The Minor Specialization Robotics at FEE CTU in Prague

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

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

I. Motivation

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

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

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

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

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

II. Overview

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

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

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

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

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

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

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

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

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

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

III. Practical robotics

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

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

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

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

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

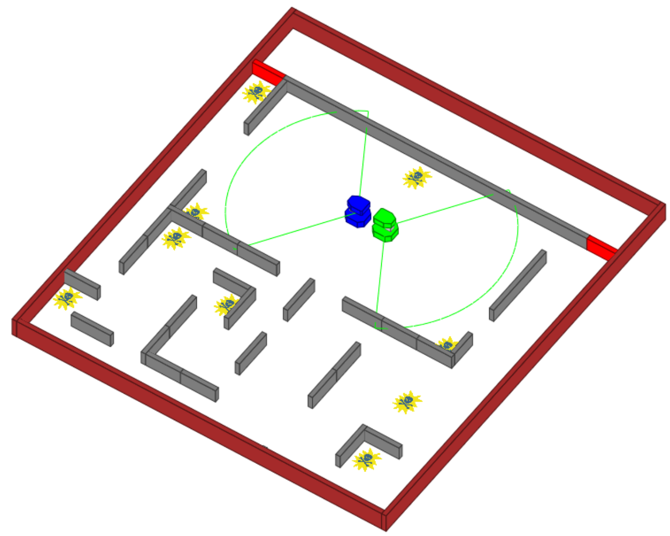

Fig. 1: Configuration of obstacles and mines for the competition.

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

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

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

A. Introduction, task formulation

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

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

B. Planning on a binary grid

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

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

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

C. Collision avoidance

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

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

D. Mapping and localization

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

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

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

E. Exploration and application integration

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

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

F. Final demonstration, competition

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

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

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

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

G. Questionnaires

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

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

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

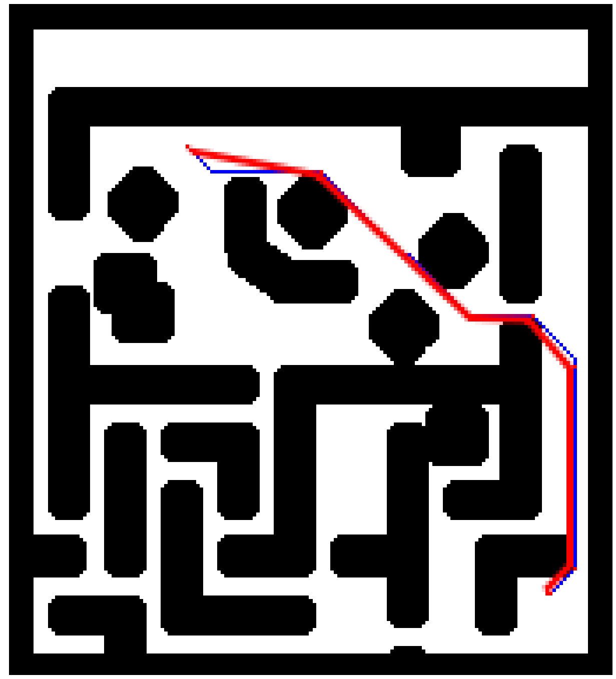

Fig. 2: A part of the Questionnaire 1. Path found by Dijkstra algorithm (blue) and the smoothed trajectory (red)

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

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

IV. Mobile and Collaborative Robotics

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

Fig. 4: The SyRoTek arena and robots

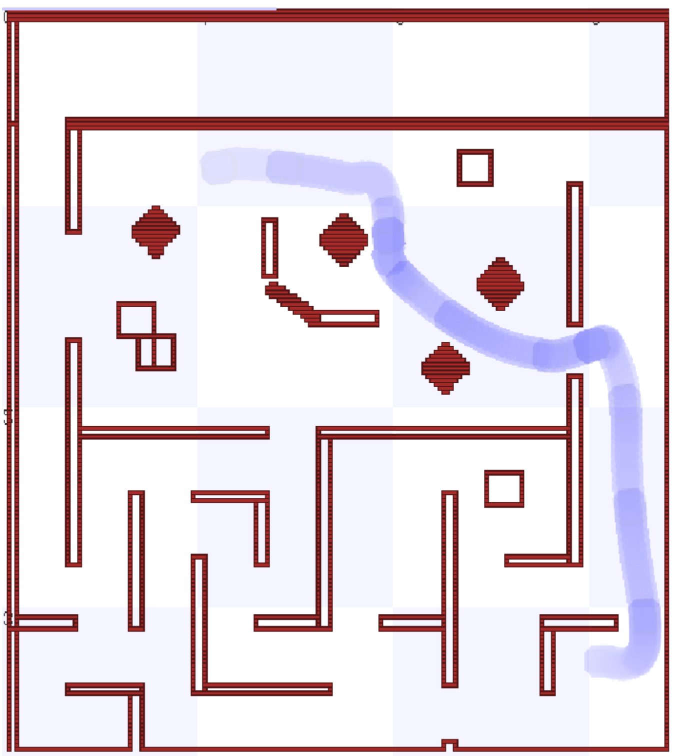

Fig. 3: A part of the Questionnaire 2 - a trajectory traversed by a robot.

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

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

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

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

V. Conclusion

During the first year of the *Practical Robotics* course we recognized that implementation of all the particular steps was too time consuming and although software modules prepared by the teachers were available, the students were not able to finish the particular steps in time. The main difficulty appeared in realization of VFH+ algorithm, which took more than twice more time for some groups than expected. In our opinion, it was caused by not enough programming experience of the students. Because of that, the schedule of the labs was modified so that information about robot's position was provided, and therefore, the students didn't have to implement their own localization algorithm. More crucial was that the majority of teams had no time to run the code on real robots and to perform experiments with them. Only one team was able to experiment with a real robot. Student's applications were thus tested in a simulated environment and also the competition was performed in simulation.

As "playing" with real robots is one of the aims and motivations of the course, the schedule for the next years will be modified to give students more time to experiment with real hardware. Therefore, implementation of VFH+ will not be requested. Instead, the students will have to properly tune parameters of the VFH+ driver distributed within Player/Stage and/or compare its behavior with Smooth Nearness Diagram Navigation (SND) [9], which is another local navigation method.

Based on the previous experience [10] `subversion` version system [11] was used in both courses as a code repository and a tool for task commitment. Majority of the students had experience with this tool from other lectures, so they had no problem to use it actively. On the other hand, the students didn't like to use prepared libraries. For example, graphical visualization of output of the particular algorithms was provided in two ways: (1) by extension to Stage, and (2) C++ API for `gnuplot`. Instead, several groups wasted time with their own implementation of visualization.

In *Practical Robotics* course, hardware parts of the SyRoTek e-learning system [12] (see also Fig.4) were used by the students particularly. As the whole system is ready now, it will be used as a major teaching platform for both *Practical Robotics* and *Mobile and Collaborative Robotics* courses. This should improve productivity of the students, their immersion into solving of the allocated task, motivate them to finish all the task in time and to simplify their communication with each other and with the teacher.

As the minor runs only one year it is too early to talk about its general effects. On the other hand, discussions with the students show that the minor is interresting for them in general. We will see, whether this interest will project into increased participation of the students in robotics projects in the future. A good news is that many students of *Practical Robotics* look for the topic of their thesis in the robotics domain, but general conclusions about positive effects of the minor can be made after several years.

Although a structure of the minor has been dictated by an administrative limitations, mixture of general courses already taught at the university giving theoretical overview of robotics and control with one practical introductory course at the beginning and one special course dealing with mobile robotics at the end of the study is viable and gives students a general overview of the field. On the other hand, time capacity of the minor does not allow to go into deep details and therefore some methods and problems are mentioned briefly. For example, one of the hottest topics today, simultaneous localization and mapping, is only introduced in few minutes without explaining at least one method.

Acknowledgment

This work has been supported by the Ministry of Education of the Czech Republic, under the Project No. 6840770038 - Decision Making and Control for Manufacturing III.

References

[1] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proc. of the 11th Int. Conf. on Advanced Robotics*, 2003, pp. 317–323.

[2] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[3] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1998, pp. 1572–1577.

[4] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," in *Proceedings of the Sixth Conference on Uncertainty in AI*. 2929 Campus Drive, San Mateo, CA 94403: Morgan Kaufmann Publishers, Inc., 1990.

[5] G. Weiß, C. Wetzler, and E. von Puttkamer, "Keeping track of position and orientation of moving indoor systems by correlation of range- finder scans," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 1994, pp. 595–601.

[6] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*. IEEE Comput. Soc. Press, 1997, pp. 146–151.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. The MIT Press, August 2005.

[8] J. Klein and L. Spector, "3D multi-agent simulations in the breve simulation environment," in *Artificial Life Models in Software*, M. Komosinski and A. Adamatzky, Eds. Springer London, 2009, pp. 79–106.

[9] J. W. Durham and F. Bullo, "Smooth nearness-diagram navigation," in *Int. Conf. on Intelligent Robots and Systems - IROS*, 2008, pp. 690–695.

[10] J. Faigl, T. Krajník, K. Košnar, H. Szücsová, J. Chudoba, V. Grimmer, and L. Přeučil, "Mobile robotics at FEE CTU," in *Proc. the 1st Int. Conf. on Robotics in Education, RiE2010*. FEI STU, Slovakia, 2010, pp. 43–48.

[11] "http://subversion.apache.org," (accessed 8 June 2011).

[12] J. Faigl, J. Chudoba, K. Košnar, M. Kulich, M. Saska, and L. Přeučil, "Syrotek - a robotic system for education," in *Proc. of the 1st Int. Conf. on Robotics in Education, RiE2010*. FEI STU, Slovakia, 2010, pp. 37–42.