OUR NEXT GENERATION OF ROBOTICS RESEARCHERS? TEACHING ROBOTICS AT PRIMARY SCHOOL LEVEL

Dr. Chee Kit Wong
Industrial Research Limited
Auckland 1140, New Zealand
Tel: +64 9 920 3646
Email: k.wong@irl.cri.nz

Professor Wai-Kiang Yeap
AUT University
2Auckland 1010, New Zealand
Tel: +64 9 921 9565
Email: wai.yeap@aut.ac.nz

Dr. Jochen Schmidt
Continental
Advanced Driver Assistance Systems
Kemptener Str. 99, 88131 Lindau, Germany
Tel: +49 8382 9699 145
E-Mail: jochen.10.schmidt@continental-corporation.com

Abstract

In this paper, we present our experience in designing and teaching of our first robotics course for students at primary school level. The course was carried out over a comparatively short period of time, namely six weeks, 2h per week. In contrast to many other projects, we use robots that researchers used to conduct their research and discuss problems faced by these researchers. Thus, this is not a behavioural study but a hands-on learning experience for the students. The aim is to highlight the development of autonomous robots and artificial intelligence as well as to promote science and robotics in schools.

Introduction

Robots have been used for educational purposes for quite some time already, mainly at high-school and undergraduate level. The purpose of teaching robotics to undergraduates is quite clear, as they have usually chosen a field of study that is related to robotics or artificial intelligence, so it fits well into the curriculum. Obviously, teaching robotics at university level is very different to teaching it in schools, as most of the university students know how to program and know the fundamental mathematics necessary to design and develop software for mobile robots. Using robots in schools is mainly to spark interest in science and engineering. In particular at primary school level, it is important that learning is fun and practical, at least to an extend that keeps the children interested, as their attention span is often short. This may be one of the reasons why the Lego robotics kit has been so popular for robotics courses, not only in schools but also at universities (Beer et al., 1999; Fossum et al., 2001; Lau et al., 1999).

Many courses are therefore targeted at high-school (Nourbakhsh et al., 2005; Nourbakhsh et al., 2004; Rodger & Walker, 1996) and undergraduate students (Billard, 2003; Blank et al., 2003; Kumar & Meeden, 1998; Lalonde et al., 2006), or even both (Ahlgren & Verner, 2002; Beer et al., 1999; Miglino et al., 1999). A popular means of attracting students and keeping them interested are contests, see e.g. (Ahlgren & Verner, 2002). Some courses are especially designed to get girls interested in engineering and science at an early age, as this group is often underrepresented in the corresponding university degree courses (Rodger & Walker, 1996).

Most publications regarding robotics for educational purposes at primary school level are in fact social studies, and do not focus on programming or the type of robots that are commonly used in research. They concentrate on how children interact with robots and behave in their presence (Bumby & Dautenhahn, 1999; Salter et al., 2004), or how they communicate with toy robots (Kanda & Ishiguro, 2005). Another important aspect are gender-based studies, where social scientists study the behavioural differences between boys and girls when they interact with robots (Fossum et al., 2001). Exceptions are (Qaiyumi et al., 1998) and (Lau et al., 1999). The first uses robots to teach concepts of science and engineering to primary and middle school students. The latter is particularly interesting because of the wide age range (10 to 18 years) targeted.

Most robotics courses are designed to be taught either as a compact course or over an extended period of time. Typical durations are for example 10 weeks with 1.5h per week (Lau et al., 1999), one week (Rodger & Walker, 1996) or even seven weeks full-time (Nourbakhsh et al., 2005; Nourbakhsh et al., 2004).

In contrast to these publications, we will present a robotics course that has been designed for primary school students, about 10 years old. The time-frame is quite tight, namely, six weeks, 2h per week; the course was held during October/November 2005. It was organised for a group of nine talented students from a private girls school in Auckland, New Zealand. The aim of the course was to highlight the development of autonomous robotics and artificial intelligence, including programming the robots, as well as to promote science and robotics in schools, and at the same time to stimulate leadership, confidence, inquisitive, and team work skills. The topics that were taught during the duration of the course included an introduction to robotics (what are they and what are they used for?), as well as presenting the mechanics and sensors of the robots to the students. In contrast to similar projects, we did not use toy robots such as provided by the Lego kits, but three mobile platforms that we use in our research projects (see Figure 1).

At the start of the course, the students were divided into three groups of three. Each group was assigned a computer and a robot; details on the actual hardware used will be presented in the next section. In order to get the students acquainted with the robots, particularly their restrictions and inaccuracies, we designed a few experiments that can be done with the robots before the students are actually introduced into programming. These experiments are described in the section on experiments performed by students. The programming language we chose for the students is Python, which has been used successfully for teaching robotics at undergraduate level before (Blank et al., 2003). The main reasons for choosing Python will be discussed in the section on software development.

Figure 1: The three Pioneer robots used for the course. Each is equipped with eight sonar sensors, the one in the middle has an additional laser range finder.

As mentioned before, contests are a popular means of keeping students interested, which is why we decided to have a competition between the three groups at the end of the course. To encourage the students to become creative, we tried not to restrict what they could do but, at the same time, it is important to give them a task that is actually solvable within the tight time-frame. Our final choice is to have a robot dancing contest. The aim was to provide an opportunity to apply the knowledge on robotics that they gained during the course in a single application, and encourage them to become creative and have fun. Each group chose a song for the dance performance, and programmed the robot to move synchronously to the music, being allowed to move only within a pre-defined region of the dance floor. The competition was held at the school, which organised an open day and invited the parents and classmates to attend the function. The judges for the competition were the dean of our faculty and the school's principal.

Hard- and Software Environment

This section describes the hardware and software that was used for the course.

Robot Hardware

We used three mobile robot platforms from MobileRobots Inc (formerly ActivMedia Robotics) ("MobileRobots Inc.," 2008), one Pioneer 2 and two Pioneer 3 (see Figure 1). All robots are equipped with eight sonar sensors and an odometer, which were the only sensors used for the classes. Although one of the robots is additionally equipped with a laser range finder and bumper sensors, we decided not use these sensors in order not to favour one of the three groups. The robots are controlled by a PC running Linux, which is on-board in one case, and mounted externally on top of the robot (in form of a laptop) in the other two cases. The students could logon remotely from an external Windows-PC using secure shell and a wireless network connection. Most students were already familiar with the Windows environment, so we did not encounter any major problems regarding the remote login and transferring files that were written on the PC to the robot.

Software

Before we started with the actual programming of the robot, the students were given a set of experiments which they had to perform, thus giving them a stepwise introduction to the functional aspects of the robots, its capabilities as well as its limitations. The experiments will be discussed in more detail in the next section. To make it easy to use, we provided a set of commands that can be executed by the students from the (remote) command-line. This includes commands for moving the robot forwards and backwards by a certain distance, given as a command-line parameter, commands for turning left and right by a certain angle, and a program to output the current sonar readings of all eight sensors. For the programming exercises we chose Python as the programming language, mostly because of the ease of use: The students can write a program on the Windows-PC, transfer it to the robot and execute it straight away, without needing to compile it first as Python is an interpreted language. We provided a template that contains all the necessary startup commands, like connecting to the robot, initialization of the sensors, etc, so that the students did not have to care about these things and could rather concentrate on problem solving.

Experiments Performed by Students

As mentioned earlier, one of the main goals of this course is to expose the “behind the scenes” of working with autonomous robots. Consequently we designed our experiments to highlight two major problems faced by robotics researchers: the errors accumulated via translational and rotational movements and the inaccuracies introduced from the sensors. These experiments, together with the findings of the students, are described in details below.

Translational Errors

To demonstrate what exactly are translational errors and its effects, we devised three experiments. Figure 2 shows a picture of one of the groups performing a translational error experiment.

Figure 2: Picture of one team performing a translational error experiment.

Experiment 1: Simple movement. In this experiment, students were required to instruct the robot to move forward by four different distances: 500mm, 1000mm, 2000mm and 3000mm. For each of these distances, the students had to measure the exact distances the robot actually travelled and compared that to the presumed distance travelled. *Result:* The students found that there were minor errors for the smaller distances but became more inaccurate for larger distances.

Experiment 2: Drift. For this exercise, the students learned all about drift: "What is drift?", "Why does it occur?" and more importantly, "How does it affect the results?". The students measured a straight line, with a length of five meters. They then commanded the robot to move five meters from one end of the straight line and recorded their findings. They were also asked to note differences in the surface of the flooring when the robot starts drifting, as the robot traversed. *Result:* The students were amazed to find how significant the floor surface adversely affects the way the robot travels. In the experimental setup, they found that the main contributor to drifts is uneven carpeting.

Experiment 3: Repetition inaccuracies. To illustrate the cumulative effects of translational errors, this experiment involved the students repeatedly moving the robot forwards and backwards (reversed) over three meters for three times. They noted the end location of the robot, compared to where the robot started originally and measured the distance. *Result:* The students by now expected that the robot would make translational errors and were therefore not surprised that the robot did not end up where it started.

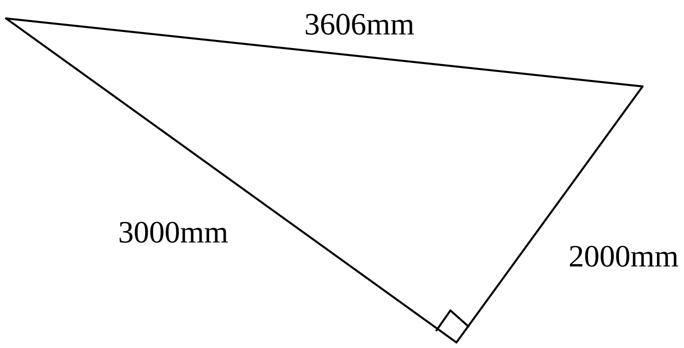
Rotational Errors
Similar to the experiments demonstrating translational errors, these experiments were to show the students the cause and effect of rotational errors.

Experiment 4: Simple rotation. Here the students had to rotate the robot by 360 degrees and upon completion, they had to note if the robot is still heading the same direction as its initial heading. *Result:* The students only found some minor rotational errors, which sometimes can be visually undetectable.

Experiment 5: Repetition errors. For this experiment, the students rotated the robot clockwise and anticlockwise alternately by 360 degrees for four times and recorded the difference between the original and final headings of the robot. *Result:* Although not surprised, the students were amazed to find large discrepancies between the two headings. Furthermore, the students also noted that the robot moved translationally during rotation as well.

Experiment 6: Combined translational and rotational errors. To allow the students to appreciate the problems roboticists face during navigation, the students were asked to program the robot to move according to the diagram shown in Figure 3 (with the robot starting and finishing at spot A). The commands for turning the robot by the correct angle at the corners were given to the students, as they are not familiar with trigonometry at that age. *Result:* As expected, the robot was not back at the starting point of the experiment.

Figure 3: Diagram showing the movement for the combined translational and rotational error experiments.



Sonar Sensors
The other goal of the course was to familiarize the students with the sensing capabilities of robots, more specifically sonar sensors. We wanted to explain to them how measurements of the environment are acquired and why sonar sensors are another source of inaccuracy that roboticists have to deal with. To do this, we devised another three experiments, which are described shortly in the following.

Experiment 7: Data gathering. For this experiment, we wanted to familiarize the students with sonar sensing by collecting sonar distances of an object at four distances: 50mm, 500mm, 2000mm, and 5000mm. *Result:* The students found that they were not able to collect any measurement for the 50mm and 5000mm tasks, due to the limited range of the sensors. However, measurements made for the other two distances were very accurate.

Experiment 8: Detection of various objects. This exercise was to get the students measuring objects of different sizes, material and shape. The objects used were paper dart, pen, and foam cup. *Result:* The students were quite astonished to find that they were not able to measure small objects and found that at different angles, the readings sometimes disappear, as if nothing is in front of the sonar sensor.

Experiment 9: Angle of measurement. This experiment was to demonstrate the affects sonar beamwidth has on measurements and how it causes inaccuracies in data collection. For this, the students held an object at a fixed distance from a sensor and noted the reading as the object is moved angularly. *Result:* The students noted that, even when the object is being moved, the sonar sensor kept detecting the object. They realized that one cannot rely on the reading, in terms of exact direction of the object.

Software Development

After performing the experiments, the students had a good impression on what the robot is capable of doing, and to what extend errors have an influence on the accuracy of motions and sensor data. At this point, they were ready to be introduced into actually programming the robot, rather than just issuing single commands in the shell. The allocated time for programming was approximately five hours (out of a total of 12). There are a number of problems that need to be addressed, in particular: How do you teach a new programming language (Python) to ten year old children, who have not done much programming before, in a few hours? How do you keep them interested? Can they actually use the complex functions that are provided by the libraries coming with the robot?

As mentioned before, the programming language we chose is Python. Although a Python wrapper for the C++ libraries has been provided by the robots' manufacturer, we still found that most functions are too complicated to use in a course like ours. Therefore, we wrote additional Python functions for the basic functionalities of the robot, i.e., for moving a certain distance with a given velocity, turning to the left and right, turning by a given angle, and checking the sonar sensors. The set of commands for obtaining sonar sensor readings was simplified considerably by grouping the sensors into left (three leftmost sensors), front (two middle sensors), and right (three rightmost sensors), and returning either the actual distance reading from the sensor group, or alternatively a "boolean" value that tells whether there is an obstacle within a given distance or not. To keep it simple, this value was in fact not "boolean" as such, but rather "yes" or "no", which is very easy to handle in Python as well. As most of these commands were very similar to those used for doing experiments using the command-line interface, the children did not have any problems using a list of these functions provided for programming.

Before the students started to program, we gave a short introduction to using Python and programming in general, where we presented the basic concepts like variables, if-, for-, and while-statements, as well the template containing the robot initialization, which the students used afterwards as a skeleton for inserting their own code. We then went straight into practice, i.e., the students tried out the provided template and started adding their own commands.

In order to keep them interested, we decided to give them the task of programming the robot to "dance", i.e., they were supposed to choose a song they liked and choreograph the movement of the robot to the song. The only restrictions were the size of the stage (3m × 4m), and the length of the song (2 minutes). Originally, the idea was to start on an empty stage, and have a second part where the robot has to avoid some obstacles placed on the dance floor. We ended up with dancing without any obstacles, though, as the given time frame was too tight for programming obstacle avoidance. To make it even more interesting, it was decided to have a competition between the three groups at the end of the course. This was held at their school, and classmates as well as parents were invited to attend the performance. The students were encouraged to become creative by dressing up the robot for the competition.

Conclusion

The paper presented a robotics course especially designed for primary school students. It was held in October/November 2005, for a duration of six weeks, the students were about 10 years old and attend a private girls school in Auckland. The students were very enthusiastic to learn and participated in all the activities throughout the whole course very well. We were very impressed with their learning skills, adventurous nature and their ability to think outside the square with little assistance. During the course, their mental picture of robots changed significantly, the initial impression being mainly influenced by movies and television. The students found that today's robots are still far from achieving what they have seen on the screen, and that they are prone to many inaccuracies and errors. Yet, they also discovered that research can be challenging while being fun at the same time, and that they can achieve the set goals by working as a team.

Having a competition at the end was definitely a good choice, as the students had the freedom to use their own creativity on what they have learned. We would recommend to have a contest to anyone designing a robotics course, be it at primary school or university level. Particularly when working with the primary school students, we found that it helps considerably in keeping them interested in what they are doing, as they have a common goal in mind.

Programming the robot using Python worked quite well, at a basic level anyway. We found that the students wrote only sequential programs, i.e., they did not use loops or if-statements. We believe that there are two main reasons for this: Firstly, the time-frame was probably too tight for more sophisticated programming; we are positive that the students would have been able to use these commands given more time. Secondly, it was not really necessary to use any of these commands in the dancing contest. However, having a more complex competition goes hand in hand with allocating more time to the course.

To conclude, we were really impressed by the students' capabilities, and how they approach problem solving tasks when programming mobile robots, without being hindered by what they know about what computers and robots can or cannot do, which is probably an advantage they have over older students. As to whether the course was successful? We believe the following departing quote from one of the students says it all: "See you in 10 years!".

References

Ahlgren, D. J., & Verner, I. M. (2002). *An international view of robotics as an educational medium.* Paper presented at the International Conference on Engineering Education, Manchester, UK.

Beer, R. D., Chiel, H. J., & Drushel, R. F. (1999). Using autonomous robotics to teach science and engineering. *Commun. ACM, 42*(6), 85–92.

Billard, A. (2003). Robota: Clever toy and educational tool. *Robotics and Autonomous Systems, 42*(3-4), 259–269.

Blank, D., Meeden, L., & Kumar, D. (2003). Python robotics: an environment for exploring robotics beyond legos. *SIGCSE Bull., 35*(1), 317–321.

Bumby, K., & Dautenhahn, K. (1999). *Investigating children's attitudes towards robots: A case study.* Paper presented at the Third Cognitive Technology Conference (pp 391–410), San Francisco, USA.

Fossum, T., Haller, S., Voyles, M., & Guttschow, G. (2001). *A gender-based study of school children working with robolab.* Paper presented at the AAAI Spring Symposium on Robotics in Education, Palo Alto, California.

Kanda, T., & Ishiguro, H. (2005). *Communication robots for elementary schools.* Paper presented at the AISB'05 Symposium Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction (pp 54–63), Hatfield, UK.

Kumar, D., & Meeden, L. (1998). *A robot laboratory for teaching artificial intelligence.* Paper presented at the Twenty ninth SIGCSE technical symposium on Computer science education (pp 341-344), New York, USA.

Lalonde, J.-F., Bartley, C., & Nourbakhsh, I. (2006). *Mobile robot programming in education.* Paper presented at the International Conference on Robotics and Automation (ICRA) (pp 345-350), Orlando, Florida.

Lau, K. W., Tan, H. K., Erwin, B. T., & Petrovic, P. (1999). *Creative learning in school with lego programmable robotics products.* Paper presented at the

29th IEEE International Conference on Frontiers in Education (pp 26–31), San Juan, Puerto Rico.

Miglino, O., Lund, H. H., & Cardaci, M. (1999). Robotics as an educational tool. *J. Interact. Learn. Res., 10*(1), 25-47.

MobileRobots Inc. (2008). Retrieved 28 Dec 2008, http://www.mobilerobots.com

Nourbakhsh, I. R., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., Richards, S., & Wilkinson, K. (2005). The robotic autonomy mobile robotics course: Robot design, curriculum design and educational assessment. *Auton. Robots, 18*(1), 103-127.

Nourbakhsh, I. R., Hamner, E., Crowley, K., & Wilkinson, K. (2004). *Formal measures of learning in a secondary school mobile robotics course.* Paper presented at the IEEE Internationals Conference on Robotics and Automation (pp 1831–1836), Barcelona, Spain.

Qaiyumi, A., Jantz, S. D., Arroyo, A. A., Bagnell, J. A., & O'Malley, P. (1998). *Robots in the classroom: Using mobile autonomous agents to stimulate interest in science and engineering.* Paper presented at the Conference on Recent Advances in Robotics, Florida.

Rodger, S. H., & Walker, E. L. (1996). *Activities to attract high school girls to computer science.* Paper presented at the Twenty-seventh SIGCSE technical symposium on Computer science education (pp 373–377), New York, USA.

Salter, T., Boekhorst, R. T., & Dautenhahn, K. (2004). *Detecting and analysing children's play styles with autonomous mobile robots: A case study comparing observational data with sensor readings.* Paper presented at the 8th Conference on Intelligent Autonomous Systems (IAS-8) (pp 61–70).