

A Comparison of Robots Used in a Senior Level Robotics Course and Senior Projects

Leslie Fife, Timothy Stanley, Anuj Sehgal, Sandip Kumar
School of Computing, Brigham Young University Hawaii
Email: {ldfife, tim, anujs, sandip}@cs.byuh.edu

Abstract: *After a course in Artificial Intelligence, many students want to go further. In our case, this translates into an elective course in robotics as well as senior projects and independent student research. While a purely theoretical robotics course is possible, we prefer a course where students further their understanding of robotic principles through labs using functional robots. The problem faced is the large variety and cost of the robotic systems available. In this paper we examine the robotic systems used in our robotics course, senior electives and independent student research. We also discuss an autonomous underwater vehicle designed and built as a direct result of these classes. The relative advantages and disadvantages of each are discussed.*

Keywords: Computer Uses in Education, Robotics

1. Introduction

Beginning in the Fall of 2003, the robotics course at Brigham Young University Hawaii has included projects utilizing robotic systems. The prerequisite for Robotics is the Intelligent Systems course. The Intelligent Systems course requires students to program agents which interact in a simulated environment. For the Intelligent Systems class, this simulated approach works sufficiently well. However, those students continuing on to the elective robotics course look forward to testing knowledge gained on autonomous (robotic) systems.

There are several reasons to include actual robots in a robotic course. Perhaps the most common reason mentioned in the literature is retention (Becker, 2001; Fagin & Merkle, 2002; Fagin & Merkle, 2003). However, retention was not the goal for including functional robots in our senior level robotics course. By the time senior electives are taken, retention is simply not an issue. Others suggest that robots may be used to teach object oriented programming (Becker, 2001; Lawhead, et al, 2002). This is also not our purpose for including robotics as objects have already been taught earlier in our course sequence.

Our purpose in introducing robotic systems is to help students visualize and experience the issues involved in robot design and use. By actually programming a robot to complete certain tasks, the student is required to understand robotic concepts. In addition, the student is required to integrate several systems (propulsion, sensing, etc.) while working alone or in small teams. Basic computer science concepts introduced earlier in the course sequence and new concepts are integrated. The additional concepts encountered include systems integration, embedded systems, electronics, mechanical engineering, physics and mathematics.

When deciding to use robotic systems in a course, the problem becomes which robotic system to use. The choice is actually quite large, with a wide range of features and prices. In addition, rarely is money not an object and so it is important to know which systems actually work well for use in a specific course. That is the focus of this paper.

In Section 2, we briefly discuss the literature on robot use in the classroom. In Section 3 we discuss the robotic systems used in our robotics class, senior projects and independent student research. We discuss both the advantages and disadvantages of each system used. This section ends with a brief description of the autonomous underwater robot built to compete in the 7th Annual AUV Competition held in the summer of 2004 in San Diego, California (www.auvsi.org). We end the paper with a discussion of our conclusions and future work in this area.

2. Literature

It is really no surprise that several individuals have used robots in the classroom. Robots are interesting to virtually every computer science student. Working with robots appeals to most students.

At the United States Air Force Academy, robots were tested in CS1 (Fagin & Merkle, 2003). The goals included increased learning and an increase in the number of students selecting computer science as their major (Fagin & Merkle, 2002). The experiment failed to show that robots either increased learning or caused an increase in majors (Fagin & Merkle, 2002; Fagin & Merkle, 2003). The study suggests the lack of a simulator to allow students to work in the dorm during study hours in their highly structured environment as the chief cause (Fagin & Merkle, 2002; Fagin & Merkle, 2003). This assumption seems to be validated by work at the United States Military Academy at West Point. A simulator was available for practice outside of the classroom and results were reported as quite positive (Flowers & Gossett, 2002).

In CS1, robots have also been used to teach object oriented programming. The use of robots to teach object oriented programming is suggested by (Lawhead, et al, 2002). Robots have been used to good effect in CS1 using the Karel robot (Becker, 2001). Robotic simulation (software simulation without accompanying robots) has also been used to teach object oriented programming in CS1 (Kuhrt & Mazur, 2000).

It is a little surprising that more examples of robots in upper division courses have not been found. However, robots are reported to have been used to teach neural network concepts in an artificial intelligence course (Imberman, 2003). Robots have also been reported in a robotics class (Harlan, et al, 2001). Harlan makes the case for using robotic systems in a robotics course, rather than having a purely theoretical course (Harlan, et al, 2001). Our focus then is not on if a robot should be used.

3. Robot Use and Evaluation

The question then becomes “which robot?” This section reports our evaluation of the robotic systems we have used as a guide to others wanting to use robots in a computer science course. This applies especially to courses in robotics and related areas.

One important consideration is the user interface of the robotic system. It is not desirable for students to get sidetracked by hardware issues unrelated to their use in a specific course. However it is very desirable for the student to have the ability to experience system control and integration. We consider among the important factors to

be cost, hardware interface, programmability of the robot and the programming interface, hardware support and ease of extensibility.

3.1. Classroom Robots

In Fall 2003 a senior undergraduate robotics elective was offered. The focus in robot selection was to find a system that could be used to teach foundation robotics concepts without having to deal much with the hardware. Some level of design flexibility was desired but, mostly the focus was on the robot software development.

For our robotics course we chose the ER1 robot from Evolution Robotics (www.evolution.com) as our primary platform. The reasons were that optical recognition was built in and the ability to interface with the robotic hardware with a high level scripting languages like Perl was available. The Lego® Mindstorm™ was also used.

3.1.1. ER 1 Robot. The ER1 robot is manufactured by Evolution Robotics Inc. and is distributed as a hardware and software robotics kit. The ER1 robot comes bundled with its own proprietary software which requires a laptop with Microsoft Windows. The laptop is not provided and adds to the overall cost.

ER1 features include (Evolution, 2004):

- Vision – ability to capture, recognize and identify objects and locations.
- Hearing – rudimentary speech recognition abilities and sound decibel level detection.
- Speech – able to talk with a text to speech conversion.
- Networking – if coupled with a wireless network card.
- Teleoperation – it can be operated over the network.
- Autonomous Mobility – movement parameters can be specified allowing for autonomous movement.
- Gripping – an option gripper arm is available.
- IR Sensing – IR sensors can be interfaced with the robot to add further autonomous behavior.

All these features are accessible through a GUI programming interface. This interface, named the Robot Control Center (RCC), enables the ER1 to recognize objects, colors, sounds, send and receive email, schedule operations, take pictures and videos, navigate autonomously or via remote control and play sounds (Evolution, 2004). These individual behaviors can be combined into complex behaviors by creating programs in the RCC by selecting behavior choices from drop down menus.

Analog and Digital I/O pins are provided on the Robot Control Module (RCM) providing an interface to devices not supported by the ER1 as part of the RCC. These pins run at a maximum voltage of 5 Volts which may limit which devices can be added.

The RCC interface can be very restrictive because it does not support any higher mathematical functions. In fact, the RCC does not allow any customized processing to occur on the data that is gathered by the robot's sensors. In order to overcome this disability, a command prompt based API is provided. The functions in this API can also be called by high level languages that provide a telnet interface, such as C++, Java, Perl, Python or Visual Basic.

Even though the API provides access to these functions, the program capabilities still remain highly restrictive because the programs are dependent on the simple responses provided by the API functions. Direct access to hardware objects on the ER1 is not possible because that would require access to proprietary design information to write device drivers. Thus, the ER1 robot is not able to perform highly complex tasks, but is adept at performing simple tasks.

An advantage of the ER1 robot is that the RCC has built in vision capabilities that allow the robot to use any webcam for guidance. The RCC provides the most common features, such as color recognition, object tracking and distance measurement, using the webcam for vision. For most simple applications, this vision system works well, but it is highly dependent upon the ambient conditions. Image dumps are not supported by the current version of the RCC software.

Another feature of the ER1 is that since it uses stepper motors for movement, it is able to measure the distance it has moved. By doing so, it is able to function on a two-dimensional coordinate system based navigational algorithm.

The projects we accomplished with this robot are as follows:

- Traveling a path using remote displays and control (Telepresence) simulating a remote rescue.
- Traveling a predefined path using “dead reckoning”.
- Traveling the same path but with objects along the way to recalibrate.
- Adapting and interfacing sensors and programming the robot to follow a path (line on gymnasium floor).
- Foraging for food (Collecting empty drink containers from an area while avoiding obstacles.)
- Mapping and solving a maze.

The computer has a nice facility to interface with additional sensors, which we used to connect photocells for line following and IR range finders for the maze project. Because of the expense of the ER1 (about \$300 plus a laptop), we had students work in teams. The main limitation we found in this robot was the inability to multitask. Another potential problem was that students enjoyed working with the robots so much that it was easy to neglect their other coursework. Figure 1 shows students using the ER1. The ER1 is backed with a warranty and customer support.

Even with some shortcomings the ER1 robot is an excellent platform to introduce practical robotics to computer science students. What the ER1 loses in its lack of hardware multitasking and extensibility, it makes up in its ease of use. As such this kit makes an effective teaching tool

3.1.2. LEGO® Mindstorm™ Robot. The LEGO® Mindstorms™ kit has been around for a long time and is one of the most popular robotics experimentation kits available. This kit was not used as a regular part of the course curriculum but was made available to all the students in the robotics class to conduct individual experiments in areas that interested them. Figure 2 shows a gripper arm made with a LEGO® Mindstorms™.

One of the greatest advantages that a LEGO® Mindstorms™ kit provides is that small blocks can be put together quickly to form any desired model. This ability combined with the electronics that are provided in the kit enables students to experiment with robots that range from line followers, walkers, grippers to any other design they



Figure 1. The ER1 robot in a maze solving project

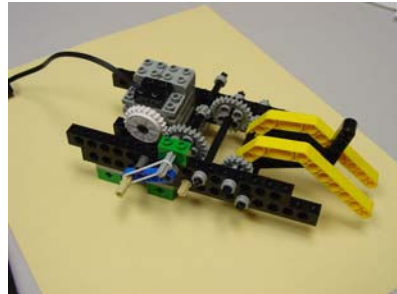


Figure 2. Gripper arm made with LEGO® Mindstorms™

have in mind.

LabView is the standard programming environment provided along with the kit. LabView provides a graphical flowchart environment to create programs and complex tasks for the robots that can easily be compiled and downloaded. The kit also comes with DLLs that can be used in Visual Basic Programs to write more complex programs or interface new hardware. In addition, other open source programming languages are available for use.

The LEGO® Mindstorm™ provides access to hardware level objects in most programming environments and also supports hardware multitasking. This enables the robots to perform complex tasks, while keeping the programs relatively simple. Another advantage of this kit is that it has a large user base. A major drawback of this platform though is the lack of any vision system that can work off the robot controller supplied with the kit.

Some of the experiments that were conducted were:

- Bump Sensor Robot
- Line following robot
- Walking robot
- Various gripper arm designs.

With a cost of \$200, availability of many expansion packs, and the ability to experiment with many different aspects of robotics, this kit is a very useful educational tool. This tool is good for a basic introduction to robotics.

3.2. Senior Project Robots

A variety of robots have been used in senior projects and independent research conducted by students in our department. Robotic platforms include OWI WAO-G Robotics Kit and several OOPic based robots.

Research includes:

- Investigation of robot navigation.
- Underwater navigation under location uncertainty.
- Cooperative multi-robot mapping.

3.2.1. WAO-G Robot. The WAO-G Robot kit is manufactured by OWI. This kit is designed for students with little or no experience in robotics. WAO-G is controlled by a Fuzzy IC which is designed for fuzzy logic navigational control. This robot can be programmed over a serial connection to a computer using QuickBasic or with push-button switches on its back. The amount of memory on this robot is very small and as such it can only hold programs that are 48 steps long and have a maximum of 16 control loops. WAO-G is also equipped with two touch sensors that are located in the front. Figure 3 shows the WAO-G.

Even with its limitations, the WAO-G can be used effectively to perform simple tasks such as maze solving. The design of the WAO-G does not allow for extra hardware to be added to the robot and is only useful to teach very basic navigational principles and conduct simple robotic experiments. However, its price of \$150 makes it too expensive for these simple tasks. This robot is shown in Figure 3.



3.2.2. OOPic Base Boards and Kits. The OOPic, manufactured by Savage Innovations, is a PIC Micro based microcontroller for robots and embedded systems. There are many kits that use the OOPic as its main processor board. Savage Innovations provides the OOPic in three configurations, the OOPic-C, OOPic-R and OOPic-S. All these boards are designed to be suited to specific tasks. Portland Robotics has also made a board that is based on the OOPic chip. Portland Robotics uses its own design in the Mark III Robot Kit.

The OOPic-S board is the most adaptable out of all the configurations provided by Savage Innovations because it has the maximum number of I/O lines. The only advantage that the OOPic-R board has is that it has RS-232 as well as TTL serial ports on it, as compared to only TTL on the OOPic-S. But this shortcoming can be easily overcome by using a level shifter. The Portland Robotics design also has an RS-232 port.

All versions of the OOPic can be programmed in Visual Basic, Java or C++. These microprocessors have built-in A/D converters, digital and I2C (IIC) inputs and serial I/O ports. The OOPic also supports hardware multitasking and the ability to link many devices together, as though they were connected in a circuit. OOPic also provides software objects for many sensors, actuators, motor controllers, etc. Supported hardware is integrated easily. Custom or unsupported hardware is accessible through the programming interface.

Being a PIC Micro architecture, the OOPic like all other PIC devices does not have a high clock speed. This makes it unsuitable for performing signal or image processing. But, its ability to network with other OOPics and I2C devices makes it easy to integrate image and signal processing by offloading those tasks onto networked processors. As such, the OOPic is highly suited to many robotic applications. This is a good microprocessor to use in a senior robotics class since it not only introduces basic electronics concepts but also makes the complexities of hardware integration easy because of provided hardware objects.

The Mark III kit is one of the best kits based on the OOPic because it provides robot chassis, servos, IR sensors, line sensors and an OOPic for only \$100. The provided electronics and chassis make it possible to organize this kit in many possible ways, to suit the task at hand. Moreover, the variety of accessories for the OOPic makes extending these robots simpler. The Mark III robots are being used as the robotic platform for the cooperative multi-robot mapping research. In order to enable communication between the robots, these Mark III robots have been equipped with Bluetooth adapters that are controlled via RS-232 ports on the OOPics. Figure 4 shows the Mark II performing a line following task.

	
<p><i>Figure 3. The WAO-G Robot</i></p>	<p><i>Figure 4. Robot involved in Mapping Research</i></p>

Research on underwater navigation under location uncertainty is also being conducted. The developed algorithm is based on a neural network approach for landmark discovery in underwater environments to aid navigation under location uncertainty. The testing for this research is being simulated and will be tested by implementing the neural network of sonar sensors on an OOPic microprocessor.

OOPic microprocessors coupled with kits like the Mark III provide many advantages. These microprocessors are easy to use, provide a good introduction to basic electronics, allow students to be creative, and allow easy extension and inclusion of any additional hardware. They are also inexpensive. As such, the OOPic and kits based on this processor make an excellent platform to introduce robotics to computer science students.

3.3. Robot Competition

One set of Senior Projects went further than anticipated. Two students decided to participate in the *7th Annual AUV Competition* (www.auvsi.org). This competition is for autonomous underwater robots. This required the students to put together what they learned in several courses as well as new information. The integration of their academic experience clearly led to additional learning. While they did not win, they returned enthusiastic and ready to go again. Figures 5 and 6 show our AUV competition vehicle which was designed with the goal of being a low-cost autonomous underwater vehicle that could be used in the department to teach underwater robotics and conduct research related to underwater robotics.

None of the robotic systems used in the classroom were utilized in this project, however, the OOPic was used as the systems primary processor. In addition to the basic systems on the AUV, a pressure sensor, digital compass, CMUCam2, passive sonar, motor controllers and twin-axis accelerometers were integrated with and controlled by the OOPic to accomplish autonomously a contest specified task. As a result of the design process, the students also learned about design and teamwork.



Figure 5. Student built AUV – Rear View



Figure 6. Student built AUV – Front View

Final cost for the Student built AUV was approximately \$600. This entry was the lightest and least expensive entry in this year’s competition. Successful operation of this AUV demonstrated that the OOPic processors can be extended for more complicated tasks (Sehgal, Kadarusman & Fife, 2004).

4. Conclusions

All of the systems used have their own advantages and disadvantages. The primary systems we use are compared in Table 1. Our experience showed that students are reluctant to learn the new hardware. However, they do it because they want to experiment with their own designs. However, a good hardware interface is very important.

Table 1. Comparison of Robots

	ER1	Mindstorm	OOPic
Cost	\$300	\$200	\$100
Hardware Interface	15 Analog 8 I/80 Digital	3I/30 A/ D	35 A/D IIC Port
Programmability	Perl, C, Java	Labview, C, Forth	Visual Basic, C+, Java
Programming Interface	telnet	Infrared	Serial
Ease of Extensibility	Difficult	Poor	Easy
Customer Support	Good	Good	Good

For future classes the plan is to have sufficient robots that both team and individual projects will be possible. This will give each student an opportunity to show individual creativity, and allow students to get closer to the hardware. Which platform is selected will be based on our experience with these robotic systems as well as available budget. The ER1 will remain the primary platform to teach teleoperation and robot control based on computer vision. The ER1 also will continue to support team projects.

Clearly, this is not an exhaustive look at robotic systems available on the market. However, we show a good selection of the types of robotic systems available. As additional robotic systems are tested and integrated into the curriculum. A future report may report on their use.

Acknowledgments

Our thanks to the BYU Hawaii Student Research and Development funds for supporting the construction of our competition robot and participation in the contest.

References

- Becker, B. Teaching CS1 with karel the robot in Java. In *Proceedings of the 32nd SIGCSE technical symposium on Computer Science Education (SIGCSE 2001)* (Charlotte, NC, USA, February, 2001). ACM Press, New York, NY, 2001, 50-54.
- Fagin, F., and Merkle, L. Measuring the effectiveness of robots in teaching computer science. In *Proceedings of the 34th SIGCSE technical symposium on Computer Science Education (SIGCSE '03)* (Reno, NV, USA, February 19-23, 2003). ACM Press, New York, NY, 2003, 307-311.
- Fagin, F., and Merkle, L. Qualitative analysis of the effects of robots on introductory Computer Science education. *Journal on Educational Resources in Computing (JERIC)*, 2, 4 (Dec. 2002), 1-17.
- Flowers, T., and Gossett, K. Teaching problem solving, computing, and information technology with robots. *Journal of Computing Sciences in Colleges*, 17, 16 (May 2002), 45-55.
- Harlan, R., Levine, D., and McClarigan, S. The Khepera robot and the kRobot class: a platform for introducing robotics in the undergraduate curriculum. In *Proceedings of the 32nd SIGCSE technical symposium on Computer Science Education (SIGCSE 2001)* (Charlotte, NC, USA, February, 2001). ACM Press, New York, NY, 2001, 105-109.
- Imberman, S. Teaching neural networks using LEGO® handy board robots in an artificial intelligence course. In *Proceedings of the 34th SIGCSE technical symposium on Computer Science Education (SIGCSE '03)* (Reno, NV, USA, February 19-23, 2003). ACM Press, New York, NY, 2003, 312-316.
- Kuhrt, M., and Mazur, N. An improved robot simulation for teaching programming concepts. *Journal of Computing in Small Colleges*, 15, 5 (May 2000), 238-246.
- Lawhead, P., Duncan, M., Bland, C., Goldweber, M., Schep, M., Barnes, D. and Hollingsworth, R. ITiCSE 2002 working group report: A road map for teaching introductory programming using LEGO® Mindstorms™ robots. *SIGCSE Bulletin* 35, 2 (June 2002), 191-201.
- Sehgal, A., Kadarusman, J. and Fife, L. LUV: The Low Cost Underwater Vehicle. In *Proceedings of 7th International Autonomous Underwater Vehicle Competition* (San Diego, CA, USA, July 29 – Aug 01, 2004).
- Evolution Robotics Inc. Autonomous Mobile Robots – Evolution Robotics ER1 Robot Kit. <http://www.evolution.com/education/er1/> (Sep 7, 2004)