Having Fun with Learning Robots

Pavel Petrovic
Comenius University Bratislava
ppetrovic@acm.org

***Abstract*—Artificial Intelligence has a long tradition at Faculty of Mathematics, Physics, and Informatics of Comenius University. The team around professor Kelemen published a textbook on AI fundamentals in 1992 [6] and some form of AI master study program existed since then, currently as part of both Applied Informatics study program and Middle European Interdisciplinary Master Program in Cognitive Science. From this perspective, and basing on a fruitful cooperation with our colleagues from the Faculty of Electrical Engineering and Information Technology of Slovak University of Technology, we established a course named Algorithms for AI Robotics for the final year of the bachelor study program at Comenius University, providing both a hands-on experience in lab robotics projects and a taste of the wide field of applications of computer science to robotics. This article is a contribution to the discussion on the different organisation forms and styles of the robotics courses for undergraduates, summarizes our course and the experience gained. Its main purpose is to inspire other educators and think of their own selection of AI material relevant to robotics. This is the time of important breakthroughs in the field of robotics, cognitive robotics and artificial intelligence and the contents and methods of selecting and presenting the material to students is very important for the future development and applicability of the field.**

I. Introduction

The composition of Computer Science study programs saturated over the recent decade into quite a standard set of courses and forms a solid theoretical basis of the future computer scientists and professionals. However, we learned these people are often lacking the important qualities of team spirit and effective cooperation and communication as well as various practical skills of dealing with technology. We believe it is the responsibility of the universities to provide valuable options for students to overcome this sufficiency. The ability to effectively cooperate in a workgroup has a crucial importance for companies being able to build and provide competitive products and services. Communication and managing skills are often equally important as expert knowledge in order to perform the work at a required quality. However, the higher education programs do not offer sufficient training, focusing too much at the technical and scientific content, usually ignoring the didactic aspects of the learning processes, leaving little space for the student communication, interaction, group work and group learning. As a consequence, the graduates in technology and science programmes cope with hard challenges of organize their team work efficiently and productively. In addition, the popularity of Computer Science programs in the previous decades has been traded recently in favour of more applied programs, where students receive background and training in some particular flavours of informatics. This allows us to harness the interest of these students and expose them to a hands-on course on Artificial Intelligence and Robotics, letting them learn more about connections between the hardware, software and real life.

II. Background and Motivation

Artificial Intelligence is a field of Computer Science that has generated many novel and some useful ideas throughout more than 50 years of research, often interfering with and perhaps spawning new fields that are now considered independent, such as artificial neural networks, logic and functional programming, search, multi-agent systems, or evolutionary algorithms. Thus it is often difficult to say what is still AI and what is already outside of its boundaries. Many researchers would have different opinions on that. Similarly, some may not like the term Artificial Intelligence, and rather use Computational Intelligence, or Intelligent Systems instead. Yet other researchers with strong background in psychology, philosophy, or neuroscience have deep interest in studying human intelligence and cognition and its relevance for a machine or other man-made cognitive systems. We abstain from taking any strong position on this, rather try to take the best from all the areas that are relevant to building physical machines that can ”think”. We believe it is the age of cooperation and building of bridges across the densely cluttered space of research areas.

III. Bachelor Program in Applied Informatics

The starting viewpoint of our program in Applied Informatics is to give the candidate an alternate path to standard Computer Science study program. In particular, the students are allowed to select their courses from a wider range of subjects including Computer Graphics, Machine Vision, Artificial Intelligence, Cognitive Science, Bioinformatics, Declarative Programming, Logic, Robotics, and Mathematics. Only a couple of obligatory courses are supplemented by several obligatory-optional courses and multiple elective courses from a wide selection. The graduates either continue to our Applied Informatics master’s program, or Middle- European International Cognitive Science master’s program with obligatory international exchange student program. Some students also choose to switch to more technology-oriented study programmes within engineering, or business-oriented computer master courses at neighboring universities.

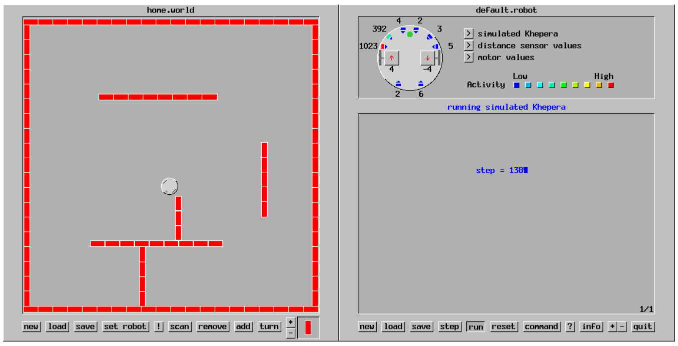

Fig. 1. A robot with a feed-forward neural network trained with the error back-propagation learning algorithm that learned to avoid obstacles in the Khepera simulator, E.Melicher.

Fig. 2. Sbot robot that learned to follow a line using Reinforcement Learning (Q-learning). It required about 50 manual restarts from the start of the line until the behavior was learned. The learned behavior was successfully verified on a different line track, P.Jurco.

IV. Project-based courses

Most of the education at all age levels in our country and world-wide is currently based on traditional academic style of education. While it is very efficient and provides solid results, it is challenged in being distant from the real-world, actual demand of the labor market, and the participants often suffer from insufficient depth of the acquired knowledge. In addition, many students especially at a younger age find the academic style unsuitable, easily loose their interest for education in general, struggle with the motivation, and as a consequence become aggressive, negligent, apathetic, or find easy shortcuts to drugs, crime, and underworld. Providing hands-on activities in the school proved to be able to attract also this kind of students and give them the opportunity to show to themselves, to their peers, and to their teachers that they are skillful, smart, and worth. Positive experience of this kind raises further their interest in learning, and improves their performance in other subjects. However, all students – and again at all age levels – benefit from the hands-on activities and project-based education in establishing the necessary links between the knowledge and the real-world, become more active in exploration and verification of the learned knowledge. Projects give the students a unique opportunity to explore beyond the standard curriculum, learn to work in groups, share and present their work within the study group and outside. Class projects often lead to small student scientific works that are presented at student conferences and similar forums.

V. AI Robotics

Although AI is a much more general field, at all times it has been looking at how its theories and methods could be applied to produce actual physical intelligent beings or devices, robots. Looking from a different viewpoint, Robotics is the latest stage or follow-up of the industrial revolution that freed the manufacturing of arduous human labour, starting from mechanisation, continuing on towards automation and robotics. These are two rather different approaches. For AI researchers the question of *what is intelligence, and how can we understand and design intelligent artifacts*, is a central theme. A typical proponent of this group is Marvin Minsky and his work [1]. What are the principles of intelligence, how can we represent various kinds of knowledge, how to reason about the environment, space, time, other agents and their intentions, how to select actions, communicate, predict, plan and prioritize, cope with uncertainty and unknown environments. On the other hand, industry is interested in precise and highly reliable, durable and productive devices that perform actions on demand. However, as the automation in factories advances, more and more operations and tasks remain for robots, including those, where complex decisions need to be taken, problems solved, and where certain level of intelligence is required. It is thus the right time for joining the efforts, forming interdisciplinary teams and getting inspired by each others' ideas and results. Our course does not make any specific assumptions on preliminary knowledge, although some mathematical and programming background is needed. The described methods and algorithms are not always studied in complete detail due to the limited time, space and scope, but we always refer the readers to the literature, where he or she can learn more. And it is also our aim to abstract from the irrelevant details, and rather provide a broader overview and explain the principles. A student who understands the principles and who can put them into actual practical real-world situations is worth ten times the one who can make a theoretical analysis of a problem, but is unable to connect this analysis to a real-world situation. And that is exactly what inspired us for setting up our course. While students learn theoretical algorithms, and methods, they are required to implement them in projects with real or simulated robots, learn and document how they were able to use them successfully. Interested students are recommended to continue their studies on selected topics further on! The study materials consist of scientific papers on various topics, chapters from textbook, and lecture notes provided by the teacher. They are arranged into chapters in the order of the course lectures. As such, they serve as the reading material for the course, explain and support the topics being covered at the lectures. The material is a dynamic entity and evolves every semester to incorporate new advances and adapt to a better content and structure.

A. Selecting the contents

Taking into account our AI perspective, we have identified several areas that are relevant for the course. We grouped them

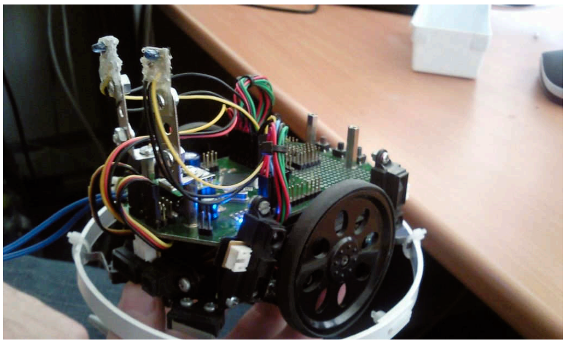

Fig. 3. A modified Sbot robot with ambient light sensors for the Braitenberg vehicles exercise, P.Hudec and R.Maurer.

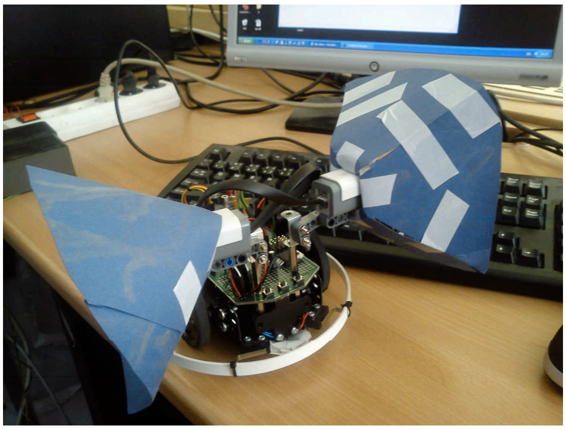

Fig. 4. A modified Sbot robot with LEGO sound sensors for the Braitenberg vehicles exercise, P.Hudec and R.Maurer.

into the following themes: *Perception and sensor systems.* In this theme, we introduce the basic nation of nature of the sensory data, types of sensors, and the challenges of signal processing. *Software robotic architectures*, where the students study the ways how robot control architectures can be organized, and learn about a few famous examples of robotic architectures. *Representation and inference in space* is a central theme of robotics, where the map organisation, localisation, mapping, and spacial planning are discussed. *Navigation and localisation* study the approaches and challenges in moving the robot, getting from place to place in the environment, and the problem of estimating the robot pose from the sequence of sensory data given that the map of the environment

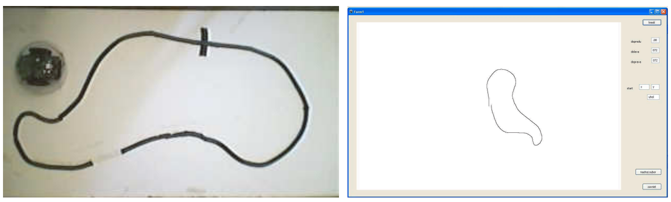

Fig. 5. A simple line track to be mapped by a robot (above) and a resulting map reconstructed from odometry by an application written in Delphi (below), E.Rapcik, R.Stupka.

is known. *Probabilistic approaches* provide a collection of methods for representing and computing with uncertainty by representing and working with the probabilistic distributions. In robotics, uncertainty is present everywhere from perception, action selection, signal processing, localisation, mapping and other areas. *Logic approaches* to robot programming based on symbolic representation and standard predicate calculus, reasoning and inference offer a rigid framework for planning, high-level action selection and representation. *Simulation of robotic systems* deals with the complexity of the robotic simulation, and they way of coping with it. Introduction to simulation types, minimalistic simulations, examples of robotics simulators and their advantages and shortcomings. *Robotics and artificial life* is an alternative and fundamental approach studying how robots exhibit some features of living systems, and how the roboticists can be inspired by phenomena found in the nature. *Applications and other topics* focus on the variety of examples of real-world applications of successful intelligent robotic systems including the educational robotics systems.

VI. The Algorithms for AI Robotics Course

Having in mind the themes that we identified in the previous section, we sat down to devise the criteria for selecting the actual course theoretical content. We try to summarize the criteria it as follows:

- The material should be related to Artificial Intelligence, the more the better.
- It should be possible to use most of the theoretical material in small-size practical projects.
- The scope of the topics should be wide to cover many different areas of AI Robotics, since this will be the only robotics course for most students during their whole studies.
- The material should be accessible using the mathematical and theoretical apparatus the students already have, or alternately, the topics would have to be covered only on the conceptual level.
- There overlap with other AI courses in our study program should be minimized so that they can complement each other.
- Ideally, the topics covered should bring some new knowledge, methods, principles that are useful, or can be easily transfered to other fields than robotics.
- The selected topics should be inspiring, motivating, and fun to learn about.

A. Curriculum

The course consists of two parts. First, lectures that cover theoretical material, several common prepared exercises with hands-on activities, where the students experience the sensing, locomotion, dependence of the robot morphology and robot software, explore the basics of the signal processing. The second part of the course is a project that the students design and implement. For the theoretical part, we have selected the following topics for the lectures:

1) Introduction, overview of basic concepts I: In the first lecture, we invite the students into the world of robotics through videos of applications and research studies. We talk about the basic elements of robotic systems – locomotion, sensing, reasoning, and communication. We give examples of different locomotion types, study various sensors and motors – both from the point of view of physical principles, the signals they generate, and the way we interface them, touch the types of control architectures, and the main communication platforms.

2) Introduction and overview II: In the second lecture, we motivate the students for the topic of learning robots. We cover the different types of learning in humans and nature, and investigate the ways learning could be useful in machines and robots. We give example of simple symbolic learning system, which learns to play an animal guessing game, and the subsymbolic ALVINN system that learned to drive a vehicle. We explain the fundamental parts of the learning system – the knowledge representation and the learning rule. We learn about the symbolic frame representations, and the symbol grounding problem. By the end of the lecture, we motivate the students for the probabilistic representations due to the nature of the data robot obtains from the environment using its sensors.

3) Review of LEGO, navigation, SBOT: The third lecture prepares the students to hands-on exercise with LEGO robots and our AVR ATmega128-based differential-drive robot platform SBOT [7]. The lecture discusses also the topic of navigation in nature and robots.

4) Evolutionary Algorithms + Reinforcement Learning = Learning Classifier Systems: From the lecture four, we dwelve into more specific algorithms and learn about the main principles of one or more selected methods at each lecture. LCS [8] form an important part in the history of learning systems and introduce the students to two crucial types of learning: reinforcement and evolutionary learning. Thus we let them both explore the fundamental concepts of EAs and get the feeling of how the sparse reward gets propagated during a RL process among the states so that the agent is able to select actions even in states when it is far from receiving any reward or punishment from the task and the environment.

5) Neuroevolution through Augmenting Topologies: The NEAT [9] lecture gives another specific application of an evolutionary algorithm, but introduces the students to further important issues of species, fitness sharing, synonymical representations, gene alignment, and popular benchmark problems in control theory.

6) Fly Algorithm, CMAES, MOEA: Fly Algorithm [10] is a simple stereo-vision obstacle-avoidance algorithm that utilizes the similarity of neighborhoods of projection pairs of a population of points randomly generated in the space, Sobel gradient for favouring points on edges, and evolutionary algorithm for searching through the space of possible 3D points. Thus it forms a beautiful and yet simple and understandable algorithm that gently brings up the topics of processing visual sensory information. The Covariance Matrix Adaptation Evolutionary Strategy [11] is an efficient state-of-the-art optimisation algorithm with strong mathematical grounding and gives the students the taste of utilizing the more advanced mathematics for useful practical algorithms, a very important connection to make. Finally, we introduce multiple-objective evolutionary algorithms [12], explain the main concepts of dominance, Pareto-optimality, and discuss the various ways how to modify standard evolutionary algorithms to cope with the multiple objectives, which in usually the case in practical applications.

7) Evolutionary Robotics, Behavior-Based Robotics, Case-Based Reasoning: We finish the journey through various evolutionary methods by looking at Evolutionary Robotics, and exploring the various ways how evolutionary algorithms could be applied to robotics. In particular we study the fitness space as defined by Nolfi and Floreano [13] by devising the types of objective functions along the three axes as behavioral vs. functional, implicit vs. explicit, and internal vs. external. The second part of this lecture touches the very central topic of AI Robotics – behavior-based architectures. From Brooks's Subsumption Architecture [15] through Arkin's motor schemas [14] to hybrid architectures. Finally, we give a short introduction to the lazy learning method Case Based Reasoning [16], describe the main CBR cycle, and the important connections to the very central theme of the artificial intelligence: making analogies, associations, and providing explanations of reasoning outcomes.

8) SIFT and SURF methods for landmark recognition, Hough transform: Visual information is the most contentful, and thus it is also the most useful one for mobile robotic systems. This lecture explains the details of the Scale Invariant Feature Transform [18], and the possible applications for mapping using visual landmarks. We compare it to another efficient method Speeded Up Robust Features [19].

9) Markov Decision Problems: The last four lectures are based on the textbook Probabilistic Robotics of Thrun, Burgard, and Fox [17]. Robots perceive the environment in the form of noisy and unreliable information obtained through sensory percepts. It is therefore skewed by inherent uncertainty. The traditional algorithms assume deterministic digital information and cannot make use of information that is stochastic. However, robots are typically short of information and typically cannot afford to not use the information that is available in any form due to the criticality of their mission and too high risk in the case of mission failure. Probabilistic representations provide a suitable framework for representing the stochastic information. It is however important to realize that probability is never an objective measure. It is rather a subjective belief established on top of subjective estimates. Markovian assumption is prevalent in most probabilistic approaches. It is natural to start with MDPs that take the assumption of complete state that is seldom applicable in practice, but they formulate the essential principles of all probabilistic reasoning.

10) Partially-observable Markov Decision Problems: A more realistic scenario is when the robot/environment state is not complete – in the sense of not being fully observable. When state, action, and perception spaces and the planning

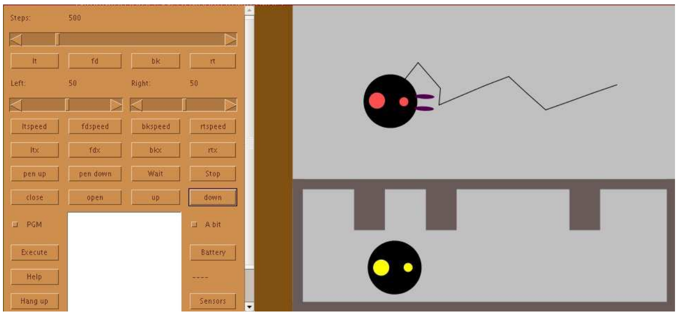

Fig. 7. Robotnacka simulator with the new functionality a gripper and the possibility to place obstacles and setup rooms with different shapes, M.Vince.

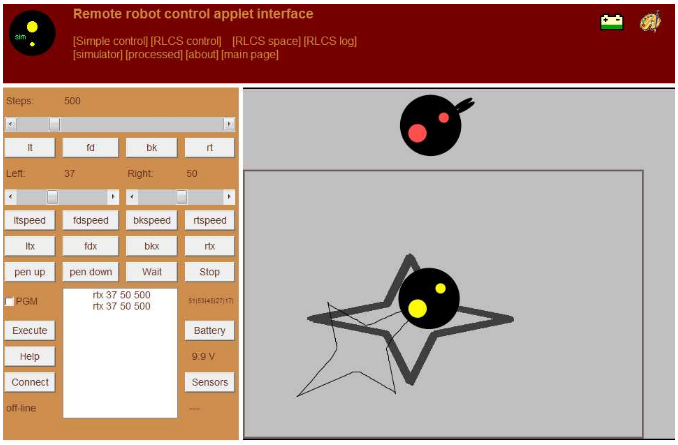

Fig. 6. A line-following robot in a remotely-operated laboratory that learns the shape of the line and draws the learned shape in a simulator, M.Svantner, S.Sitas.

horizon are finite, the ideal policy can be expressed by a piecewise linear function and obtained using the POMDP algorithm discussed in the lecture and demonstrated in an example.

11) Kalman filters: Gaussian filters are practical when representing probabilistic information that has a nature of normal distribution. They require only the mean and variance, or mean vector and covariance matrix in multidimensional case to be represented. Gaussian filters can be used when the belief over the state is probabilistic and represented by the normal distribution. When the state transition probability is linear function of state and action, Kalman filter can be used to estimate the posterior probability of the state belief. [17]

12) Probabilistic Robotics, Bayesian Robot Programming: Having seen the first three key methods in probabilistic robotics, students are exposed to wider range of topics including localisation, mapping, SLAM, and particle filters. A unifying framework of probabilistic reasoning is provided by the Bayesian Robot Programming [20] demonstrated on a little learning experiment where a Khepera robot learns to push obstacles, follow contours, or perform complex tasks in the environment. BRP framework is inspired by the concise organisation of logic programming, and attempts to provide a universal alternative that is based on probabilistic reasoning as contrasted to the predicate calculus.

The selection of the topics is based on a set of original papers that describe the methods. This is an important aspect of the course requiring the students start learning to work with the scientific literature and start preparing for the kind of work they will be required to do at the master's level.

B. Common Laboratory exercises

The exercises of the course start with several exercises for all students, where the students are introduced to different robotic platforms. This hands-on introduction allows them to experience the work with sensor data, their inaccuracy, the issues in the control, the high coupling between the mechanical and program design, and various control paradigms. We start with Lego Mindstorms exercise where all students go through a tutorial on sensors. They are asked to program the robot to drive in a square, 8-shape, avoid obstacles, react to sound, perform simple line-following, and in a line-counting task establish understanding for the difference between a pure-reactive control and control with an internal state, or even a deliberative control. The exercise has a voluntary follow-up, where the students are allowed to experiment with their own ideas and build simple projects.

The second organized exercise is centered around the Sbot platform developed by Robotika.SK. The students learn about programming this embedded platform in C language using AVR studio, learn about programming the various features of the single-chip microcomputer including timers for PWM control of servos and DC motors, analog-to-digital converters for reading from various analog sensors, digital inputs and outputs for reading from bumpers, and controlling the LEDs, I^2C for communication with advanced sensor devices and serial communication over Bluetooth. Students program simple line-following and obstacle avoidance using bumpers and SHARP infra-red proximity sensors. In a voluntary follow-up exercise, students experiment with simple projects based on their own ideas. The strength of the platform lies in the practical serial Bluetooth console with redirected standard input and output allowing instantaneous debugging output and control, as well as easy collection of data on a PC for further processing.

In the third exercise, the students work with the Acrob platform [21], which is based on the Arduino. The user-friendly development environment with the possibility to graphically visualize the sensory measurements provides a wonderful teaching platform. In addition, good-quality robotics exercises that were prepared by the Acrob author are put in use.

The last organized group exercise is working with remotely-operated robots in our Remotely-operated robotics laboratory [23] or the one of the Czech Technical University [22]. Our laboratory contains also a realistic simulator, which introduces the students to the concept of robotic simulations. The robots operating in the laboratory are equipped with high-precision stepper motors and can be viewed from an overhead network camera. The exercise utilizes the computer vision library OpenCV for recognition of shapes and undistorting the frames obtained from the camera with wide lenses.

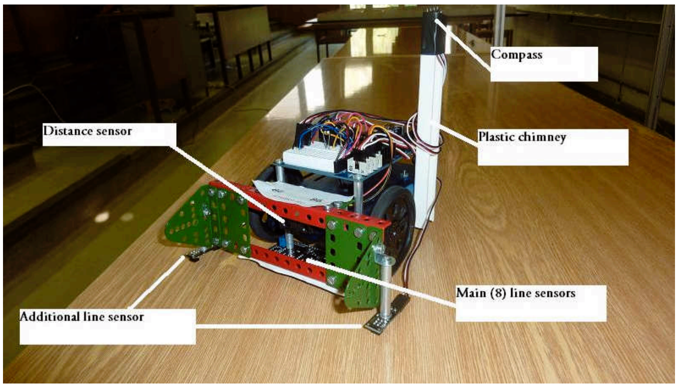

Fig. 10. Ketchup robot that won the Istrobot contest, I.Lack and F.Nagy.

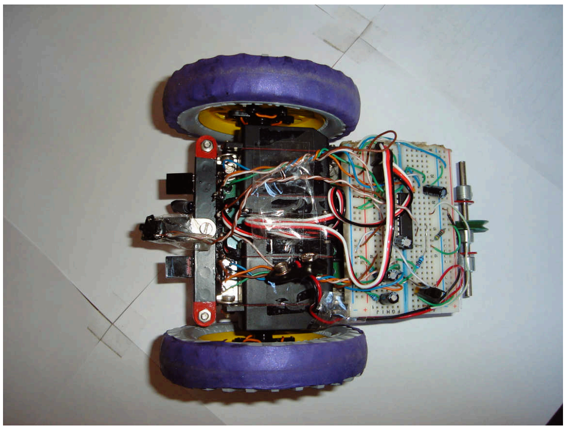

Fig. 8. A hand-made robot of a first-year physics student for the line-follower competition, R.Vanta

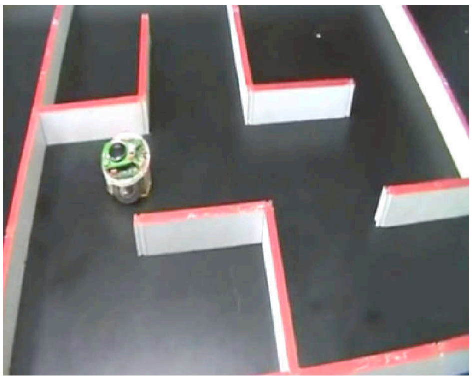

Fig. 9. E-puck implementation for the Micromouse contest, M.Galik and Z.Koysova.

C. Student projects

The student group project according to the student feedback in the poll organized by our faculty is the most praised part of the course. The students are asked to develop a simple real or simulated robotics project on a selected (or own, but approved) topic. The requirement is that the designed system is working, and that the students write a project report containing the source-code and photo or video documentation on the course wiki page. We encourage the students to use some AI algorithm or let the robots learn, although it is not a requirement. Some projects are focused more on the hardware, other on the algorithms.

The quality of the projects vary, we find some of them particularly interesting, and we describe them in this section.

The project *Obstacle avoidance training in Khepera 2 simulator via neural network* is a typical project in simulation. It utilizes the old-fashion, but simple to use simulator of the Khepera robot to train a feed-forward neural network using the standard error back-propagation algorithm, Figure 1 In this project, the student tested the ability of the neural network to learn the obstacle-avoidance behavior. Thus he first wrote a deterministic program for obstacle-avoidance, collected and filtered the training data from its run, and used it to successfully train a neural network that performed the obstacle-avoidance behavior taking the sensory readings on its input and producing the motor speeds on its output.

Line Following Sbot Using Reinforcement Learning - in this project, the student developed a RL algorithm with the purpose to learn to follow a black line on a white surface using two light sensors and SBOT robot. The robot is using a third sensor that is placed in between to compute the reward. After the robot leaves the line, the operator has to lift the robot and place it back on the start. The learning session required around 50 starts. The important aspect of this project is that unless the student writes the learning formulas and tunes the parameters of the system correctly, the system won't learn. Thus despite the simplicity of the learning task, the student must acquire a full understanding of the learning algorithm. See Figure 2.

In a similar project *Light Avoiding Robot*, the SBOT robot learns to navigate away from the light. The students built two simple analog light sensors using phototransistor and a resistor and placed them on the sides of the robot. The state of the system was determined by the sensory readings, while the actions moved the robot for half a second in one of the possible directions. After the action, the Q-value for the particular state-action pair was updated using standard Q-learning algorithm. The robot learned a working light-avoiding behavior within two minutes of learning time.

Two other projects we mention here used the SBOT platform. In the project *SBOT pushing a vehicle* the student designed a rotation sensor that detected the direction of a trolley attached to the front side of the robot using a joint that could rotate along the vertical axis. The sensor used a light sensor and LED in a sealed box so that turning the trolley resulted in changes of the light intensity detected by the sensor. The student developed a control algorithm for stabilizing the straigth direction of the robot with the pushed trolley using the designed sensor. In the project *Braitenberg vehicles implemented using SBOT robot*, the students designed several usual and unusual types of Braitenberg vehicles utilizing for instance the LEGO sound sensors, Figure 3 and 4.

Several projects always deal with our remotely-operated robotics laboratory, which serves as our perpetual playground. In the project *Controlling robots in remotely-operated laboratory using Objection language*, a student who designed his own object-oriented functional programming language developed a library for his language for controlling the remotely-operated robots, designed the corresponding interface, and implemented example programs. In the project *Gripper Functionality for Remotely-Operated Robotics Laboratory*, the student learned about the implementation of the existing simulator and implemented the new feature of gripper manipulator. In another project, *Robot with ultrasonic distance sensors*, the student designed a control board for servicing five ultrasonic distance sensors, learned about the electronic interface and the protocol for communication with the main board, developed a higher level protocol and built an extension turret for the Robotnacka robots. This extension has been in a successful operation for several years since. Yet in another project *Reproducing a learned geometrical object using drawing pen of Robotnacka robot*, the student utilized bottom light sensors and odometry to learn a shape of a drawn object, consequently, the drawing robot reconstructed the learned shape using its pen.

Students are also allowed to work on their own inventions – robots they want to build using their own parts and tools as long as their projects and reports satisfy the requirements, while they are still invited to use the lab facilities. These robots often participate in robotics competitions, example of such being the project *Highlander* built using PICAXE platform shown in figure 8. We also provide the students with robot hardware so that they can build the robots that participate in the competitions – examples being the e-puck robot in project *Mouse in Maze with robot E-puck* or the Acrob-based *Ketchup Robot* that won the first place in the Istrobot contest this year, see Figure 10.

VII. More Fun with Robots

Our students get involved with robotics also in their bachelor and master theses. Jakub Kondela examined the Khepera experiments with the Bayesian Robot Programming and reimplemented them in the SBot platform in his bachelor and master theses. Lukas Risko studied the probabilistic mapping and localisation with the robots in our remotely-operated laboratory both in his bachelor and master thesis. Dominik Misanic and Tomas Stibrany considered the problem of localisation using the distance sensors in a simulated environment and Vladimir Satura implemented a simulated Sokoban-robot and verified it on the SBOT platform. Two master theses dealt with educational robotics - Mikulas Pataky developed a set of robotics laboratory exercises for physics in upper secondary schools, while Daniela Lehocka focused on particular experiments with mechanics for lower secondary schools using Imagine Logo. In his bachelor thesis, Robert Maurer reimplemented the stereo-vision Fly Algorithm using the Surveyor Stereo Vision System and Pavol Hudec implemented a Simulator of robotic sailing robots for our cooperating partners from Innoc. Perhaps the most exciting (from here comes the title of this paper) was the master thesis of Miroslav Nadhajsky who built an outdoor robotic platform for the Robotour contest and investigated various neural architectures for detecting the path in the camera image. He has also designed the complete control architecture that integrates information from multiple sensors, such as GPS, compass, ultrasonic distance, and makes use of a map downloaded directly from the public Open Street Map server. Peter Jurco studied the various uses of the Reinforcement Learning in Robotics, and demonstrated its use in an interesting simulated example implemented using the MS Robotics Studio. Most recently, Peter Pukancik built a mobile robot with a 5-degrees of freedom arm with a 3D vision controlled by a custom-built control board and Gumstix microcomputer with Ubuntu Linux and implemented algorithms of forward and inverse kinematics for the precise arm control, while Daniel Skalicky implemented and compared various algorithms for the e-puck educational robot for Micromouse contest utilizing also its camera for maze exploration. A really funny robotic experience, however full of learning, our students experience when getting involved in one of the robotics contests - FIRST LEGO League, RoboCup Junior, RobotChallenge, Istrobot, or Robotour, all of which they are regularly volunteering as referees or supporting staff.

VIII. Evaluation

During the three years we organized the Algorithms for AI Robotics course, 45 students participated and finished with a credit. The distribution of the grades was as follows: 15% students earned A, 5% with B, 39% with C, 21% with D, 18% with E. Half of the points are awarded for the project, and the other half for a written exam containing questions from the theoretical material covered in the lectures. Students are given access to the original papers describing the algorithms, thus we focus on their ability to comprehend the ideas rather than being able to remember specific details and formulas. As mentioned above, the most popular part of the course is having fun with the robots – i.e. the practical exercises and project work. However, the theoretical content is required so that the students know where to start and get exposed to wider range of the methods in addition to the one they choose for their project. Our laboratory does not accommodate more than 6 students at a time, therefore we have a dedicated time of about 15 hours throughout the week when the students can come to the lab after they have registered on-line, and get a support from us during the registered hours in the laboratory. On one hand, this puts high demands on the teaching staff, but the students need to get the necessary personal guiding anyway as they usually work on completely different types of projects. Once they receive an introduction and get started, they tend to work individually, and require help only occasionaly, thus allowing the teaching staff to work on their own projects during the lab hours. Most students who chose their topic for the bachelor or master thesis within the field of robotics did take the course, which often inspired them to do so. To summarize, we see the following benefits of incorporating the course in the study program of Applied Informatics:

- Provides a different viewpoint on artificial intelligence, which is one of the two main specialisations of our study program. This viewpoint is being crucial since many scientific views in AI assume the body is a requirement

for an intelligent agent. Allowing the students actually experience the work with such physical agents, students can acquire a much better understanding of the issues the embodied intelligence must solve.
- One of two courses where the students work with hardware within the study program.
- Project-based education.
- Motivating aspect and benefit for the whole study program (attractive to students).
- Provides a unique bridge between theory and applications.
- Helps students select topics for bachelor and master theses.

When asking the students about the one thing that was useful, interesting, or otherwise positive, and one thing that could be improved/changed, we have received the following representative answers: 1A) I enjoyed the "playing" with robots. One finally sees the results of programming in the real world, not only on the screen. 2A) I would appreciate if the algorithms covered by the lectures were introduced more practically, perhaps some examples of use. 1B) It was interesting to see how solving simple practical problems sometimes requires a bunch of theory. 2B) When the lecture discussed something I saw for the first time, I was getting lost, I might have been missing some introduction. This suggests that the course is definitely meeting some of the objectives, but also that we still have a way to go to make the material even more accessible and understandable, making even more connections between the theoretical part and its use.

IX. Conclusions and Future Work

We described and shared our experience from a particular course on Robotics for students of Artificial Intelligence. The course is a combination of theoretical lessons covering various algorithms that are useful for robotics and a set of practical exercises, and project work. From the feedback provided by the students we learned that the students liked the course, in particular the hands-on activities. We discussed the contents of the course and some of the student projects. The article is a contribution to a discussion of how to organize the courses on robotics, which are difficult to manage, maintain and service. We find that the course suits well in the study program so that the students learn to work with technologies that were otherwise beyond the standard curriculum. We consider other benefits of our course as well. Before the next occurrence of the course, we plan to finalize a concise study text materials for the course so that the students are not dependent on reading the articles, which will thus become a supplementary reading material. We are also permanently updating the contents of the course, and the hardware provided to the students.

When setting up our course, we were not inspired by any other particular course. Robotics study program and courses at a neighboring university (Slovak University of Technology) are taking the engineering perspective, we wanted to be different. Also, the size of the course is not large enough to cover a complete introduction to robotics. There were no robotics courses in our study program or faculty previously, and thus we were doing something completely new: emanating from the tradition of a stronger theoretical background of the computer science students in our faculty and attempting to extend this tradition towards more hands-on experience and practical knowledge. When trying to compare this course to other robotics courses at this type of university, we find it unique, in the sense that it allows the AI students to peek into the world of robotics from their perspective, without having to go through a systematic process of taking a sequence of introductory robotics courses, yet, it allows them to use high-level knowledge in AI in a form of an application in this practical and interdisciplinary field.

References

[1] Minsky M. *The Society of Mind.* Simon and Schuster, New York, 1998.

[2] Cortenkamp D., Bonasso R.P., Murphy R. (ed.) *Artificial Intelligence and Mobile Robots, Case Studies of Successful Robot Systems*, MIT Press, 1998.

[3] Nehmzow U. *Mobile Robotics: A Practical Introduction*, Springer, 2000.

[4] Corrochano E.B. *Geometric Computing for Perception Action Systems. Concepts, Algorithms, and Scientific Applications.* Springer 2001.

[5] Murphy R.R. *Introduction to AI Robotics*, MIT Press, 2000.

[6] Kelemen J. et al. *Zklady umelej inteligencie*, Alfa, 1992.

[7] Gustafik D. *SBot v2.0 – Educational Robot for Clubs and Classrooms*, Robotika.SK, 2009.

[8] Sigaud O. and Wilson S.W. *Learning Classifier Systems: A Survey*, Soft Computing - A Fusion of Foundations, Methodologies and Applications, vol. 11, number 11, pp.1065–1078, 2007.

[9] Stanley K.O. and Miikkulainen R. *Evolving Neural Networks through Augmenting Topologies*, Evolutionary Computation, vol. 10, number 2, pp. 99–127, 2002.

[10] Lochet J. et.al. *Dynamic flies: a new pattern recognition tool applied to stereo sequence processing*, Pattern Recognition Letters vol. 23, pp. 335–345, 2002.

[11] Hansen N. *The CMA Evolution Strategy: A Comparing Review*, in Lozano J.A. et. al. (eds.), Towards a new evolutionary computation. Advances in estimation of distribution algorithms, pp. 75–102, Springer, 2002.

[12] Zitzler E. *Evolutionary Algorithms for Multiobjective Optimization*, in Giannakoglou et.al. (eds.), Evolutionary Methods for Design, Optimisation and Control, CIMNE, 2002.

[13] Nolfi S. and Floreano D. *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, 2000.

[14] Arkin R.C. *Behavior-Based Robotics*, MIT Press, 1998.

[15] Brooks R.A. *A robust layered control system for a mobile robot*, IEEE Journal of Robotics and Automation, vol. 2, nr. 1, pp. 14-23.

[16] Kolodner J.L. *An Introduction to Case-Based Reasoning*, Artificial Intelligence Review, vol.6, pp. 3–34, 1992.

[17] Thrun S., Burgard W. and Fox D. *Probabilistic Robotics*, MIT Press, 2005.

[18] Lowe D.G. *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision, vol. 60, pp. 91–110, 2004.

[19] Bay H., Fasel B. and Van Gool L. *Interactive Museum Guide: Fast and Robust Recognition of Museum Objects*, in Proceedings of the First International Workshop on Mobile Vision, 2006.

[20] Lebeltel O. et.al. *Bayesian Robot Programming*, Autonomous Robots, vol. 16, number 1, pp. 49–79, 2004.

[21] Balogh R. *Laboratory exercises with Acrob robot*, in Proceedings of 2nd International Conference on Robotics in Education (RiE 2011), pp. 41–46, 2011.

[22] Kulich M. et.al. *Users Access to the Robotic E-Learning System SyRoTek*, in Proceedings of the 4th International Conference on Computer Supported Education, SciTePress Science and Technology Publications, vol. 1, pp. 206–211, 2012.

[23] Petrovic P. and Balogh R. *Experiences from Deployment of Remotely-Accessible Robotics Laboratory*, in Proceedings of the 1st International Conference Experiment@ (exp.at'11), 2011.