A new approach to teaching robotics

Dylan Evans
Senior Lecturer in Intelligent Autonomous Systems
Room 2N13, Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England
Frenchay Campus
Coldharbour Lane
Bristol BS16 1QY

21 June 2004

1. Introduction

From this September, I will be running a new module called 'An Introduction to Robotics' for first-year engineering students enrolled on the new BSc in Robotics that has just been set up at UWE. The module will comprise 24 one-hour lectures and 24 two-hour workshops. This essay sets out some of the thinking behind this course.

2. Experiential learning

David Kolb is a distinguished writer on pedagogy whose book, *Experiential Learning: Experience as the Source of Learning and Development* has been highly influential since it was first published in 1984 (Kolb, 1984). Put very briefly, Kolb sees learning as a cyclical process in which the learner reflects on his or her concrete experience and formulates general abstract principles which may then be put to the test by active experimentation. The experimentation gives rise to further concrete experiences, and so the cycle begins again. When applied to teaching, this view of learning tends to be linked with an emphasis on setting problems for students to solve and then asking them to reflect on their experience, and a correspondingly lesser emphasis on more traditional teaching methods such as lectures.

At first, these ideas appeared quite attractive to me, as they seemed to chime well with the teaching style that I intend to adopt in the 'Introduction to Robotics' module that I will be teaching from September on. In this course, which is a first-year undergraduate module aimed at students with no prior experience in robotics in particular or engineering in general, I intend to proceed very differently to the standard way in which engineering, and robotics in particular, have been taught previously. The standard approach is theory-led, in the sense that it begins by presenting the student with abstract principles and large doses of complex mathematics. Only much later is the student encouraged to make use of these intellectual tools in his or her own work by, for example, designing and building a robot. This approach tends to favour students who are comfortable with theorising, and abstract thinking, and who are highly literate and numerate. Conversely, it tends to go against those who are good with their hands but may not be so good at writing essays or solving complex mathematical problems. Undergraduates who study engineering at UWE tend to fall into the latter category, and may therefore find the standard approach to teaching their subject difficult. In the 'Introduction to Robotics' course that I am currently preparing, however, students will start building real robots from day one. The abstract principles and the various mathematical tools that play an important part in robotics will be introduced as and when students need them to complete a given task. For example, students may find that, in the course of programming a robot to explore new territory and then find its way back home, they need to learn some trigonometry. Then, and only then, will they be presented with the various functions, laws and identities that make up the core of trigonometry. It is my hope that this kind of theory-in-context approach will help UWE students to find it easier and more enjoyable to acquire the skills and knowledge required for robotics than the standard theory-first approach.

Despite the apparent similarity between the approach I have decided to adopt in my 'Introduction to Robotics' course and Kolb's ideas of experiential learning, it would be wrong to describe my course as 'Kolbian' (or whatever the adjectival form of Kolb is). Kolb's ideas are much more specific than my initial description might suggest, and we must be careful to distinguish between his specific claims and the more general idea that it is good to involve the student in practical hands-on activities rather than simply lecturing them. Kolb cannot claim credit for the latter idea, which is at least two and a half millenia old. The Chinese sage Confucius, writing in the fifth century BC, observed:

> “Tell me, and I will forget. Show me, and I may remember. Involve me, and I will understand”

This idea is so intuitive as to be almost common sense, and that is all that my approach to teaching robotics relies upon for its justification and design. An appeal to Kolb would be fatuous and misleading.

It would be misleading since my approach to teaching robotics explicitly disavows a central principle of all the various Kolbian approaches to teaching; problem-based learning. It is important to be precise here, so I'll give my own definition of problem-based learning: it is an approach to teaching and learning in which the student is forced to infer the abstract principles for him or herself. In my approach to teaching robotics, I give the students hands-on exercises to complete, but I don't expect them to be able to come up with the theory for themselves. It has taken thousands of years for the human race to gradually distill mathematical intuitions and insight into elegant formulae, and hundreds of years for great minds to transmute the concrete experience of collapsing bridges and exploding steam-engines into the beautiful equations of structural engineering and thermodynamics. It would be ridiculous to expect unexceptional students to be able to accomplish the same feat in a few hours. It would also be a waste of their time and effort. Science progresses because the fruits of the intellectual labour of previous generations can be passed on to subsequent generations without the need for re-growing. Isaac Newton famously said that he was only an intellectual pigmy when compared to previous generations of natural philosophers, but that he could see further than anyone because he stood on the shoulders of giants. My students are pigmys too – let us not be afraid to admit it – but they can see further than Newton, because they can be taught in five minutes things that it took Newton a lifetime to discover. They should not be forced to reinvent the wheel.

Yet that is exactly what problem-based learning does. And, predictably, students hate it. A recent study conducted by researchers at the Institute of Education in London found that students are eight times more likely to drop out of courses that have adopted problem-based learning in place of traditional lectures. The students complained that they were not being 'taught' and that teachers did not know what they were doing (Utley, 2004). While most educationalists argue that students will eventually acclimatise to the novel methods of problem-based learning, this study suggests that they do not – rather, they simply drop out. The students who do not drop out tend to get similar marks to those who are taught in more traditional ways, but they enjoy their courses far less (Utley, 2004).

3. Use of Lego Mindstorms in teaching robotics

As little as four years ago, low-level robotics courses were almost entirely theoretical, because it was too expensive to provide enough equipment for everyone to get direct practical experience. Thanks to the ever-decreasing cost of computing, however, mobile robots are now cheap enough that it is feasible to provide a class of undergraduates with enough equipment to enable all of them to get hands-on experience of building robots. Of the various off-the-shelf kits that may be purchased, my favourite is the Lego Mindstorms Robotic Invention System, which retails for around 170 pounds sterling.

3.1. What is Mindstorms?

Mindstorms is the name that Lego gives to their range of robotic toys. The basic kit retails for around 170 pounds sterling, which may be supplemented with other pieces sold separately. Inside the basic kit is a variety of Lego components, ranging from the basic building blocks that many adults today remember from their childhood, to gears and axle accessories that can be combined to construct complex steering assemblies. The most important component is the RCX brick, a yellow Lego block about the size of a cigarette packet that contains a microcontroller chip. Robots of many different shapes and sizes can be constructed by attaching sensors, motors and other components to the RCX brick. Programs can then be written on a computer and downloaded to the chip on the RCX, where they are stored for future use by the robot.

The blurb on the box states that Mindstorms is intended for children aged 12 and above, but this figure on its own is highly misleading. In my capacity as a Science and Engineering Ambassador I have taken Mindstorms into primary schools and witnessed children as young as seven using it without much difficulty. I have also listened to papers delivered by sixty-year old professors at international scientific conferences in which they describe their use of Mindstorms to conduct advanced research in robotics. Mindstorms is used by a number of universities around the world to teach robotics, and the average age of people owning their own Mindstorms kits is over thirty. In other words, Mindstorms approaches the ideal of a 'low-floor, high-ceiling' edutainment system that is simple enough for children to use yet also flexible enough to keep adults engaged and provide a research tool for professional scientists.

One of the key reasons for the wide age-range of Mindstorms' users is that a wide variety of different computer languages can be used to program the RCX. Lego provides a very simple programming language with the commercial version of the Mindstorms kit, and another slightly more complex one with the educational version of the kit, but for a number of reasons neither of these is really suited to undergraduates or more advanced users. For one thing, both of these languages are icon-based rather than text-based. In other words, users write programs by clicking on various icons in a GUI (graphical-user interface) and dragging them across the screen. This is fine for young children, but frustrating for those who already have some experience in text-based programming languages such as C or C++. Also, the icon-based languages supplied by Lego lack important features such as floating-point variables (they are restricted to integers) which are vital for programming mobile robots.

Thanks to the effort of hundreds of hackers, there are now a variety of text-based languages available for programming the RCX, including complete versions of C and Java. These are all unofficial (that is, they are not supported by Lego), and were only made possible after the hardware details of the RCX internals were reverse-engineered by Kekoa Proudfoot, a hacker based at Stanford University.[1] These languages are all

1 See http://graphics.stanford.edu/~kekoa

Open-Source, and can be downloaded for free from such Open-Source clearing-houses as Sourceforge. [2] This makes them attractive for financial reasons as well as technical ones. I intend to use one of these languages in my planned robotics course at UWE. The language I intend to use is LegOS, which allow developers to write C and/or C++ code for the RCX. This has the advantage of including floating-point variables and other advanced features, as well as providing the students with experience in the very languages they will use if they do any programming when they leave UWE and enter the workplace.

3.2. How do I plan to use Mindstorms in my teaching?

From this September, I will be running a new module called 'An Introduction to Robotics' for first-year engineering students enrolled on the new BSc in Robotics that has just been set up at UWE. The module will comprise 24 one-hour lectures and 24 two-hour workshops. The workshops will lead the students through a series of activities designed to take them from scratch to a basic practical knowledge of robot design. Almost all of these activities are based around the Lego Mindstorms Robotic Invention System, which will thereby provide a consistent hardware platform throughout the year.

Although the use of a single hardware platform throughout the course will provide a degree of consistency, there is a danger that it might also limit the range of potential learning experiences available to the student. Thankfully, this risk will be reduced in this course because the reconfigurable nature of Lego means that a wide range of different robot bodies can be constructed. This gives Mindstorms a considerable advantage over most other robots used in education, which generally have fixed bodies and which are therefore only suitable for teaching programming. With Mindstorms, students can explore the mechanical challenges of building robot bodies as well as the computational challenges of programming robot minds. Mindstorms thus allows students to explore one of the most fascinating and important aspects of robotics – the interaction between the physical design of the robot and its software. Students will discover that many problems in robotics can be solved by making small mechanical changes to the body of the robot rather than by introducing complex subroutines into the program.

At the beginning of each workshop, students will receive a handout describing the task for that session. They will then work in groups of two or three, each with its own Mindstorms kit, to complete the task. My role, as the teacher, will be to act as a mobile learning resource, on hand to help groups that require assistance, while not interfering with those who are progressing well on their own. As already noted, this approach to teaching has a superficial similarity with the approach pioneerd by David Kolb. However, the students will not be forced to rediscover the abstract principles of engineering for themselves on the basis of their experience with Lego, but will rather

2 See http://sourceforge.net

be encouraged to apply these principles to the practical activities they are engaged in.

3.3. Mindstorms and constructionism

It does not require an advanced knowledge of pedagogical theory to see that Mindstorms embodies a distinctly constructionist philosophy of learning, one in which great emphasis is placed on the need for students to 'get their hands dirty' by experimenting with things for themselves.[3] In fact, constructionist theories of learning played a direct role in the creation of Mindstorms, which was originally developed by a partnership of the Lego group and the Massachusetts Institute of Technology (MIT) in the 1980s, and named after a book by an MIT Professor called Seymour Papert. Papert was a long-term collaborator with Jean Piaget, one of the most influential proponents of constructionist approaches to learning.

Not everyone likes to learn by doing. Students differ in their learning styles, with some preferring to 'sit and think' and not to 'poke and see'. A course based around Mindstorms will favour students whose learning styles are more hands-on and may alienate students of a more abstract disposition, but there are several reasons why this is not a grave problem.

Firstly, this course is aimed at precisely those students who are 'good with their hands' but do not excel at the more abstract kinds of exercise which are the staple diet in more traditional engineering courses. Students who like dealing with abstractions can still choose more traditional engineering courses, which tend to put off other students who have difficulty seeing the relevance of the mathematical techniques which are often introduced without any practical context. A traditional course in programming, for example, might start by getting students to write a program that sorts numbers or draws a geometrical pattern. Only much later do students learn to build more useful programs out the rather dry components they learnt at the start. In this course, by contrast, students will be writing programs to make real robots move around, from the very first week. Not only are the results of their programming more visible and more engaging from the very start, but debugging the program is also much easier when the output is robot behaviour (movement and flashing lights) rather than a series of numbers on a screen.

Secondly, if students who are not 'good with their hands' take this course, they will be forced to develop the manual and visualisation skills that are vital for engineers. Withou more traditional engineering courses, there has always been a risk that students of a more conceptual bent might be able to pass with flying colours by

3 'Constuctionism' must be distinguished from " the kindred, but less specific, family of psychological theories that call themselves contructivist" (Papert, 1991). According to Seymour Papert, "constructionism shares constructivism's connotation of learning as 'building knowledge structures' irrespective of the circumstances of the learning", but then "adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe" (Papert, 1991).

relying exclusively on their abilities to reason about abstract principles, resulting in graduate engineers who have no feeling for the messy realities of real-world construction.

3.4. Mindstorms and abstraction

Although the course does place great emphasis on hands-on experience, it will also encourage students to reflect on what they are doing and to derive more abstract principles from the practical activities they are engaged in. This will be achieved by means of carefully-designed questions for discussion that will punctuate the task descriptions on the handouts distributed in the workshops. By grounding all these questions in a physical object – a robot that students can constantly manipulate and observe directly – the abstract thinking they encourage will be constantly referred back to and tested against the real world. I hope that this 'physical anchor' will help students feel safer when venturing into the airy realms of speculation.

Robots are wonderful 'tools for thinking with'. They are 'philosophical toys', embodied thought-experiments, physical prostheses for the mind. They raise all sorts of questions about the nature of machines and animals, and ultimately about ourselves. I intend to take full advantage of the power of robots to help people think through abstract questions by encouraging my students to draw comparisons between the components of their robots and other things such as the organs of animals and the pieces of very different machines. Students may draw inspiration from other machines and from animals when designing their robots, and this in turn may lead them to ask new questions about animals and machines. Their Lego blocks will be transformed into versatile tools for modelling a wide variety of natural and artificial systems. For example, students will be encouraged to discover more about how animals find their way back to their nests/burrows/etc and to write programs based on the same principles that will enable their robots to navigate.

Robots can also provide an exciting means of teaching what is otherwise usually very dull – the techniques for gathering data about the performance of a machine, and for analysing that data. Such techniques are a fundamental part of any engineer's skillset, but they are usually taught by means of context-free exercises in statistics. By contrast, students in this course will be required to use the sensors onboard their robots to collect data about the robot's performance, to find a way of transferring that data to their computers, and to analyse that data in such a way as to derive suggestions for improving the robot. Once again, the robot will provide a familiar and meaningful context for an otherwise abstract and disembodied task.

Finally, students will be encouraged to think about the differences between the real world and the virtual world by using a range of computer programs that simulate a variety of Lego robots. Students can program these virtual robots and then watch them run around in the virtual world of the simulation, then download the same program into the real robot and compare its performance to that of the virtual robot.

This will introduce the students to one of the most interesting debates in contemporary robotics – the relative merits of using computer simulations and real phsyical systems when designing new robots.

4. Possibilities for further research

As this course is still in preparation, I will not be able to tell whether the various ideas I have described here will work out in practice until I start teaching in September. No doubt there will be many unforeseen problems, but there will also be serendipitous discoveries and new ideas that occur to me only as I deliver the course. The first year of the course will therefore be as much of a learning opportunity for me as for the students. The number of students enrolled will be kept low for the first year, which will suit the experimental nature of the first run of the course.

Bibliography

Baum, D., Gasperi, M., Hempel, R., and Villa, L. (2000) Extreme Mindstorms: An Advanced Guide to Lego Mindstorms. New York: Apress

Ferrari, M., Ferrari, G., and Hempel, R. (2002) Building Robots with Lego Mindstorms. Rockland, MA: Syngress Publishing.

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

Papert, S. (1991) 'Situating constructionism', in *Constructionism*, edited by Idit Harel and Seymour Papert, Westport, Connecticut: Ablex Publishing Corporation.

Utley, A. (2004) 'Problem mehtod has high dropout', *Times Higher Educational Supplement*, 18 June 2004.

Wakeman-Linn and J., Perry, A. (2002) 'A proposal to to incorporate LEGO Mindstorms into an introduction to engineering course.' *Proceedings of the 2002 ASEE/SEFI/TUB Colloquium*.

--

Dylan Evans
Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England
Frenchay Campus
Coldharbour Lane
Bristol BS16 1QY