Example URDF description and resulting tree

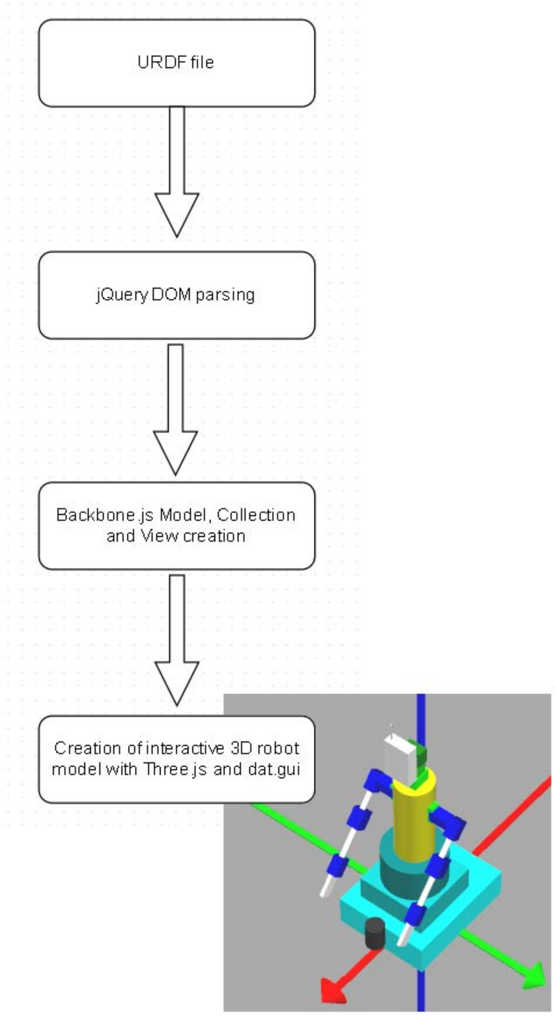


Fig. 5. *Mymodel robot* aplication diagram

Even though the structure of URDF file is quite clear, students had some problems to create these descriptions from scratch. The biggest reason for that is the number of steps required to launch such a file to visualize robot and manipulate its joints. It would be much easier if they could interactively modify a file and see results immediately.

1) Online app Mymodel robot: We have created a tool to simplify testing of URDF model files by presenting them directly in the web browser. It is a network tool, that does not require any installation on the students behalf – only a modern browser (that supports WebGl format) is needed. Our aim was to make usage of this tool as simple as possible. Users only need to put their URDF file into form field, which is then parsed into a robot model and shown in the browser with appropriate control sliders for all movable joints. *Mymodel robot* application diagram is shown in Fig. 5.

The application that does most of the processing is created using modern web applications libraries and it follows MVC (Model View Controller) model where data is isolated from its view. The most important advantages of the proposed application are:

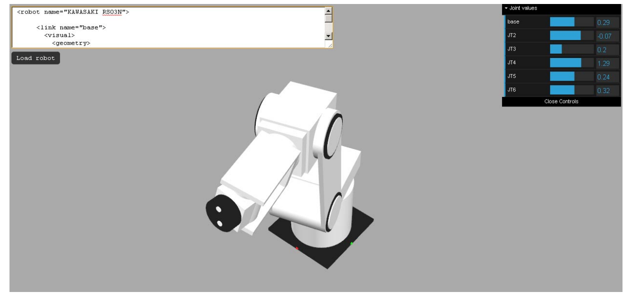


Fig. 6. KAWASAKI R303N robot model created by Pawel Guzdraj and Patryk Blesinski with *Mymodel robot* application

1) Beginning users do not install Linux and ROS Framework to start working with URDF files
2) Tool is easy and straightforward, allowing fast model testing and evaluation
3) Users have an easy way to show their models to other people
4) As a web tool, there is a guarantee that each student uses the same tool, what simplifies debugging

Our tool was well received by the participants of our workshops, we have introduced it also to students in our normal curriculum, where they could write and test models of industrial manipulators, an example of a Robot model created by our student is shown in Fig. 6. Also members of ROS Users forum were interested in our tool. We have received a number of emails with suggestions for further development or usage. One of them was suggesting to use it on ROS websites to show the manipulable image of the robot model used in the project.

Currently, there are some other projects that aim to make browser based tools for ROS – even move whole Rviz visualization tool to browser. We expect that the learning curve for these tool will be still rather steep as they are too sophisticated. From our experience and ROS forum suggestions there is a need for simple, straightforward tools that can be used by beginners.

D. Working on big projects

Students of Robotics and Automation course usually plan to become engineers. Because of that, they are entirely focused on acquiring practical skills and receiving experience that would be appreciated by their future employers.

To motivate them more, we tried to involve even the beginning students in our "real" work: they could participate in modeling a mobile manipulator which we are designing or be involved in the preparation of robotics contests like Robotour.

ROS is used in all of our current projects and we demonstrated to our students the functionality that is available. We have spend considerable time describing and demonstrating our in-house manufactured sensor glove that was designed to control a three finger gripper [**?**] – right now we are in the process of connecting a real gripper to ROS so that the sensor glove will not only be controlling a Gazebo simulation(shown in Fig. 7) but also a mechanical unit.

As a result students become motivated as they could see real live applications of ROS framework. They could also relate to ROS better – their questions have become specified.

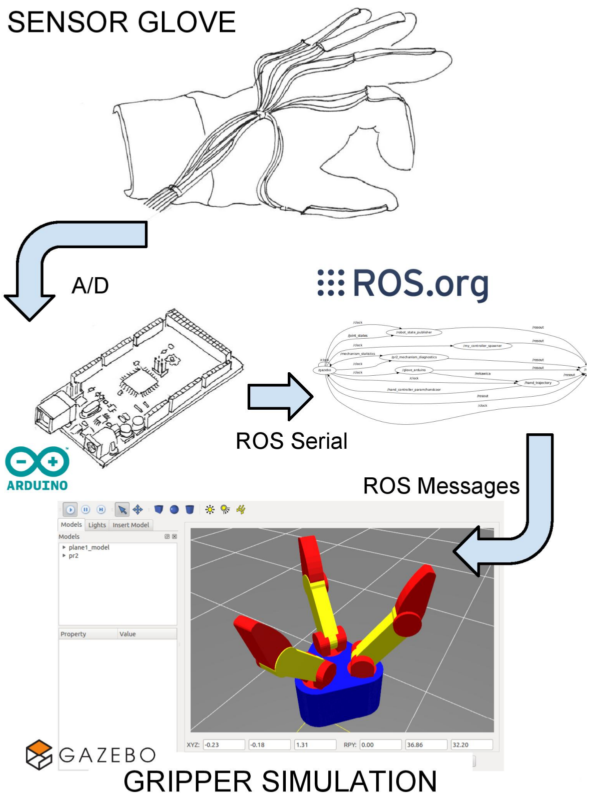


Fig. 7. Structure of our Sensor Glove acquisition and control system [10].

V. Next steps and conclusions

Our main conclusion of the work we have already done with ROS is that the best way to introduce this framework is to use simplified solutions. Our students are not experts in computer science and have little experience in typical software development. Yet they have broad knowledge in other disciplines that can be used to introduce them to ROS. Making use of physical devices such as Arduino boards with sensors makes ROS functionality easier to understand as well as gives more motivation than just a simulation.

Experiments with online tools convinced us that this approach is also attractive. We can introduce students to some parts of ROS functionality without having them install the whole ROS system. This will be especially valuable in normal curriculum where time and students' motivation is limited.

Students work more effectively with some guidance (in addition to tutorials and books) and when divided into groups working together on the same project they can teach and motivate each other.

Our last observation is that it is important to show students some "impressive demos". ROS is very broad and students need to have reasons to explore it. After showing them our applications which use Kinect, sensor gloves or smartphones they were much more motivated and wanted to increase their knowledge.

References

[1] G. Granosik and D. Zarychta, "Application of simulation software for robot modeling," in *Proc. of VII National Robotics Conference*, Ladek Zdroj, pp. 101–109.

[2] M. Quigley, K. Conley, B. 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*, vol. 3, no. 3.2, 2009.

[3] B. Smart, "Teaching robotics with ROS: Experiences, suggestions, and tales of woe," Jun. 2012, accessed: 4 May 2013. [Online]. Available: http://www.youtube.com/watch?v=5Agb7qXi3Cc

[4] P. Goebel, *ROS By Example*. Raleigh, N.C.: Lulu, 2013.

[5] "ROS answers forum," May 2013, accessed: 5 May 2013. [Online]. Available: http://answers.ros.org/questions/

[6] "ROS developers guide," May 2013, accessed: 5 May 2013. [Online]. Available: http://www.ros.org/wiki/DevelopersGuide

[7] "ROS tutorials," May 2013, accessed: 3 May 2013. [Online]. Available: http://www.ros.org/wiki/ROS/Tutorials

[8] "rosserial tutorials," Oct. 2011, accessed: 20 April 2013. [Online]. Available: http://www.ros.org/wiki/rosserial/Tutorials

[9] "URDF tutorials," May 2013, accessed: 6 May 2013. [Online]. Available: http://www.ros.org/wiki/urdf/Tutorials

[10] I. Zubrycki and G. Granosik, "Test setup for multi-finger gripper control based on robot operating system (ros)," in *Proc. of 9th Int. Workshop on Robot Motion and Control, July 3-5, 2013, Wasowo, Poland*, 2013, to appear.

Designing a competition for autonomous robots with a restricted set of sensors with a case study of LEGO NXT

Wojciech Marian Czarnecki, Krzysztof Szarzyński, Andrzej Wójtowicz
Adam Mickiewicz University in Poznań
Faculty of Mathematics and Computer Science
{w.czarnecki, quati, andre}@amu.edu.pl

***Abstract*—Arrangements for a competition are not only difficult in terms of logistics of the event, but also require an assurance of quality. In this paper we analyze limitations which arise from design of the contest for robots equipped with a very poor sensor set. This issue is faintly explored – up to now research work usually has focused on results of a certain task and in addition it assumed almost having a free hand with a choice of components. The discussed question is significant on the grounds of primary principles: objectivity in grading, equal opportunities among participants and preservation of attractiveness of the tournament at the same time.**

All of our actions have been evaluated through several years of existence of the PozRobot robotics contest. Over a three-year period we had an opportunity to test our approach on nearly 50 teams and almost 150 contestants from many Polish universities.

We analyze various aspects of performing the tournament and we indicate solutions to common problems, e.g. we touch upon dealing with an arena and objects which are placed on it. In particular, we propose a list of features which a well-designed competition should fulfill. To show our experience we describe an instance of a model competition. We outline new directions of further development of the contest, which are connected with a structure of the arena and possible changes in the limited set of the sensors.

I. Introduction

The best way to put theory into practice is to introduce a challenge. This way we motivate students to use their creativity, learn [1], [2] and improve their technical and social skills [3].

Robot construction is expensive, quite complicated and most of the robots are very specialized. That is why majority of the robotic tournaments are based on similar competitions like Sumo or Soccer [4]. The competitors face with a difficult task of constructing the best robot for each competition and when they decide on a project, it can be very hard to change its parameters.

Many students do not have any access to robotic laboratories equipped enough to construct specialized robots. One of the solutions to this problem is to restrict the competition to an easily modifiable, cheap and uniform robotic platform. This way every team can experiment with the shape and the construction of their competition robot.

Our core assumption was to design a competition with very diverse tasks. This implies contestants' focus on advanced artificial intelligence methods rather than construction.

II. Programmable robotics kit

LEGO Mindstorms NXT is a good example of affordable robot construction kit. First of all, LEGO blocks give almost unlimited potential in creating versatile, mobile robots and the NXT Intelligent Brick provides sufficient computing power. Secondly, NXT comes with a variety of sensors. In context of our competition, the biggest advantage of NXT over other robotic platform (e.g. Arduino) is its standardized modularity which entails cost-free modifications of its construction.

The NXT system is currently available in two versions, both are similar and come equipped with:

- 3 servo motors with an odometer;
- 2 touch sensors;
- 1 light/color sensor;
- 1 ultrasonic distance sensor.

The NXT 1.0 version is equipped with a simple light sensor (returning the ambient or reflected light level), and the 2.0 version comes with a color sensor instead (which is able to measure amount of ambient or reflected light and recognize up to six basic colors). All of the sensors are compatible with both bricks. NXT uses a RJ-11 connector to connect the motors and sensors which are compatible with I^2C interface.

Many manufacturers provide auxiliary sensors and devices which can be used with the NXT brick. Most of them outperform the basic kit sensors in precision. Our goal was to create equal chances for all contestants, so we decided to limit the competitions only to the basic sensors manufactured by LEGO and included in the NXT kit. In the context of mechanical parts we allow use of any LEGO parts.

Restriction to only NXT robots with the basic set of sensors presents a big problem in designing the competition. Sometimes the very same limits motivate the participants to create very ingenious robots and this "atmosphere" makes the tournament interesting and enjoyable.

A. Motors and sensors

Basic NXT components are very simple and have some limitations. When the participants use them they have to take those constraints into consideration. Based on our experience we describe some characteristics of four mostly used devices:

1) The simplest, yet the most precise, is the binary **touch sensor**. It has a single button placed at the tip of a

Proceedings of the 4th International Conference on Robotics in Education
Lodz University of Technology, Poland, September 19-20, 2013

standard Lego NXT sensor casing. The button is small, so the contestants usually build some kind of leverage mechanism to indirectly contact with the environment and maximize their chances of detecting an obstacle or an object.

2) All of the **servo motors** are equipped with an odometer. According to the manufacture the device can work with a 1° precision [5]. To increase torque or speed the participants use gears. Unfortunately, LEGO blocks are not usually well fitted and sometimes the robot looses a grip with a surface. This accounts into an increasing odometer error. The teams cope with this problem by usage of reorienting algorithms.
3) The most characteristic sensor is the **ultrasonic distance sensor**. Its shape stands out from the others. It uses a ultrasonic transmitter located in the right "eye" and a receiver in the left one. Ultrasonic waves are good in assessing a distance to objects from couple of centimeters up to 1 meter but the accuracy is related to the shape of the distant object. For example, a simple cube placed in front of the sensor can appear 10-20 cm further if the cube is placed at an angle, or in some cases even "disappear" (due to the ultrasonic waves bouncing off and not returning to the receiver). This requires some experience from the contestants when using this sensor.
4) The most used sensors in our competition are the **light and color sensors**. Because of small reliability of the ultrasonic sensor the participants use the light or color sensor for navigation. The color sensor returns more information about the environment than the light sensor, but when used properly they are very precise. The lighting conditions and proper calibration are crucial when using these sensors. Some of the contestants build a casing around the sensor to minimize the external light conditions influence.

III. PozRobot tournament

PozRobot[1] is an annual national robotics tournament held in Poznań and organized by Adam Mickiewicz University. The first edition in 2009 was directed at primary and middle school children. In 2010 a special category – *Student* was introduced. The main focus of the *Student* category is to advertise artificial intelligence and robotics among students from all over Poland. The competition designed for students are more complex and strongly rely on usage of multiple sensors and advanced algorithms. Throughout the years we gained experience in creating competitions and developed a set of basic rules.

A. Basic assumptions

When we design competitions we try to follow four assumptions:

1) Competitions should be as objective as possible, so that we could evaluate the team based on just one run per team.
2) The competition setup or arena must be easy to reproduce. The contest is open to participants from all over the country, the competitions changes every year, and we can not demand from the participants to incur high costs of participation.
3) Competition design should blur the differences between the NXT 1.0 and NXT 2.0 sets. The color sensor from NXT 2.0 is much more accurate than the light sensor from NXT 1.0 and gives a big advantage to the teams which use it.
4) The tasks should maximally encourage usage of data gathered from the sensors and complex artificial intelligence methods.

The process of creating a competition for robots with a restricted set of sensors can be divided into four main elements:

1) an arena that is suited for their sensors;
2) objects which the robot is able to recognize;
3) character of the contest, a choice between tasks for a single or multiple agents;
4) a method of communication with the judges.

Each of these elements will be discussed in next sections. We will show the best solutions based on our experience from previous years.

B. Arenas

The environment which robots explore is the key element in every competition. Its form and quality decides the final result of a run of a robot and distinguishes between actual skills and pure luck. In classic robotics tournament this issue is not as crucial – robots can be equipped with any type of sensors, including cameras, laser range finders etc. In this case the robots can approximate their position in virtually any environment (a good example is the contest DARPA Urban Challenge [6]).

What kind of restrictions does a usage of a single light/color sensor, ultrasonic distance sensor or two touch sensors give? Because of large odometry errors (due to the precision of LEGO servo motors and the robot construction made from LEGO bricks not specially designed metal parts) it is necessary to add easily recognizable key points to the environment. In accordance with the sensor restrictions, the only reliable source of data is the light/color sensor, which suggests that the key points must be recognizable by it. To ensure constant data access those key points should be located on the ground.

We divide the arenas into two types: discrete and continuous. The discrete arenas are those wherein the key points locations allow to interpret the localization problem in context of a simple finite space. The best example, used frequently in PozRobot (and lately in other Polish NXT competitions or Istrobot [7]) is a matrix-type arena (see Fig. 1) which gives two major benefits. Firstly, it discretizes the problem by giving known in advance structure of N accessible fields. Secondly, the simple geometry provides easy methods of correcting the odometry errors exploiting a reorientation algorithm using the

[1] http://pozrobot.pl

matrix lines (see Fig. 2). It is worth noticing that the usage of the single light/color sensor makes line following algorithms ineffective in this type of competitions, so the line geometry is the key factor in robots unambiguous positioning. Other interesting form of the arena with positioning support is an arena consisting of concentric circles, where the given method will set the robot perpendicular to the tangent of the circle (in the direction of the arena center).

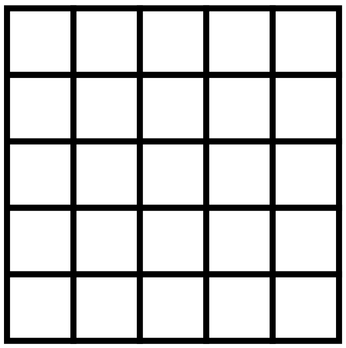


Fig. 1. An outlook on the matrix-type arena. Perpendicular lines help robots with self-localization.

Second type are continuous arenas where the robots explore an environment with much less key points, usually limited by a thick black line and with no additional information on its surface. We experimented with this kind of setup in the previous years, but the results had clearly shown that this approach is wrong. The results in continuous competitions were very badly correlated with the final results of participants (see Table I). Low correlation of such competitions like Marbles and Overtaking are consequence of direct interaction between robots which sometimes led to a jam. We draw a conclusion that continuous competitions are random. In addition the contestants were pointing that those competitions present a low objectivity level. Correlation in the last column of the table I is a Spearman correlation between scores of robots in some particular task and their final results after whole competition. One of the tasks – namely Bumper, was a very specific one, where construction of the arena was negligible so discrete/continuous distinction does not apply.

TABLE I
POZROBOT TASKS COMPARISON

name	single/multi agent	discrete/continuous	correlation
Bumper[1]	single	NA	0.85
OCR[2]	single	discrete	0.81
Map[3]	single	discrete	0.74
GPS[4]	single	continuous	0.54
Marbles[5]	multi	discrete	0.19
Overtaking[6]	multi	continuous	-0.12

[1] pursuit of a randomly moving object
[2] pattern recognition (see Section IV)
[3] detection positions of known landmarks
[4] self-localization problem
[5] competitive gathering of objects
[6] race of two robots

C. Objects

Second important matter of each competition are the objects that the robot must find, recognize, compare or transport. Again the key factor is to choose the objects, so that the restricted set of sensors allows to complete the task. Another impediment is the maximal number of servo motors connected to the NXT brick. Two of them must be used to movement of the robot – this leaves only one for manipulations of the objects and movement of sensors. So if the robot uses a light/color sensor to navigate (it is directed accurately down) it must rely on an auxiliary distance and touch sensors. The other possibility is that the robot uses the servo motor to change the position of the light/color sensor. In this case it is not hard to connect the sensor with some kind of a manipulator, but it is practically impossible to use it with a movable ultrasonic or a touch sensor.

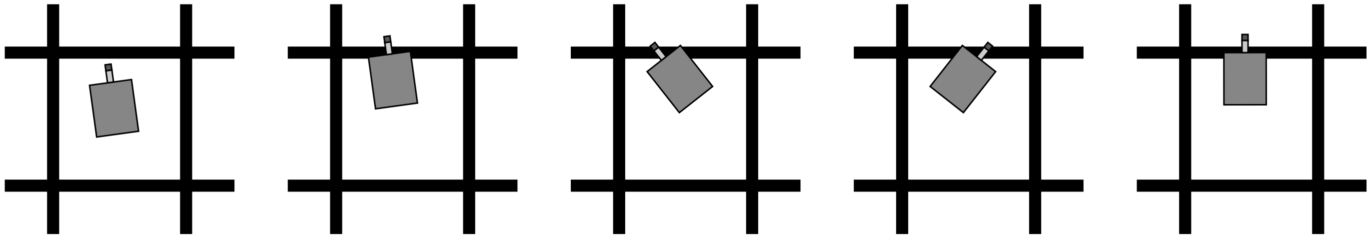
Because of this problems the objects must differ at least in size, preferably the height. This feature is easy enough to recognize by all of the sensors and is independent of the robots position on the arena (the difference of heights between two objects on a 2D arena is maintained). The difference in a color of the objects is not a good feature. The main problem is maintaining a backward compatibility with NXT 1.0 sets that uses the light sensor. This sensor can reliably distinguish no more than three colors ”on the ground” and two ”in the air” (two colors of objects and one ”ambient color” meaning the absence of the object).

We experimented with many different objects ranging from cardboard models, through bricks, up to balls and LEGO blocks constructions. Our experience shows that the best option is to use cuboids built from LEGO blocks. Their doubtless advantage is that they are standardized and the participants from all over the country can easily reproduce them. In addition their modularity allows forming different shapes (for example cuboids of different height).

We believe that the optimal solution is to use objects that can be distinguished by using the color or just by measuring their shapes. For example we propose cuboids build from LEGO blocks with given height and each construction is also built from different colored blocks. This way we give the participants two ways of recognizing the structures and also allow them to minimize errors by confronting the measurements from two different sensors. An alternative solution of the recognition problem is to encode information about an object directly onto the arena (for example by using a printed barcode in front of the object) but this will be the topic of our future experiments.

D. Competition

It is doubtless that robotics tournaments are more attractive that algorithmic competitions. And what could be more impressive than two robots fighting each other? Unfortunately, creating a competition in which two robots simultaneously perform their tasks and can physically interact with each other are not a good choice for NXT tournaments. The basic issue



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

Fig. 2. The reorientation algorithm for the matrix-type arena. (a) Let us consider a robot (a dark-gray rectangle) which can turn in place, equipped with a light/color sensor (protruding pointer in the front) directed to a floor. Due to odometry errors it may happen that the robot is not in a desired position (perpendicular to a line). (b) The robot moves forward as long as its sensor overshoots the line. (c) Then it stops and turns left until the sensor detects the line. (d) Afterwards it turns right – again until the sensor detects the line, but this time the robot also counts motors rotation angle X. (e) Finally, the robot turns left by $\frac{X}{2}$ rotations and as a result it stays perpendicularly to the opposite line.

is the problem of detecting a collision with the other robot, especially when using the basic set of sensors.

We experimented with those types of competitions in the early editions of PozRobot. Definitely most of the ”duels” wherein both robots had the opportunity of contact ended in some form of deadlock or pushing one of the robots randomly out of the arena (see Table I). This leads to an obvious conclusion that we should not introduce competitions that create a risk of physical contact between the robots. The lack of direct competition increases the objectivity of scores. It is worth noticing that competitions with two competing robots also requires the usage of the tournament formula which prevents objective scoring of all runs.

E. Methods of communication

The last key element is the form of communication with the judges. Except the most obvious one – direct manipulation of objects on the arena, we have three more possibilities:

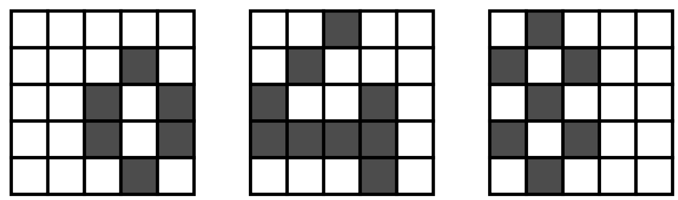
1) displaying the results on the robot’s LCD;
2) emitting a sound signal;
3) communication using Bluetooth.

These three ways help to improve the competition, i.e. by independent scoring of its different parts. The robots task can be moving some physical objects across the arena but is additionally scored for collecting data about the environment. To verify the robots belief we use those mechanisms e.g. we require it to display the arena map on its LCD. The Bluetooth communication although giving the greatest opportunities used to be the most unreliable. It lead to unnecessary increase in complexity of the competition and to dependence from external factors. Because of this reason we stopped using this method of communication.

This way we can distinguish robots which successfully completed parts of the task from these who did not.

IV. Exemplary competition

One of the most interesting competition played on PozRobot was ”OCR”. The arena was a 5×5 matrix where some of the fields were filled with a red color (pixels) and the rest was white (empty fields). The pixels were shaped to resemble a digit (see Fig. 3) unknown to the participants (but the same for all starting teams). There was no known finite set of symbols, only some basic rules were given to the participants. In the center of each field lied a single 2×4 pin LEGO brick. The task for robot was to collect and deliver to the starting point a number of blocks given by the encoded digit. As in each of the PozRobot competitions – robots dimensions were limited to the 30 cm cube and had to be fully autonomous. It was also forbidden to alter the arena anyhow (except moving LEGO bricks).



(a) (b) (c)

Fig. 3. Sample digits from OCR task. A size of a single pixel is 20 cm x 20 cm. Boundary lines have 2 cm width.

The scoring was based on:

1) correct mapping of the map and displaying it on to the LCD;
2) minimizing the number of visited fields;
3) correct digit recognition;
4) delivering the correct amount of blocks;
5) time of completion.

This competition fulfilled all of the key assumptions given the type of robots:

1) The arena was discrete and allowed for easy navigation.
2) It was a single agent competition.
3) It was multilevel evaluated, minimizing the variance of the results.
4) It required maximal usage of rudimentary knowledge (in the context of a restricted set of sensors and in the fact that the highest score was achieved by minimizing the number of visited fields – which is an additional optimization factor).

One could observe many interesting ideas for completing OCR competition. Firstly, it is worth noticing that contestants used various artificial intelligence methods containing (but not limited to):

- A* algorithm for rapid path-finding to the most informative fields on the arena;
- K-NN supervised classifier for scanned digits recognition;
- Lightweight implementation of feed-forward neural network used for color recognition and for digits recognition.

Secondly, we could also observe very interesting constructions. The restricted set of sensors/actuators is not only the limitation – it also motivates creative thinking. One of the teams built robot with smaller bot inside, composed of one motor (for movement) and color sensor (for fields scanning), that was sent to distant parts of the arena while robot was just standing in one position (see Fig. 4). Contest rules stated that robot ”visits” some field if its main brick is placed above it – so this strategy maximized the amount of points gained for not visiting fields while still collecting all required information about the arena.

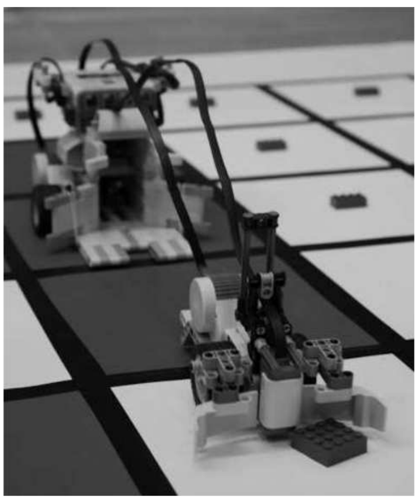


Fig. 4. One of the OCR robots containing smaller bot inside, used to scan distant arena fields.

OCR was very well received by both contestants and viewers. Even though some teams did not score any points (or even got negative ones – see Table II for details), most of robots did quite well (low score was a result of e.g. exception thrown by the contestants code after their robot correctly mapped the whole arena). Task itself was a challenging problem that correctly distinguished between very good robots and average ones. We have also received positive response directly from the contestants. One of the Jagiellonian University teams referred to OCR as *the best designed competition for NXT robots*.

V. Possible development

Firstly, interaction between robots on the map, especially giving permission to fight, turns a competition to a spectacular show (e.g. Sumo). Naturally it is a tip of a hat to the audience but as we stated earlier this approach simply leads to a random results. Certain consensus might be so-called separated rivalry. As an example we can imagine a situation where robots are walled off and compete for some shared resources located in holes of the wall. This illusive permission raises attractiveness of the competition and ensures objectivity in scoring.

TABLE II
OCR Scoreboard

team name	points
Jagiellonian Team Karel 2.0	72.5
superkonfodopieszczaczoszybotron	40
Robomaniacs Prime	25
Who's NeXT?	25
Goffery z Dzemorem	23
bierzsiedogarow.pl 2	21
Aniolki Charliego	14
Jagiellonian Team Valentino 2.0	12
RoboWarriors	7
GRIK	5
Yet Another Robomaniacs Team	0
Garwutki	0
Robomaniacs Academy	0
ROBOLE	0
bierzsiedogarow.pl	-3

Secondly, in the course of time we may consider resignation from adjustment to the light sensor from NXT 1.0. This case looks similarly to the RCX – the first LEGO Mindstorms generation, which nowadays is used very rarely. Moreover, in 2013 LEGO plans release the third generation – EV3 [8], which may speed up a change in competitors' equipment. This could give us a chance to put into a map larger variety of points of interest.

Thirdly, we consider creation new kinds of map, despite of foregoing matrix and circle. Of course it should follow all mentioned earlier principles. The most promising area to explore are maps with some kind of knowledge encoded in barcodes and improved interactivity (as most of the currently considered arenas were fully static). This way one can augment the robots reality with arbitrary complex data. One such idea is to place (also NXT powered) non-player robots (NPR) on some of the matrix fields and use barcodes to inform competing robot about the action, that this NPR will perform if its touch sensor is hit. This would not only improve the attractiveness of competitions, but also would create a new abstraction layer for solving which sophisticated artificial intelligence algorithms would be required.

Last but not least – we want to open our competition for contestants from other European countries. Hopefully PozRobot 2014 will be our first International Robotics Competition for LEGO Mindstorms robots.

VI. Conclusion

As a result of several years in organizing the PozRobot tournament we gained some experience in selection of appropriate tasks. The restricted set of the sensors ensures equal chances among competitors in terms of technology but present a great challenge to the organizers (in designing a competition which gives equal chances) and to contestants (who need to use creative thinking in overcoming the limitations of the restricted set of the sensors). It is beyond doubt that a choice of a discrete map helps with evaluation of a task and reduces problem of the imperfect sensors.

This year we organize together with Jagiellonian University the first edition of KrakRobot competition, based on the exact same concepts as PozRobot. These two contests will be alternating in following years. All this year's tasks are discrete, single agent, and using objects easily distinguishable by light sensor and touch/ultrasonic one.

We are interested in feedback from other organizers, especially with regard to fixing problems connected with continuous maps. We deeply believe that our collaborative effort will make robotic competitions much better in the future.

References

[1] F. Wyffels, K. Bruneel, P.-J. Kindermans, M. D'Haene, P. Woestyn, P. Bertels, and B. Schrauwen, "Robot competitions trick students into learning," in *Proceedings of 2nd International Conference on Robotics in Education (RiE 2011)*, R. Stelzer and K. Jafarmadar, Eds. INNOC - Austrian Society for Innovative Computer Sciences, 2011, pp. 47–52.

[2] J. Stier, G. Zechel, and M. Beitelschmidt, "A Robot Competition to Encourage First-Year Students in Mechatronic Sciences," in *Research and Education in Robotics – EUROBOT 2011*. Springer, 2011, pp. 288–299.

[3] M. Kandlhofer, G. Steinbauer, P. Sundström, and A. Weiss, "Evaluating the Long-Term Impact of RoboCupJunior: A First Investigation," in *Proceedings of the 3rd International Conference on Robotics in Education, RiE2012*. MatfyzPress, Czech Republic, 2012, pp. 87–94, ISBN 978-80-7378-219-1.

[4] D. Calkins, "An Overview of RoboGames [Competitions]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 1, pp. 14–15, 2011.

[5] (2013) Interactive Servo Motor. [Online]. Available: http://shop.lego.com/en-US/Interactive-Servo-Motor-9842

[6] (2013) DARPA Urban Challenge. [Online]. Available: http://archive.darpa.mil/grandchallenge/

[7] R. Balogh, "Ketchup House – A Promising Robotic Contest," in *Proceedings of the 3rd International Conference on Robotics in Education, RiE2012*. MatfyzPress, Czech Republic, 2012, pp. 41–45, ISBN 978-80-7378-219-1.

[8] (2013) Announcing LEGO MINDSTORMS EV3. [Online]. Available: http://mindstorms.lego.com/en-us/News/ReadMore/Default.aspx?id=476243

RoboCupJunior: Promoting STEM Education with Robotics Competition

Amy Eguchi
Division of Education
Bloomfield College
New Jersey, USA
amy_eguchi@bloomfield.edu

Luis Almeida
Dep. de Eng. Electrotécnica e de Computadores
Universidade do Porto/Faculdade de Engenharia
Porto, Portugal
lda@fe.up.pt

***Abstract*—This paper addresses RoboCupJunior, an educational robotics initiative that aims to enhance learning through educational robotics competitions around the world. RoboCupJunior is a division of RoboCup, a robotics initiative that aims to promote Robotics and AI research, by offering publicly appealing, but formidable challenges. RoboCupJunior has three distinguished leagues – Soccer, Rescue and Dance, which attract students from all the continents. RoboCupJunior periodically conducts self-study of the impacts that RoboCupJunior has on participating students' learning. This paper reports the most recent study conducted during RoboCupJunior 2012 competition and discusses the positive results that this study shows. The results of the study highlighted that the participation in RoboCupJunior competitions enhances the participating students' interests in STEM fields and studying further in post-secondary education.**

Keywords—Educational Robotics; Educational Robotics Competitions; STEM education

INTRODUCTION

Robotics in Education (RiE) has been attracting more and more attention in recent years. What contributed to the gaining popularity are the educational robotics competitions for school age children around the world. Some of the most popular robotics competitions include the FIRST Robotics Competition, the FIRST Tech Challenge, the FIRST LEGO League, and the Junior FIRST LEGO League, organized by The FIRST organization (http://www.usfirst.org/); BotBall organized by KISS – The Institute for Practical Robotics (http://www.botball.org/); WRO - World Robot Olympiad organized by the World Robot Olympiad Association (http://www.wroboto.org/); RoboChallenge (http://www.robotchallenge.org/) and RoboCupJunior organized by the RoboCup Federation (http://www.robocupjunior.org). Educational Robotics competitions have been in existence for more than two decades despite the main growth occurring in the last decade. For example, the FIRST Robotics Competition, BotBall, and RoboboCupJunior all planted the seeds into robotics competition far back in the late 90s.

Those competitions employ goal-oriented and project-based approaches to teaching, which are popular approaches in the fields of engineering, computer science, and artificial intelligence, but have not been widely integrated in pre-university education. In that sense, educational robotics competitions provide a unique learning opportunity for school age children. The educational goals for students to accomplish differ between competitions. Some of the educational robotics competitions, including FIRST competitions and BotBall, set new tasks or themes that teams work on every year. Other competitions, including RoboCupJunior, do not change the goals or tasks year to year and focus on continuity of student learning.

Positive impacts on learning among participating students have been reported from those educational robotics competitions [1-7]. Some of the highlighted impacts that educational competitions can have include:

- Increased confidence in using technology [2],
- Increased understanding of the role of science and technology in solving real-world problems [1],
- Increased interests in pursuing degree/career in technical, math, or science related field [2],
- Increased understanding of the value of working in teams [2]
- Increased self-confidence [2]
- Enhanced learning on physics, programming, mechanical engineering, electronics, and science [6]
- Enhanced skills of communication, team work, and personal development [6]

These positive impacts can be seen without geographical or cultural barriers. RoboCupJunior, which we believe to be the most international of the events referred, typically receives teams from all the continents and from countries as diverse as the USA, Brazil, Portugal, Germany, Turkey, Israel, Iran, China, Japan and Australia, to name a few. Moreover, it also caters for a broad range of challenges and technical requirements, offering both entry-level competitions that impose low costs and involve low complexity, as well as competitions for advanced students that promote technological research and development, which can be continued in the RoboCup leagues.

In this paper we start by introducing RoboCupJunior, and then we present preliminary research on the impact of RoboCupJunior competition on participating students' STEM education using the latest feedback gathered during RoboCup 2012 in Mexico City.

RoboCupJunior

Studies on educational robotics competitions highlight the benefits that such competitions provide to participating students. However, RoboCupJunior (RCJ) stands apart from other educational robotics initiatives for several reasons. First, its goals remain approximately the same from one year to the next, providing a scaffolded learning environment in which students continuously develop and sophisticate their solutions as they grow and expand their skills and knowledge. Second, it focuses more on *education* than competition. Although the goals remain the same each year, the rules are improved every year through intensive discussions among the technical committee members to not only improve the competition itself but also to improve the learning experience that participating students can have. Third, its challenges, called leagues, use topics – soccer, rescue and dance – that are familiar to students to attract and motivate them into educational robotics. All three Junior leagues emphasize both cooperative and collaborative nature of design, programming and building in a team setting [4].

What makes RCJ unique comparing to other educational robotics competition is its position as the entry-level to the international RoboCup (RC) initiative. RoboCup is strongly committed to research, education, and involvement of young people in technology in general and robotics in particular. The relationship between RCJ and RoboCup provides a venue for students who completed their participation in RCJ to continue advancing their skills and knowledge by participating in RoboCup's more advanced research programs.

In the following sections, the history of RCJ and the introduction to RCJ leagues – Soccer, Rescue and Dance are presented.

A. History of RoboCupJunior

The first Robot World Cup Soccer Games and Conferences were held in conjunction with IJCAI in Nagoya in 1997, with over 40 participating teams. Since the inaugural competition, RC has grown each year to involve more robotics scientists. The idea to create a league for young robotics participants was initiated by a group of researchers to respond to the need to foster robotics in the next generation of children. It was first introduced in 1998 as a demonstration by Henrik Hautop Lund and Luigi Pagliarini with robots playing soccer [4]. After further planning, a pilot project was implemented at RoboCup Euro 2000, with twelve participating teams with a total of 50 students, ages 13 to 16. They developed soccer robots to play one-on-one robot soccer games [4]. Also in 2000, the first RCJ competition was organized during RoboCup, Melbourne, 2000, with 25 participating teams from three countries. There were three challenges (leagues) – Dance, Sumo and Soccer. The success of the first RCJ competition led to creation of subsequent annual RCJ International competitions. In 2001, RCJ International was held in Seattle, USA. There were 25 teams with 104 participants, including both students and mentors. They represented four countries: Australia, Germany, the UK and the USA. In this year, Rescue league was introduced which replaced the Sumo challenge, leading to the current three leagues – Soccer, Rescue and Dance.

Since 2000, RCJ has grown to be a very popular educational activity for school age children in many countries from around the world. In recent years, there are between 200 to 300 teams competing at the Annual RCJ International competition. In 2011, the Annual RCJ International competition was held in Istanbul, Turkey, with a total of 251 teams comprised of 955 students from 30 countries. In 2012, a decade later since its initiation, the Annual RCJ International competition was held in Mexico City, Mexico. There were 209 teams participating with 796 students from 27 countries. The number of participating teams changes every year with RCJ due to the size of the venue where the competition is held. But the overall population participating in RCJ activity around the world is growing with more countries willing to start RCJ initiatives. As of March 2013, there are 37 countries reporting participation in RCJ initiatives.

Fig. 1 RCJ Soccer

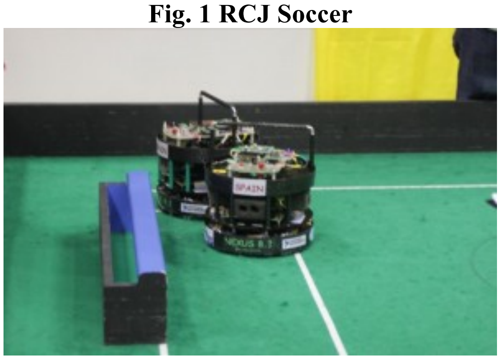


Fig. 2 RCJ Rescue

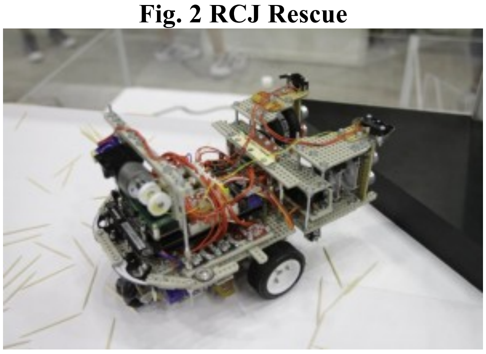


Fig. 3 RCJ Dance

B. RoboCupJunior Leagues

As referred before, RoboCupJunior currently offers three leagues – Soccer, Rescue and Dance (Figures 1-3). The RCJ Soccer league was created after the RoboCup Major Soccer leagues, being considered an entry level. The size of the robots is similar to that of Small Size League robots. The Junior soccer usually attracts boys. The RCJ Rescue league was inspired by the major Rescue league and is also considered to be an entry level for that league. Initially, the Junior Rescue generally attracted boys. However, in recent years, there are more girls involved in the Junior Rescue than before. On the contrary, the RCJ Dance league was created with the aim to attract girls into STEM and robotics by combining art with robotics. From 2013, RCJ CoSpace Rescue and Dance were introduced as sub-leagues of Rescue and Dance. Junior CoSpace was introduced in RCJ 2010, combining virtual simulation robotics with real robotics.

There are two age categories under each sub-league that are set by the RCJ general rules. Primary category is for students up to 14 years old who can construct and program a robot on their own (without adult assistance). Secondary category is for students ranging in age from 15 to 19 years. However, some of the sub-leagues do not specify age category (open age category), typically involving more advanced challenges that, nevertheless, do not exclude advanced younger students that wish to try their skills against those of older ones. All the rules for RCJ leagues in 2013 can be found in [8].

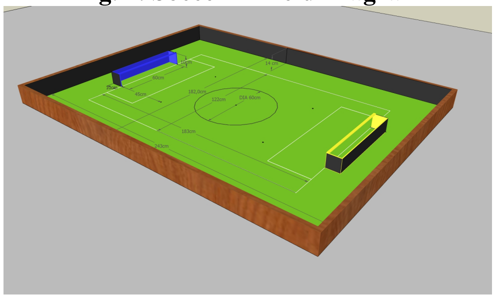
a) RoboCupJunior Soccer

The Soccer league was inspired by Lund's demonstration and the major Soccer leagues. Two teams of two soccer robots (2-on-2) play on a special field. During a game, the robots are programmed to detect and maneuver a soccer ball emitting infrared light.

There are two Junior Soccer sub-leagues – Open League and Light Weight league. With the Open league, the maximum weight of a robot is 2,400g, whereas with the Light Weight league, the maximum weight of a robot is 1,100g.. The latter sub-league is further divided by age of team members.

From 2013, both soccer leagues use the same field. The size of the field is 122cm x 183cm plus an additional 30cm outer area, which is surrounded by walls (Fig.4). The floor of the field is covered by the same green carpet that is used by the major Soccer league. Since the ball can easily go outside the field limits, the robots need to use fine controls more than just speed. Moreover, the goals are painted in different colors to facilitate the transition to vision-based sensing for more advanced teams.

Fig. 4. Soccer B Field Diagram



The Soccer league uses all age-categories to promote different levels of learning experience among participating students. The Light Weight Soccer league has primary and secondary age categories as their sub-leagues. The Open Soccer league is open to any age up to 19 years old (the maximum age for participating in RCJ). Table 1 shows all Junior Soccer sub-leagues for RCJ 2013.

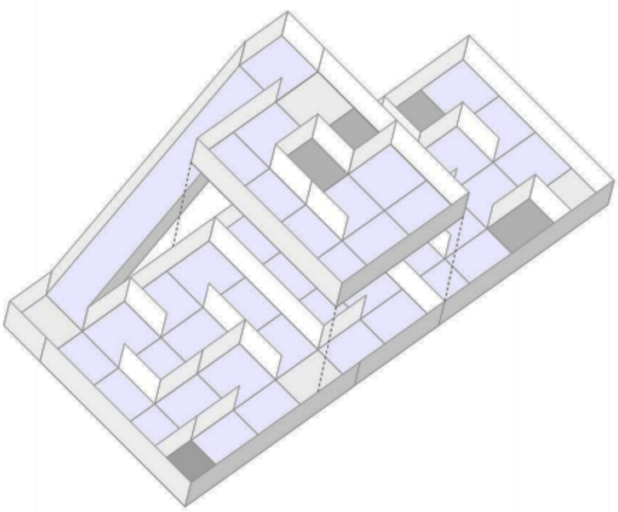
Table 1. Soccer Sub-Leagues

Sub-Leagues	*Age Categories*
Light Weight Soccer Primary	Primary Age
Light Weight Soccer Secondary	Secondary Age
Open Soccer	Open up to 19 year old

b) RoboCupJunior Rescue

Inspired by the RoboCup Major Rescue league, the RCJ Rescue was implemented in 2001. The RCJ Rescue requires teams to develop a rescue robot that can navigate through the rescue arena (Fig. 5 shows Rescue B arena) which represents a scaled-down, simulated disaster scenario, and find victims. The Junior Rescue league has two sub-leagues – Rescue A and Rescue B. With Rescue A, which has games for Primary and Secondary age categories, teams use line-following strategies to navigate through the rescue A arena where debris and obstacles are scattered, possibly blocking the line. In 2013, a new challenge has been added with which the robot needs to climb up to the upper floor and down from it using a ramp before the final mission to rescue a victim in the final room. The victim is considered *rescued* when it is moved into the evacuation zone. For primary teams, the robot needs to push the victim into the evacuation zone, while with secondary teams the robot has to pick up and move the victim into the evacuation zone. In 2011, Rescue B sub-league was officially added aiming to provide challenges for more advanced teams. Rescue B is open to any age up to age 19. With Rescue B, a robot needs to navigate through a maze using wall following algorithms. The same as Rescue A, there are debris and obstacles which the robot needs to either go over or move around to avoid a collision. Victims for Rescue B emit heat and are scattered across the arena. The robot's mission is to rescue the heat generating victims by indicating their locations.

Fig. 5. Rescue B arena diagram



c) RoboCupJunior Dance

The Junior Dance league, one of the original Junior leagues since 2000, attracts more girls than the other two leagues because of its focus on combining arts and technology. The RCJ Dance league is more open-ended than other RCJ leagues in terms of the size and kind of the robots that teams can create. It has no size or number limit as long as they stay on the 6m x 4m stage performance area (Fig. 6).

Fig. 6. Dance performance

A dance team can build a robot or multiple robots that move to music which is 2 minutes or less in duration. The creative and innovative presentation and performance of robot(s) is emphasized in the Dance league. The Dance league was introduced as an entry-level event that focused on primary school children in 2000. However, the dance performances have gained in complexity over years. The robotic dance performances have become more advanced in construction and programming. For the assessment of robotic performances, score sheets are used as rubrics. This helps teams to understand what is required to make their robotic performance successful. The score sheets emphasize the demonstration of creativity, innovation, taking risks with complicated or advanced programming and construction, and creative use of different sensors. Part of the assessment process, all teams are interviewed by a set of technical judges including technical and organizational committee members. The Dance league also divides teams into the age categories (primary and secondary).

d) RoboCupJunior CoSpace

CoSpace Robotics [7] uses the technology of *Coexisting Space* where both physical and virtual worlds communicate and interact. CoSpace Robotics combines and connects robots in a real, physical space with robots and the entities in 3D virtual-reality world. It allows students to experience and interact with robots not only in the real world but also in virtual reality that is based on the physical model in the real world.

Since 2013, the RCJ CoSpace was integrated in Rescue and Dance leagues as their sub-leagues. The Junior CoSpace Rescue has a specific theme each year. Each team has to develop appropriate AI strategies for a virtual robot to navigate through the treacherous terrain by avoiding obstacles and collect treasures in the 3D virtual environment while competing against an opponent robot performing the same mission. Next, the teams applied the same AI strategies to the identical real robot to search the treasures in the real world with the same set-up of the virtual arena. With the Junior CoSpace Dance, teams have to develop robots (both real & virtual) to create a performance in a co-existing space. It is a requirement for teams to establish a communication between real and virtual robots.

EDUCATIONAL IMPACT OF ROBOCUPJUNIOR

A. Learning from RCJ 2004 Study

Since the inception of RCJ in 2000, we have conducted surveys and interviews with some of the students and mentors who participated in the international event. The analysis of the collected data was presented in 2004 [3]. The focus of the study was to distinguish overall impact of RCJ on students' learning in various areas, which include various subject areas/content knowledge, and personal skills including communication, collaboration, problem-solving skills.

The study shows the rapid growth of RCJ popularity in participating countries. In 2001, there were only 25 teams from Australia (10 teams), the US (8 teams), Germany (5 teams) and the UK (2 teams). The total number of teams participating doubled in 2003 with the total of 57 teams from fifteen countries. This has continued to grow each year. For the evaluation of the educational value, we used 5-point Likert scale to identify educational impacts on the participating students. The study shows that more than 50% of the students who responded to the survey in 2003 indicated that, through participating in RCJ activity including their preparation period, they gained subject matter knowledge in physics, programming, mechanical engineering, electronics, and science in general, and developed skills in communication, collaboration and in other areas of personal development [3]. Their positive responses were especially higher in the area of mechanical engineering and electronics. This is mainly because many of the teams participating in the international competition do not simply use off-the-shelf robotics kits such as LEGO Mindstorms and Fischertechnik robotics kit. Rather, they learn to modify the kits and/or design and develop their robot from scratch. As indicated previously, since RCJ does not restrict teams to use a certain robotics kit or materials, it provides more freedom for participating students to learn to use robotics materials of their choice, and continue advancing their knowledge and skills every year.

B. RCJ 2012 Study

In 2012, another survey was conducted with students participated in RCJ 2012, Mexico City.

PARTICIPANTS: There were 209 teams with 796 students participating from 26 countries in 2012. The breakdown by league is Soccer – 79, Rescue – 87, and Dance – 43 teams. The survey was distributed to the teams randomly during the competition. The participation in the survey was voluntary. The survey was collected in a box at the RCJ organizing committee office. There were 168 students participating in the survey, which is 21% of RCJ participants. The participants of the survey were from 19 countries, which represent 73% of the participating countries. The age of the students ranged from 10 to 19 year old. The majority (92%) of the students are 13 to 18 year olds. Eighty percent (80%) of the participating students were male. Among those, 64% responded that it was their first participation in the annual international RCJ competition. Among the 58 students who have participated in the international RCJ competition in the past, 22 students (13%) participated more than once in previous years. Among those who participated in the survey, 73% participated in Soccer competitions, while 27% participated in Rescue competition. None of the students participated in Dance or CoSpace competitions. Twenty-four students (14%) indicated that they have tried other leagues as well. This shows that the majority of the participating students continues with the same league that they started with.

RESULTS: The analysis of the data indicated the following results.

- *Do you like STEM subject?:* The survey asked if the participating students like Science, Mathematics, Engineering, Electronics, or Programming. Overall, around 90% of the students agreed that they like the STEM subjects (Science, 98%; Math, 93%; Engineering, 93%, Programming, 89%; Electronics, 92%).
- *Did you like STEM subject before participating in RCJ?:* The majority of the participating students did like STEM subjects before their participation in RCJ (Science, 96%; Math, 86%; Engineering, 83%; Programming, 69%; Electronics, 75%).
- *Did RCJ help you to enjoy learning the STEM subjects more?:* Again, the majority of students agree that RCJ did help them enjoy learning the STEM subjects (Science, 89%; Math, 70%; Engineering, 91%; Programming, 86%; Electronics, 90%).
- *Do you consider going to college/graduate school?:* The majority of the students agree that they are interested in post-secondary education; however, considering participating in a graduate school is less than in college (College, 84%; Graduate School, 69%). The majority of participating students consider STEM fields as their future majors (Science, 76%; Math, 68%; Engineering, 71%; Programming, 68%; Electronics, 68%; Robotics, 74%). Their interests in Science, Engineering and Robotics are a little higher than other STEM subjects.
- *What did you learn the most from RCJ experience?:* This question has an option for participating students to mark all applied. The results show that more than half of the students agreed that they learned the most in 1) programming, 61%; 2) teamwork/collaboration, 54%; 3) electronics, 52%; and 4) communication skills, 52%.
- *Will you participate in RCJ again?:* 82% of the participating students answered positively to this question.

DISCUSSION

The results show that the students who participated in RCJ 2012 like STEM subject areas. Robotics in general attracts students who are interested in STEM. The result is not surprising; however, almost 100% of the students who participated in the survey indicated that they like Science, which is notable. When comparing the first and second questions, the results indicate that participating in RCJ increased their interest in the STEM subjects. In addition, the impact of participating in RCJ on their interests in STEM subjects is shown to be strong. Although RCJ competition attracts students who are already interested in STEM areas, the results show that it has some positive impacts on the participating students' interests in the STEM fields. The results on the students' future education choice show that the students are interested in moving on to post-secondary education; however, they are not as many as we expected to see. This is possibly because the results include rather young students who might not be able to perceive their future at this point. When we analyzed the data from secondary students (15 to 19 year olds), that number is much higher. The students who consider going to college went up to 89%. With graduate schools, 80% of the secondary age students (11% increase) indicated their interests in studying at graduate schools, which shows a positive impact of their educational robotics experience.

Although the participating students indicated that their learning from the participation in RCJ is stronger in the areas of engineering, science and electronics, they highlighted their learning in programming more when asked what they learned the most. Also, it is interesting to point out that they highlighted the skills in teamwork/collaboration and communication. Educational robotics, especially through robotics competitions, is known to enhance students' learning of teamwork/collaboration, as well as communication skills. This is because of the nature of educational robotics activities that focus on group work and project-based learning [9].

CONCLUSION

Educational robotics is proven to promote STEM interests among students from around the world. The hands-on learning experience that it provides has long-lasting impacts on student learning and motivation for further exploring related fields. RCJ has committed to promote educational robotics learning experience among students from around the world. The self-study conducted in 2012 has enhanced the results from previous studies and provided stronger evidence that RCJ has positive impacts on participating students' STEM learning.

To advance our mission, RCJ alongside with RoboCup, is dedicated to collaborate with organizations/schools interested in promoting RCJ in their country. We believe RCJ can provide valuable impacts on the education of next generations, by promoting technological awareness and capabilities.

REFERENCES

[1] FIRST: About FIRST - Impact, Retrieved from http://www.usfirst.org/about/impact.htm , 2006.

[2] KISS Institute for Practical Robotics: Botball Educational Robotics, Retrieved December 11, 2010, from http://botball.org/about, 2009.

[3] E. Sklar, & A. Eguchi, "RoboCupJunior - Four Years Later," *Proceedings of RoboCup-2004: Robot Soccer World Cup VIII,* 2004.

[4] E. Sklar, A. Eguchi, & J. Johnson, "Children's learning from Team Robotics: RoboCupJunior 2001," Proceedings of RoboCup-2002: Robot Soccer World Cup VI, 2002.

[5] E. Sklar, A. Eguchi, & Johnson, J, "Examining the Team Robotics through RoboCupJunior," Proceedings of the Annual Conference of Japan Society for Educational Technology, Nagaoka, Japan, 2002.

[6] E. Sklar, A. Eguchi, & Johnson, J, "Scientific Challenge Award: RoboCupJunior - Learning with Educational Robotics," AI Magazine, 24(2), 43-46, 2003.

[7] CoSpace Robotics, Retrieved March 31, 2013 from http://cospacerobot.org/

[8] RoboCupJunior 2013 rules, Retrieved March 31, 2013 from http://rcj.robocup.org/rcj2013/

[9] A. Eguchi, "Educational Robotics Theories and Practice: Tips for how to do it Right," in B.S. Barker, G. Nugent, N. Grandgenett, V. Adamchuk (eds.), *Robotics in K-12 Education: A New Technology for Learning*. Hershey PA: Information Science Reference (an imprint of IGI Global), 2012.

Lessons Learned in a Ball Fetch-And-Carry Robotic Competition

Marco Cigolini, Alessandro Costalunga, Federico Parisi, Marco Patander,
Isabella Salsi, Andrea Signifredi, Davide Valeriani, Dario Lodi Rizzini, Stefano Caselli
RIMLab - Robotics and Intelligent Machines Laboratory
Dipartimento di Ingegneria dell'Informazione, University of Parma, Italy
E-mail {dlr,caselli}@ce.unipr.it

Abstract— **Robot competitions are effective means to learn the issues of autonomous systems on the field, by solving a complex problem end-to-end. In this paper, we illustrate Red Beard Button, the robotic system that we developed for the Sick Robot Day 2012 competition, and we highlight notions about design and implementation of robotic systems acquired through this experience. The aim of the contest was to detect, fetch and carry balls with an assigned color to a dropping area, similarly to a foraging navigation task. The developed robotic system was required to perceive colored balls, to grasp and transport balls, and to localize itself and navigate to assigned areas. Through extensive experiments the team developed an initial prototype, discovered pitfalls, revised the initial assumptions and design decisions, and took advantage of the iteration process to perform successfully at the competition.**

I. Introduction

Robot competitions constitute an effective mean in robotic education [1], [2]. Through the contest students can learn to address robotic problems and tasks, to work as a group, to design complex systems including mechanical structure, electronic components and software architecture, and to check the initial assumptions with the results on the field. In common robotic practice as well as in student projects, researchers and students tend to concentrate on specific aspects of robotics such as perception with a specific sensor, localization or navigation. Thus, the main result is a single component or an algorithm, whose experimental assessment is usually accurate but aims at achieving proof-of-concept and sometimes artificial demonstrations. On the other hand, solutions developed during a robotic competition must be effective and take into account the interaction of each component with the whole robotic architecture. A method that works correctly in laboratory experiments may not achieve the same results when used in different setups like those involved in a competition. Thus, students can learn through competitions that "the whole is greater than the sum of its parts" as well as appreciate the importance of tests on the field.

Sick AG, a leading manufacturer in sensor technologies and laser scanners, organizes *Sick Robot Day*, a competition open to student teams from universities and other educational institutions aimed at promoting mobile robotics and automation technologies in education. In 2012 Sick Robot Day reached its fourth edition. While previous editions involved perception and navigation capabilities, in the latest challenge the robots were required to detect, fetch and carry balls with an assigned color to a designated area called pen. The proposed problem falls in the well-studied category of the foraging tasks [3]. The contestants had to address several problems including which sensors to use for detecting balls, obstacles and pen, how to carry the balls, how to find the pen, and which tasks to execute. The robot systems were developed by participating teams under imperfect knowledge of the final competition environment, shown in Figure 1.

Fig. 1. The arena of Sick Robot Day 2012 delimited by a fence and three pens. A pen is shown in the bottom-left.

In this paper, we illustrate the robotic system implemented for Sick Robot Day 2012 by a team of students of the University of Parma and the lessons learned during its development. The implementation of the control architecture required the team to make design decisions and to verify the obtained results on the field. Experiments have proven fundamental for discovering pitfalls and for developing more robust and effective solutions. The robotic competition has proven a valuable experience to check initial assumptions and to learn how to implement components that can perform the required tasks in practice. The final autonomous system has proven quite effective and our robot, *Red Beard Button*, achieved first place at the competition.

The paper is organized as follows. Section II summarizes the competition rules. Section III illustrates the architecture of Red Beard Button and shortly describes the development history. Section IV illustrates the experiments performed before the competition and the problems met. Section V discusses the lessons learned in this experience, while section VI provides the concluding remarks.

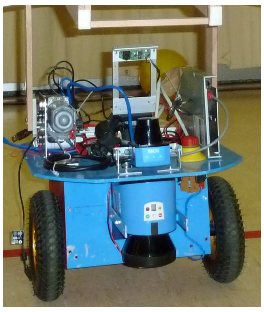


Fig. 2. The robot equipped with Sick LMS100 and TiM300 laser scanners, Logitech C270 camera, and the motorized fork lift.

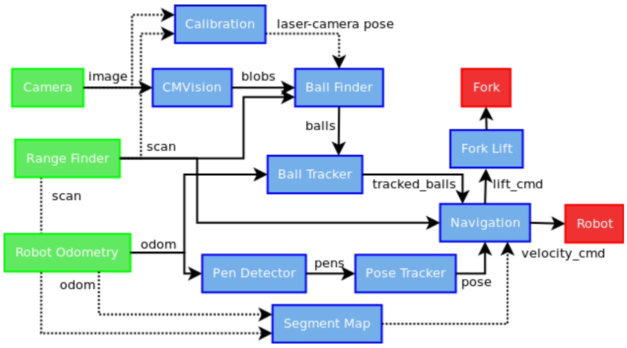


Fig. 3. The robotic architecture of the system composed of ROS framework nodes.

II. Competition Rules

This section summarizes the rules of Sick Robot Day 2012 in order to clarify the design decisions to the reader. The contest takes place in an indoor polygonal arena, whose diameter size is about $10 \div 20$ m. The arena contains balls of three different colors with $20 \div 25$ cm diameter. The ring fence of the arena gaps in three zones where three pens are placed. Each pen is distinguished by one of the three colors and is used as a starting position for one of the robots and as the ball dropping area.

The aim of challenge is to detect, fetch and carry to the pen as many balls of the assigned color as possible. The contest consists of several 10 minutes rounds (also called runs) and three robots compete at the same round, each looking for balls of a given color. Each robot participates to two rounds and a different color is assigned in the two rounds. The score of each round is equal to the number of balls of the assigned color, except for penalties. The balls of a wrong color reaching the pen are subtracted from the score of the round. Furthermore, every contact of the robot with the fence is sanctioned with a half point and collision with another robot leads to instant disqualification from the current round. Contact with balls is allowed irrespective of their color. Thus, the position of the balls is likely to change during a run since robots may carry or push them. The final placement of the teams depends on their best performance in either of the two rounds. Several details, like ball colors, exact dimensions of the balls and of the pen, or number of balls placed inside the arena, were not defined by the rules of procedure and have been discovered by teams with short notice or the very day of the competition.

III. Robot Architecture

In this section, we present the final architecture of the Red Beard Button robot implemented for Sick Robot Day 2012. We also briefly discuss the variants implemented before reaching the final one and the motivation for the design decisions. The system has been decomposed into parts to address the main three challenges given by the competition: ball detection, ball picking and transportation, and robot localization for returning to the pen. These three tasks are coordinated by the robot navigation system.

The robotic platform used in Red Beard Button is a MobileRobots Pioneer 3DX equipped with two laser scanners, Sick LMS100 and Sick TiM300, and a Logitech C270 camera (Figure 2). The scan plane of the LMS100 laser scanner is approximately parallel and 10 cm above the ground plane. The TiM300 laser scanner has been included in the architecture to overcome ball occlusion problems. However, it has hot been used in the final robot setup due to design decisions discussed later in the paper.

The perception component detects the balls of the required color by performing sensor fusion. The device adopted for carrying balls is relevant for the navigation strategy. Two ball picking structures have been implemented: a simple static fork, that requires specific navigation policies to avoid loosing the ball, and a motorized fork, that lifts and cages the ball thereby avoiding any occlusion in front of the robot. A localization and mapping algorithm is required to estimate the robot position w.r.t. the pen area where the ball must be dropped. Since the map of the environment is unknown, the robot must extract landmarks to find its position. The only stable elements in the given competition arena are the fence and the pens. Finally, the navigation component handles the robot task state and coordinates perception and action using the information provided by the other components. The different tasks have been implemented as ROS[1] nodes and are illustrated in Figure 3. In the following the details of different components are described.

A. Navigation

The navigation component is responsible for the execution of robot motion and for the management of the state of competition. The navigation task coordinates all the other tasks, since it receives and uses their outputs to carry out robot main task. In the arena, the robot interacts with different kinds of objects:

- *static objects* like arena fence, that must be avoided in order not to incur into penalties;
- *semi-static objects* like balls, that may be moved or avoided depending on the adopted policy;

[1]ROS (Robot Operating System - http://www.ros.org) is an open-source meta-operating system for robots.

- *dynamic objects* like the other robots, that may lead to disqualification if a collision occurs.

The presence of several dynamic and semi-static objects in the arena makes path planning an ineffective solution, since a plan may quickly become outdated due to the change in obstacle configuration. Thus, a reactive approach has been preferred for robot navigation. The development of navigation components has been simplified by the choice of the motorized fork lift that is discussed in section III-C. The navigation task is divided into several subtasks, each corresponding to a robotic behavior with a specific goal:

- *exploration*: the robot moves and searches target balls;
- *ball approaching*: when a target ball has been detected, the robot approaches it;
- *ball grasping*: the robot reaches the ball and raises the fork;
- *transportation*: the robot returns to the pen to drop the ball;
- *ball release*: the ball is released into the pen.

Figure 4 illustrates the flowchart of navigation decomposed into subtasks.

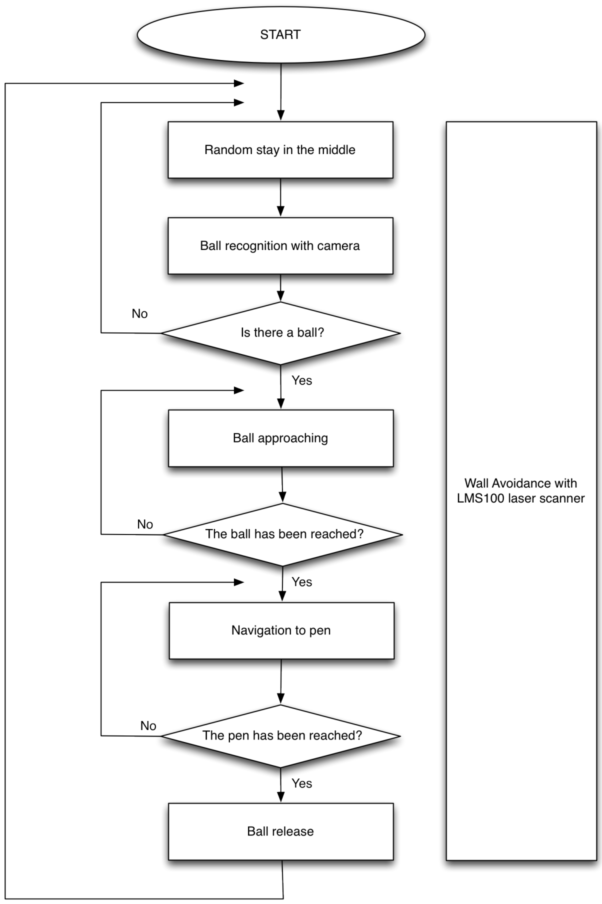


Fig. 4. Flowchart of navigation decomposed into subtasks.

Safe navigation is guaranteed by a collision avoidance behavior, which interrupts the execution of current subtasks when the distance from the closest obstacle is less than a given threshold ($0.55\ m$). When collision avoidance is active, the robot steers in the opposite direction w.r.t. the obstacle until free space is observed in front of the robot. Such behavior is disabled only during the approach to or the release of a target ball.

The exploration task has been developed using a hybrid approach: the main behaviour is a standard *stay-in-the-middle* behavior [4] that allows the robot to move in the environment keeping about the same distance from the nearest obstacles on its left and on its right. In order to move to all the directions and explore the environment, every 12 seconds the robot randomly steers. During exploration, the robot speed may reach $0.45\ m/s$ and the fork lift is held raised in order not to occlude the laser scanner.

When the ball detector component observes a target ball, the ball approaching behaviour is activated. Then, the mobile robot rotates towards the centroid of the ball and moves with a speed proportional to the ball distance. If the ball is lost, e.g. the collision avoidance switches on, the exploration task is reactivated to search and reach other interesting balls. However, the ball tracking module described in the following avoids intermittent observations of the goal and prevents unnecessary transitions between ball approaching and exploration.

When the distance to the ball is less than a given threshold (about $0.70\ m$), the fork is lowered and ball grasping task is performed. During ball grasping, perception of the target balls and obstacles is handled by a specific procedure due to the limited field of view of the camera, which prevents the observation of balls, and the occlusion of the laser scanner caused by the lowered fork. The robot moves towards the ball until it correctly grabs the ball or fails. The outcome of such operation is monitored by a selected subset of frontal range finder beams that are not occluded. When the ball is caught, the robot raises the fork and starts to navigate towards the pen. Otherwise, after having lifted the fork, the robot resumes exploring the environment. Since the ball is caged by the fork, the ball never falls down during the lift.

The navigation back to the pen is driven by the information provided by the localization module. This subtask directs the mobile robot towards a goal point placed in the middle of the pen, setting the orientation properly to approach the pen frontally. In order to prevent collisions, the *collision avoidance* behavior runs in background with higher priority. Moreover, when the robot is near to the pen ($1.2\ m$) the linear velocity is reduced to $0.2\ m/s$ to perform a more accurate motion.

When the final position is reached with the right orientation, the *ball releasing* task is activated. After lowering the fork, the robot pushes the ball in the pen moving forward and suddenly backward. If the ball is correctly released, the robot rotates around its axis about $180°$ and restarts the exploration of the arena to search another ball of the assigned color.

B. Ball Detection

The main task of the detection module is to distinguish the target balls from all the other objects placed in the arena. Therefore, during exploration the robot must be able to segment its sensor measurements and extract those segments that meet the requirements of goal objects like shape, aspect ratio, size, colour and a position consistent with physical constraints (e.g. balls lie *on* the ground). Since two different types of sensors, namely a RGB camera and a laser scanner, are available, recognition of candidate target balls is separately performed in the two sensor domains (laser scans and images) and the results are associated only in a second phase. In this way, the algorithm takes advantage of both devices and, at the same time, processing can be performed by two separate components. The laser scanner provides an accurate estimation of ball position, while the camera is able to assess the color and the aspect ratio of the region-of-interest (ROI) corresponding to balls.

The robot control application, developed for the ROS framework, consists of four nodes. The first node is the *CMVision* package (Color Machine Vision project) [5] that extracts blobs of a given color from the frames acquired from the camera. Since the segmentation of images is independent from the laser scanner, it has been easy to integrate this library package into our system. The second node is dedicated to the calibration procedure, which is performed only offline before using the detector. The third node is the ball detection core component, which processes laser scans and associates laser segments to the color blobs extracted by CMVision. The fourth node is a *ball tracking* node that addresses the intermittent detection caused by laser scan and image segmentation failures or by missing associations between the two sensor domains.

The purpose of the calibration node is the estimation of the transformation matrix between a point P_{laser} in the laser reference frame and the corresponding point P_{img} in the image plane and viceversa as expressed by equation

$$P_{img} = KK \cdot {}^{C}_{L}T \cdot P_{laser}$$

where KK is the intrinsic parameters matrix of the camera and ${}^{C}_{L}T$ the transformation matrix from laser frame to camera frame. While there are several packages for estimating KK, the few libraries for assessing ${}^{C}_{L}T$ strongly depend on the setup and the calibration object. The calibration object must be chosen so that it is possible to detect and match a pair of homologous points in the two sensor domains. We have investigated the algorithm proposed in [6] that jointly calibrates a laser scanner and a camera by matching slices of a planar checkerboard with the plane of the same checkerboard. Unfortunately, we have not achieved satisfactory results, possibly due to the noisy perception of the checkerboard or to numerical stability problems of the proposed method.

Thus, we have implemented an iterative procedure based on the manual association of the measurements of a ball acquired with the laser scanner and the camera. Although not automatic, this method allows quick and reliable estimation and has the advantage of using the object to be detected (the ball) as a calibration target. This method exploits the same segmentation procedures of the image and of the laser scan used during detection. However, since the algorithm starts from an initial guess of the transformation ${}^{C}_{L}T$ to be estimated, the blobs returned by CMVision are filtered according to strict criteria on the area and aspect ratio of the balls. Then, the centroids of the laser segments are projected into the image plane according to the current value of ${}^{C}_{L}T$ and roughly associated with the blobs. The user can iteratively change the values of translation and rotation parameters of ${}^{C}_{L}T$ until the projected laser points overlap with the centroids of blobs.

After the initialization of parameters, the detection cycle consists of four steps:

- segmentation of laser scan using a discontinuity threshold and selection of intervals checking their diameter;
- projection of these valid segments in the image frame;
- if a segment falls into a bounding box, it takes on its colour and it is classified as belonging to a ball;
- publication of the recognized balls list, including useful information for navigation and collection, such as colour or position in the laser reference frame.

The tracking node has been designed to address intermittent detection of balls due to temporary failure of the ball detector illustrated before. The node keeps an estimation of the observed balls by updating their position w.r.t. the robot according to robot odometry and the sensor observations. The tracking algorithm implements Kalman filter equations. Objects that have not been observed for a given time interval, are removed from the state.

Tests in the laboratory, with controlled light, have shown that the algorithm is able to identify and locate with satisfactory accuracy all the balls. The association is correct, even though the calibration is performed with the manual algorithm. However, larger environments with reflections and abrupt light changes strongly affect the performance of the CMVision component. The problems of this component are further discussed in section IV.

C. Ball Grasping Device

An important requirement to succeed in the competition was to provide the robot with a device to move the balls that are inside the arena. Among several possible solutions, we have built a static fork and a motorized fork lift. The first device consists of two plain wooden bars that can be used to push the target ball as shown in Figure 5(a). This device requires the availability of an additional laser scanner at a different height (in our case the TiM300) since the LMS100 is occluded during ball transportation. The second device is a motorized fork lift, shown in Figure 5(b), that can raise the ball when it has been caged among the fork bars. Since the fork is raised during exploration and ball transportation, the laser scanner is occluded only during ball grasping and release.

We experimented with both solutions until few weeks before competition. The static fork was appealing for its simplicity in construction and reliability, but ball transportation proved difficult since the ball was not caged. The fork lift required some iterations in mechanical and electronic design and was eventually preferred for the competition. Indeed, with the fork lift the robot does not lose the ball while

(a)

(b)

Fig. 5. The static fork (a) and the motorized fork lift (b) built to cage and carry balls.

moving because the ball is well caged without occluding the sensor.

The construction of the motorized fork lift requires a mechanical structure, an electric motor, the electronic components for its control, and a software interface with the laptop computer. The system is composed by the following components:

- a DC geared motor with a high reduction ratio, so as to decrease the maximum speed and increase the torque output;
- a *Microchip Technology Inc PICDem2* board, which consists of a microcontroller, the output interface with the powerboard, an Ethernet port and other elements not used in this project;
- a power board, built in the university laboratory, which controls the power supply of the motor according to the logic signals output from the PICDem2 board;
- two limit switches, which signal when the fork is completely raised or lowered.

The limit switches are the only devices available to monitor the fork state. No other information is available while the fork is in an intermediate position.

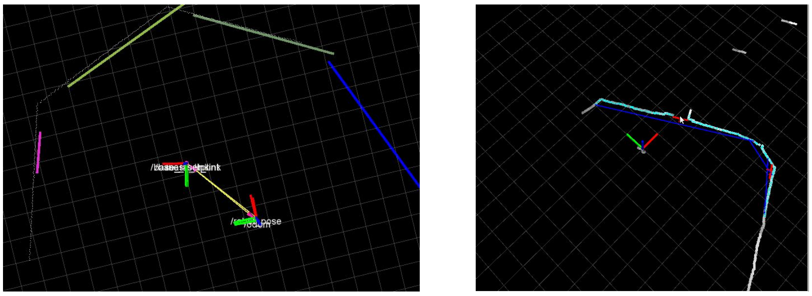
A ROS node is responsible for the communication between the laptop computer and the control board through a custom protocol on TCP/IP port. The microcontroller waits for commands from the computer and sends control signals to the motor when it receives a command. To control the motor, the board generates a PWM modulation: a pair of square waves, one opposite the other, are generated and overlapped into a single signal to the motor. The amplitude of the signal is 12 V. The final performance of the system is satisfactory, since the fork reliably raises and releases balls.

D. Localization and Mapping

Localization is a crucial task for the successful accomplishment of the proposed challenge. When a ball is fetched using the fork lift, Red Beard Button must reach its pen and drop the ball there. Without knowing its pose, the robot cannot plan its path or even guess the direction toward the pen. The information provided by odometry is unreliable, since odometry is sensitive to steering and its error increases with the travelled path length. In order to estimate its own position and orientation, the robot requires a map containing the landmarks or implicit references that can be easily detected in the environment. When such map is not available, the system must be able to build a map from the acquired measurements. This problem has been investigated by robotic research for decades and is known as *simultaneous localization and mapping* (SLAM) [7].

In the scenario of the Sick Robot Day 2012 competition, a major complication is represented by the lack of stable and continuously observable landmarks. The arena shown in Figure 1 chiefly consists of balls, whose position rapidly changes and which occlude the border of the arena. The fence and the pens, which appear as gaps in the fence, are the only invariants in the scene. Both types of candidate landmarks are distinguishable in laser scans by detecting aligned points. Two different approaches have been developed for map construction and localization, each using one of the two landmarks. Figure 6 illustrates the output of the two methods.

The first method builds a map of segment landmarks to represent the boundaries of the arena. These boundaries do not change, but they may be occluded by other dynamic or semi-static elements of the environment like balls and other robots. The scan plane of laser scanner Sick TiM300 does not intersect the balls. Thus, this range finder can be used to extract boundary segments, although its maximum range is limited to $4\ m$. More in detail, the algorithm performs four main operations. First, the scans acquired by the laser scanner are segmented into intervals and are split according to endpoints [8]. In the second step, the parametric model of the segment and its uncertainty are computed through least square estimation within the geometric limits represented by the two segment endpoints [9]. The association between the segments and the landmarks already stored in the map is performed using the Hausdorff and the Mahalanobis distances. Finally, a Graph SLAM algorithm takes the odometric data, the previous landmarks, and the landmark measurements given by the associations to estimate the pose of the robot. The sensor model uses the *SP Map* representation [10] applied to segments. Instead of using Bayesian filtering, the map has been represented by a graphical model that encodes the constraints between the variables of the problem. The estimation has been performed using the *G2O* library [11] for the optimization of constraint networks. Unfortunately, this promising and general approach has proven unreliable



(a) (b)

Fig. 6. Ouputs of the two localization and mapping nodes: the segment landmark graphical map (a) and the pen landmark localizer (b).

in this case due to the limited visibility of the fence, as well as prone to the numerical instability.

The second localization method, developed to address the limitation of the first solution, focuses on the detection of the pens. Although there are only three pens in the arena (one for each robot that concurrently takes part to a round) and only the initial pen is frequently observed, the detection of a gap in the fence is rather robust. Furthermore, the range finder view of the pen is seldom occluded by balls, since the robot starts with the closest balls right in front of the dropping area and progressively cleans the space. The developed method exploits the odometry to predict the robot pose and then corrects the estimation by using the landmark when available. After taking the ball, the robot tries to reach the pen assuming that it is located in the origin of the reference frame, located in the initial pose. Moreover, it activates the pen detection routine. A pen has been modelled with two segments lying on almost parallel lines with a gap in the between. The laser scanner data are used to build this model using an algorithm based on the *Hough Spectrum* and *Hough Transform* [12]. Whenever a pen is detected, the system checks whether the pen is the one assigned to the robot for the current round by computing the Euclidean distance between the pen and the map reference frame origin. If this is the case, the current estimation of the robot pose, which is updated using odometry at each iteration, is corrected according to the observation.

During the competition the second approach has been used. This approach has the advantages of being simpler, more goal-oriented and it better fits the problem. The first approach would have been more general and the provided correction potentially more frequent. However, it suffers from the inaccuracy of the fence detection with several occluding balls, from the numerical instability of segment landmarks and from the ambiguity of landmark association criteria, either based on the segment endpoint position or on the support line parameters. Moreover, the environment of the competition had a lot of balls that occluded the laser perception.

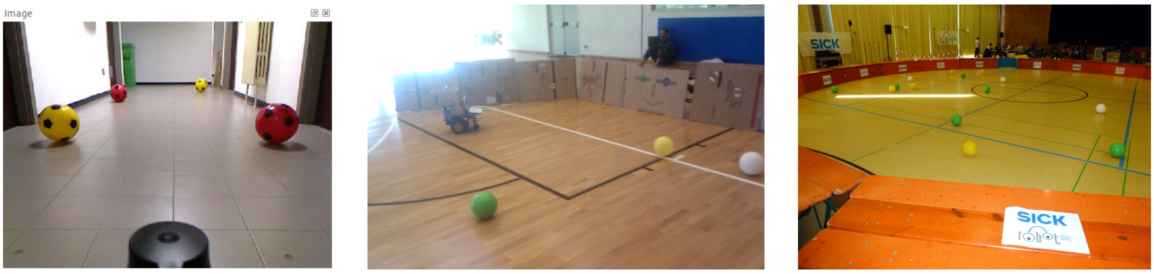
IV. Experiments

The development of the robotic architecture illustrated in the previous section has been supported by experiments in the Robotics Laboratory of the Department of Information Engineering (`lab`) and in the gym of the University of Parma (`gym`). The second environment has been chosen for its presumed similarity with the Sick Robot Day arena (`arena`). The three environments are illustrated in Figure 7. In this section, we present the experimental assessment, the correction proposed to the observed pitfalls, and the final results achieved in the competition.

A. Training Tests

The initial tests in `lab` allowed the development and fast testing of some components of the robotic architecture. In particular, the implementation of the ball detection algorithm, the fork lift and the robot navigation core have taken advantage of the laboratory test. However, only the next set of tests in `gym` allowed the full assessment and the identification of the system pitfalls. There are two main differences between `lab` and `gym`: the scale and the lighting conditions. The hallway of the department can be approximately divided into two narrow trunks, each with size about $10 \times 2.5\ m$. On the other hand, the region of `gym` used in the experiments has $18\ m$ diameter and is more similar to the competition field. Such large field does not constrain the robot motion and allows the tuning of parameters like maximum linear and angular speeds, segmentation thresholds, and pen size.

During such extensive tests, which have taken place for about a month, new problems and limitations have been detected and addressed. First, the ball detection algorithm failed when the light conditions were difficult as shown in Figure 7(b). Abrupt changes in light intensity, reflections on the ground, etc. make the color segmentation of the acquired frames unreliable. The three colors of the balls (green, yellow and white) have been announced about 2 months before the competition, when the detection algorithm had already been implemented (and team members were busy with exams and other academic duties). In order to lessen this problem, some solutions have been developed. For example, the ball tracking module described in section III-B has been applied to keep the previously detected position of balls in case of intermittent detection. The extended components worked well in the case of green and yellow balls. However, the detection of white patches in the image is unreliable when the light conditions are not fully controlled like in `lab`. This perception pitfall remained unsolved in the final competition field, since a radical change of approach and new design of



(a) (b) (c)

Fig. 7. Environments where Red Beard Button has been tested: the RIMLab Robotics laboratory `lab` (a), the gym of the University of Parma (b), and the Sick Robot Day arena (c).

the ball detection component would have been required to address it. In fact, color segmentation using an off-the-shelf component like *CMvision* has proven unreliable outside the laboratory. A customized, laser-driven approach could have been more effective.

An unforseen deadlock condition has been identified in the fork control module. In a trial, while the robot approached the ball, the fork has been lowered too early causing the block of the fork on the ball. Since the robot waits for completion of fork lowering, the system stays indefinitely in such state. A trivial solution to address such sporadic condition has been implemented by setting a timeout on the lowering action. If this action is not completed before the deadline, the fork lift is raised.

In the `gym`, the localization component has proven to be crucial for reliable robot operation in large environments. Estimation of robot pose w.r.t. the pen can be performed using only the odometry only if the size of the environment and the travelled path are limited. However, if the robot moves for 10 minutes at high speed and frequently steers, the odometric error of Pioneer 3DX largely increases and the localization of the robot becomes unreliable. In early odometry-based trials the robot missed the pen with an error up to 5 m. We then developed the two methods discussed in section III-D: localization and mapping using segment landmarks and localization using pens as landmarks. Experiments on the two methods had to cope with the limited availability of the `gym` as well as with the time pressure of the incoming competition. After some experiments in the `gym`, we adopted the approach based on pen detection, which was simpler, more robust and effective. Although only the starting pen is usually observed due to the travelled path and occlusions, Red Beard Button has always been able to reach its target configuration.

B. Competion Results

Sick Robot Day 2012 took place on October 6th in the Stadthalle in Waldkirch (Germany). Although the rule of procedure describes the general geometrical features of the competition field, the `arena` (Figure 7(c)) was seen for the first time by the 14 teams from Germany, Czech Republic and Italy only few hours before the beginning of the competition. The diameter of the real arena was about 15 m and the arena contained 29 balls for each of the three colors. The morning was devoted to setup of the mobile robot, to parameter tuning and system configuration testing whenever the field was available. Assignment of ball colors and of the rounds have been announced to the teams just before the morning trials. The competition started at 2 pm by alternating 10 rounds of 10 minutes each.

In its first round, Red Beard Button had to collect green balls. The detection algorithm has always been able to correctly identify the items with this color both during the morning tests and in the competition. In fact, during the competition the robot has collected 7 green balls in the assigned time. However, Red Beard Button hit the arena fence four times due to too low safety distance in the ball dropping phase. Hence, the final awarded score was 5, accounting for 2 point penalty assigned.

In the second round, Red Beard Button was required to collect white balls. As mentioned above, correct white ball detection was an unsolved problem. Due to the non-uniform lighting and too strong false positive control, Red Beard Button was unable to fully identify white balls in the arena. Thus, the ball detection method never estimated false positives, whereas other teams incurred in significant penalties due to the collection of balls with the wrong color.

The 5 points score achieved in the first round eventually won our team the first place in the competition, with the second and third teams obtaining 3 points and 1 point respectively. The whole system implemented in Red Beard Button has worked properly, except for the arena edge hits in the first round and the white ball detection problem in the second one.

V. Discussion

Experiments and the competition itself have allowed the team member to learn some lessons about the design and implementation of autonomous robotic systems. In the following, we propose a list of suggestions that summarize our experience.

- Perception is the most important reason for the success or failure in accomplishing a given robotic task. The correct detection of green balls has allowed the successful execution of the foraging task, while the uncertain identification of white balls within cautious acceptance policies has led to an opposite result. The interpretation of sensor measurement is critical when the decisions of the autonomous robot depend on the outcome of a classifier.

- The robotic system becomes more efficient and less prone to error when the sensor measurements are collected and organized in a coherent representation. The importance of the enviroment representation increases with the complexity of the task and the scale of the environment where the robot operates. This lesson has been proven both by the ball tracking module and by the robot global localizer. The former method is an example of short-term memory suitable to track dynamic and ephemeral objects like balls. The success of localization depends on the presence of invariant elements of the environment that can be used as landmarks.
- The complexity of the solution should be proportional to the complexity of the problem. The color segmentation used to detect balls in images has proven unsatisfactory in many cases. Such naive approach has not worked well for white balls outside the robotic laboratory, whenever the color is not an invariant property of the target objects. On the other hand, solutions like the general segment-based graphical map algorithm have proven too complex for the problem.
- Robot system development should be guided by experiments on the complete system. Each robot component has been tested in depth in the lab before the integration tests in the gym, but the problems arose only with the complete system. Unpredicted conditions may depend on the interaction between robot components and the environment: perception deficiencies may appear only when the robot (and the sensor) moves, the motion of the robot and the actuated components may be affected by objects (e.g. the fork blocked by a ball), etc. Furthermore, the experimental setup should be as similar as possible w.r.t. light conditions, dimension, etc. to the environment where the task must be performed. Of course, experiments are time consuming and the complete system is not available until the development reaches an advanced state.
- Robot developers often design and implement the system under uncertain information and cannot control all the possible conditions. For example, the color of the balls was not initially known and the ball detector has been designed without exploiting such information. Moreover, the high density of balls in the competition arena, which could be critical for a planner, was apparent only the day of the competition. Several critical conditions arose only during the last extensive experiments. Thus, the only possible countermeasure is to arrange multiple solutions to address the same task and to anticipate the criticalities by performing experiments in difficult environments. Indeed, we developed two ball carrying tools and two localization methods, and for each feature the most effective approach has been selected.

VI. Conclusion

In this paper, we have presented Red Beard Button, a robotic system designed for the Sick Robot Day 2012 competition, and the lessons learned during its development. The aim of the contest was to detect, fetch and carry balls with an assigned color to a dropping area, similarly to a foraging navigation task. The developed robot system consists of several software and electro-mechanical components to perceive colored balls, to grasp and transport balls, and to localize the robot and navigate to assigned areas. Some subtasks like ball grasping and localization have been addressed by multiple solutions and experiments have proven fundamental for selecting the most effective one. Through extensive tests in the field, the team discovered pitfalls, revised the initial assumptions and design decisions, and took advantage of the iteration process to perform successfully at the competition.

VII. Acknowledgements

We thank Sick Italia Spa for sponsoring the logistic of the competition to the *Red Beard Button* team and Sick AG for promoting and organizing the competition.

References

[1] T. Braeunl, “Research Relevance of Mobile Robot Competitions,” *IEEE Robotics & Automation Magazine*, vol. 6, no. 4, pp. 32–37, 1999.

[2] P. Lima and R. Cortesao, “IROS 2012 Competitions,” *IEEE Robotics & Automation Magazine*, vol. 20, no. 1, pp. 10–12, 2013.

[3] R. Arkin, *Behavior-Based Robotics*. MIT Press, 1998.

[4] P. Althaus and H. I. Christensen, “Behaviour coordination in structured environments,” 2003.

[5] J. Bruce, T. Balch, and M. Veloso, “Fast and inexpensive color image segmentation for interactive robots,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2000.

[6] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004, pp. 2301–2306.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[8] D. Douglas and T. Peuker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartogr. Int. J. Geogr. Inf. Geovis.*, vol. 10, no. 2, pp. 112–122, Dec 1973.

[9] K. O. Arras, “Feature-based robot navigation in known and unknown environments,” Ph.D. dissertation, Swiss Federal Institute of Technology Lausanne (EPFL), Thèse No. 2765, June 2003.

[10] J. Castellanos, J. Montiel, J. Neira, and J. Tardos, “The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building,” vol. 15, no. 5, pp. 948–952, 1999.

[11] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: A general framework for graph optimization,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[12] A. Censi, L. Iocchi, and G. Grisetti, “Scan matching in the Hough domain,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

Table of Contents