ROBOTICS SIMULATION AS A CROSS DISCIPLINE PROJECT IN ELECTRICAL AND COMPUTER ENGINEERING

Kyle Mitchell, Ph.D., William Ebel, Ph.D., Roobik Gharabagi, Ph.D
Center for Sensors and Sensor Systems
Saint Louis University

Abstract

Robotics is a discipline that cuts across disciplinary boundaries from automation in mechanical engineering through autonomy and mobility in electrical and computer engineering to cognition in computer science [1][2][3][4]. In an effort to instruct students in robot mobility and autonomy in the Electrical and Computer Engineering program at Saint Louis University, a robotics investigation environment has been developed that engages students in topics of robotics without requiring advanced knowledge of embedded systems. This environment primarily includes the sensor and drive characteristics of a mobile robot, but it can easily include signal processing and control topics. A primary goal of this effort is to help students link material from different courses to carry out multi-disciplinary open-ended projects.

Overview

The Electrical and Computer Engineering Department at Saint Louis University has been developing a Robotics Investigation Environment (RIE) comprised of an integrated software simulation and hardware platform. This environment integrates tangible hardware into classroom curricula. The RIE is currently being used in two courses with integration into a third planned for fall 2012. The current courses are Robotics Design, a senior-level elective, and Junior Design, a preparatory lab leading into the senior capstone design project. The course being integrated is Freshman Electrical and Computer Engineering. This course introduces the disciplines and topics of study in electrical and computer engineering.

Robotics Design is a three credit hour lecture course with a heavy experimentation component. In this course students learn to characterize and autonomously control a wheeled robot, build computational simulations, and characterize electromechanical systems through a series of lectures and targeted homework assignments. Students enrolled in this course are expected to have course work in microprocessor programming, object oriented programming, and linear systems.

This course is intended for students in engineering degree programs. Students are required to have knowledge of frequency domain signal analysis and software programming. The course touches on signal processing techniques to process raw sensor values and control topics to keep motors spinning at fixed rotational rates. These topics are combined to develop control systems that drive a target robot through an obstacle course autonomously.

In robotics design students are expected to investigate:

- Robot Control under selectable/non-ideal conditions.
- Sensor Calibration.
- Sensor and motor placement as they affect response time and stability.
- Sensor response time, motor response time and control loop iteration time and their affect on stability.
- How simulations are constructed in computational environments.

Through their studies in robotics design, students develop skills in:

- Discerning characteristics of signals and noise and their influence on operational dynamics.

- Representing information in a computational system and display processed information and results.
- Developing reasonable computational models for motors, sensors, and other robot hardware.
- Reading and interpreting data sheets to extract information important to design.

The integration of robotics hardware into a lecture course is traditionally hindered by:

- Required detailed knowledge of embedded systems design.
- Lack of ability to view and control parameters that limit study to a small set of observable variables.
- Lack of robotics specific simulation.
- Diversity of backgrounds.

Background

Students express the thought that lecture material without tangible experimentation is difficult to conceptualize. Robotics is a discipline capable of supplying projects to several disciplines in electrical and computer engineering [3][4][5]. One of the challenges of using robotics projects in course work is preparing students with the background or prerequisite knowledge to obtain meaningful information from the assigned projects. Traditional off-the-shelf robotics experiments either require detailed knowledge of embedded systems or do not allow access to the parts of the platform required to perform meaningful experimentation [6].

Robots are being used to engage students in learning in many disciplines [1]-[5]. The linkage between lecture topics and tangible results creates a feedback system where students can experience the outcomes of application of the theories they are studying. Robots are used to demonstrate the complexities of real-time embedded systems [1], and they reinforce the uses of digital design topics [2]. In this work a different tact is being taken. This work presents a technology that can be used to study robotics directly without the need to have advanced courses in digital design or embedded systems design. This facilitates the study of the interaction between sensors, motors, control algorithms and the real world as an intermediate course. This technology also lets the student experience the difficulties of real robotics while allowing study to continue in a more controlled environment of simulation. Students experience the linkage to tangible results while instructors can give projects in a variety of situations without having each of the situations physically realized.

Students do not feel they get the same experience out of pure simulation as they do from a hands-on hardware experiment. Getting access to parameters such as speed, sensor values, and drive angle can be difficult in a robot traveling across a table. The robot’s response to noise and environmental differences can be hard to appreciate from a simulation alone. This work presents an environment that is a mix of robotic hardware and software simulation that have been integrated using the National Instruments DAQ for interacting with hardware and Matlab as the back-end computational platform [7][8]. The combination of Matlab and the NI DAQ hardware form a hardware foundation that does not require advanced knowledge of embedded systems to perform meaningful experimentation. The software portion of the environment developed is a rewrite of certain Matlab DAQ Toolbox commands to interact with a simulated environment. This gives students an environment where they can develop one set of code and test it in both hardware and simulation by changing only a few lines of configuration code.

The developed framework allows students to investigate and design systems in a simulation environment and then transfer their designed systems to target hardware. This approach to interchangeable hardware and simulation gives students the feel of real hardware design while allowing them to work in a simulation environment. Facilitating the students to perform design work in a simulation environment allows them to complete algorithm development and initial testing without access to target hardware.

This instruction method is most effective when the simulation accuracy reflects the target hardware. For example, as an assignment in Robotics Design, the students are required to write a simulation module to replace the robot drive module in the current simulation environment. They are given a different drive wheel configuration and asked to develop equations of motion, then implement them in a module that interacts with the developed framework. This exercise is designed to give students an appreciation of how simulations can be used to gain insight into a design problem.

The development of this technology is meant to facilitate the following three goals:

- To allow the study of autonomous robotics without advanced knowledge of embedded systems.
- To allow autonomous robot experimentation without requiring construction of the physical environment or the physical hardware.
- To allow students to experience how simulation modeling can be used to test ideas that are subsequently transferred to physical hardware.

These goals have been evaluated through the use of homework and test projects given to students in Junior Design and Robotics Design.

Integration

The integration of the RIE into the robotics design course is described in this section.

The students have a series of assignments utilizing the RIE to reinforce the topics and skills studied during the lectures. In the first assignment they are asked to generate and plot a set of points representing a sine wave. They are then asked to quantify attributes of their wave such as, sampling rate, wave frequency, and sample duration. In the second assignment students are asked to develop a system that updates the position of a ball bouncing around the inside of a box. They are also asked to produce a graphical representation of the current state of the system. After completing the first two assignments they have the skills required to develop simulations of simple systems that progress in time and have graphical display of the system state.

In exercise three the students are given a multi-link arm and asked to derive equations for attaining certain attitudes of the end effecter. The knowledge gained through this exercise is that in robotics, like most designs, there are many ways to arrive at the same output [9], and that in over-constrained problems there is no one correct answer, but rather a family of correct answers.

The next group of experiments are designed to give the students the tools necessary to perform autonomous control of a physical robot. In exercise four they use the robot investigation environment to drive the motors and measure aspects of the motor system such as: control dead band, drive speed vs. speed command, and motor symmetry. In exercise five they are given two different distance sensors and their data sheets. Using knowledge extracted from the data sheets, the students exercise the sensors by taking static distance measurements. From these measurements the students are to characterize parameters such as: minimum sensor distance, maximum sensor distance, measurement linearity, and sensor noise vs. distance. In exercise six they attach one or more sensors to a mobile robot platform and drive the robot in an ad-hoc closed loop control. In this experiment the students are expected to be able to control the robot to maintain a certain distance from a straight flat wall for several meters.

The third group of experiments is designed to allow students to add to their knowledge while moving their activities to the virtual environment. In exercise seven the students perform the sensor and motor characterization on the virtual sensors and motors. In exercise eight they are asked to achieve a driving task in a simulated landscape. They are free to use any of the virtual sensors in any configuration to perform the assigned task. The students are encouraged to try different sensor placements and comment on the performance difference.

The annual IEEE Region 5 conference includes a student robotics competition [10]. This competition traditionally has a unique localization component that generates material for the fourth group of exercises. This year's sensing challenge was to navigate a field that was defined by 5 LEDs each of which was flashing at a different frequency.

The final group of exercises is a study in automatic control of the virtual robot. This activity involves study of PID controllers and how to tune them. The students practice by creating a single controller that is tuned to allow the robot to maintain a fixed distance from a wall that has straight sections, curved sections and outside corners. They are encouraged to think outside the box in observing and interacting with this virtual environment, but the solution must be a single self contained controller.

The students are tested 3-4 times during the semester to assess their performance in, and knowledge of, the material covered in the course. In the first of these examinations, students are given a robot simulation environment and a robot platform and are required to construct a robot position update algorithm and function. In the second examination they are given simulated data containing noise from a laser range finder. They are required to process the data and determine the position of the walls present in the room. In the final examination they are given a narrow hallway with bends and corners for the robot to navigate within. They are required to place sensors and develop control algorithms to allow the robot to navigate from one end of the hallway to the other in a minimum amount of time and without hitting a wall.

Methods

In this section we describe the robot simulation environment details.

The simulation is designed around having an interchangeable physical robot and robot simulation. One of the challenges in designing a set of material to be used semester after semester is to facilitate maintenance of the material in the world of planned obsolescence. This robotics investigation environment has been developed as modular as possible. The physical hardware has been individually modeled to be replaced or upgraded as replacement parts become unavailable. The one part of the environment that is not easily replaceable is the simulation back-end engine. The back-end of choice currently is Matlab. A new back-end would require development of a new event system compatible with both hardware and simulation as well as re-integration of all the hardware and simulation modules into the new event system.

The physical robot is a Budget Robotics Octabot [11] modified to use a Dimension Engineering Sabertooth Dual H-Bridge [12] and a National Instruments USB-6009 DAQ [13] box. The laboratory is stocked with a variety of sensors the students can attach to the physical robot. Several sensors have been modeled as part of this simulation environment. The IR Wall distance sensor simulation is a model of the Sharp GP2D12 sensor being filtered by a Phidgets IR Distance Sensor Interface Module 1101 [14]. The Acoustic Wall distance sensor is a model of Devantech Ultrasonic Range Finder SRF05 [15]. The IR Floor sensor was originally modeled after the Taos TRS1722 [16] reflective color sensor but has been reworked to model the Lynxmotion SLD-01 Single Line Detector [17]. The Lynxmotion SLD-01 is a (Optek OPB745) sensor element with custom electronic to allow attachment to a digital IO pin. The IR Emitter sensor is a simplified model of a photo diode circuit measuring ambient light levels. This sensor was designed to explore ideas relating to the 2009 IEEE Region 5 Conference Robotics Competition which requires a robot be able to perform localization based on detecting the position of flashing LEDs. The physical robot uses the NI USB-6009 to interface Matlab to the sensors and H-Bridge.

The simulation environment is based on a rewrite of portions of the Matlab DAQ Toolbox. *Robot_analoginput* is a rewrite of the *analoginput* class. Instantiation of this type creates an object with properties and member functions of *analoginput* necessary to perform robot simulations. It also creates a timer object that drives the simulation background processes.

Several of the Matlab DAQ Toolbox classes have been rewritten as part of this effort. The rewriting of these classes makes changing from the simulated robot to the hardware robot a simple matter of instantiating different functions during initialization time. *Robot_analogoutput* is a rewrite of the *analogoutput* class with properties and member functions of *analogoutput*. *Robot_digitalio* is rewrite of *digitalio* with properties and member functions of *digitalio*. To change a project from interacting with target hardware to interacting with the simulation environment simply requires that DAQ classes be instantiated with robot versions of the classes rather than the DAQ Toolbox versions. Figure 1 shows an example of how the start of a project can be set up to be interchangeable between the hardware robot and simulated robot.

```
%Initializing the environment:
%Uncomment these three lines to use
hardware
%ai = analoginput('nidaq', 'dev1');
%ao = analogoutput('nidaq', 'dev1');
%dio = digitalio('nidaq', 'dev1');

%Uncomment these three lines to use
simulation
ai = robot_analoginput();
ao = robot_analogoutput();
dio = robot_digitalio();
```

Figure 1: Code change for simulation verses hardware.

The simulation places a graphical version of the robot in a simulated field, providing the same stimulus and response as the physical robot. Figure 2 shows a close up of the simulation robot. The robot is drawn as a square with a user definable radius. The drive axel is assumed to be down the center of the robot with the drive wheels being the only modeled interaction with the ground. The line that runs right from the center is an indication of the driving direction of the robot. The octagons at the right side of the robot are an indication of the position of a set of sensors that are pointed down and are intended to see a high contrast line on the ground. The V indicators that open down in the middle of Figure 2 are an indication of the position and direction of a set of IR distance sensors meant to determine distance from a vertical service. The V indicators that open toward the right are an indication of the position and direction of a set of light intensity sensors. These sensors are intended to be used to find and track a light with a specific flashing pattern.

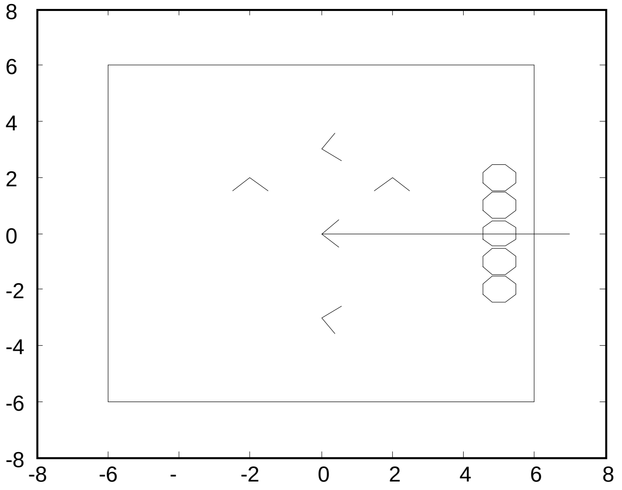

Figure 2: Close up view of simulated robot showing sensor placement.

Figure 3 and Figure 4 depict a robot driving in the simulation environment. The narrow hallway path the robot is navigating, shown in Figure 4, was developed as part of a final examination for Robotics Design. Figure 3 shows the IR Distance sensor reading from the robots trip down the hallway.

The simulation environment has been developed with configurable noise parameters. These parameters can be adjusted from a system

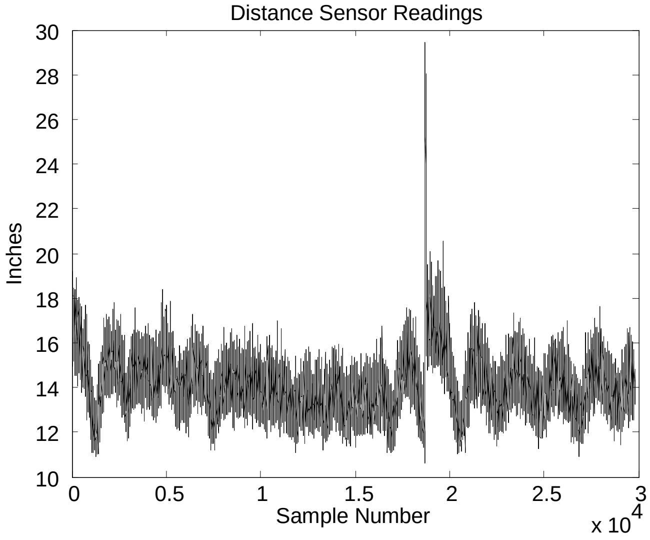

Figure 3: IR range data from a simulated robot navigating a narrow hallway.

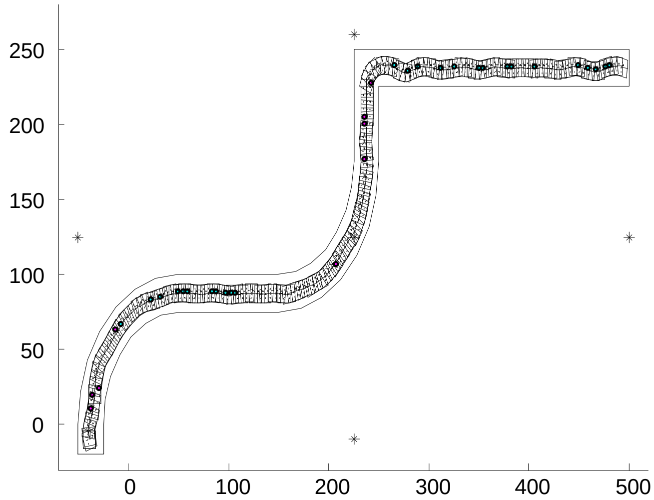

Figure 4: Travel path of a simulated robot navigating a narrow hallway.

with no noise present to a system that is dominated by the added noise. Noise is added to the system by specifying signal-to-noise (SNR) ratios for individual sensors and motors. The motor acceleration, wheel diameter, and max RPM are configurable. The configuration of sensor noise allows representative noise to be present in the sensor signals requiring similar signal extraction techniques to be employed in both the simulation and the target robot. The configurable motor parameters allow the simulation to model properties like unequal tire radius, unequal and limited motor acceleration, and out of round tires. These configurable motor parameters are also meant to challenge students in a fashion similar to that presented by real robots.

Figure 5 and Figure 7 show a part of a project currently under development. In this project the students are to navigate a robot down a hallway containing doors. They are to find the third door and turn into the room if the door is open. These figures depict the same controller implementation controlling the simulated robot in Figure 5 and the hardware robot in Figure 7.

Figure 5 and Figure 6 show a simulated robot driving past a wall with a closed door. Figure 5 shows the path the robot drove under control from a proportional controller. Figure 6 shows infrared distance sensor readings that the proportional controller used to produce the motor control commands.

Figure 8 shows the infrared distance readings as recorded through the NI DAQ box from the hardware IR Distance sensor of the hardware robot driving along a wall past a closed door. The same proportional controller that produced motor control commands for the simulation trial was used to control the hardware robot in this trial.

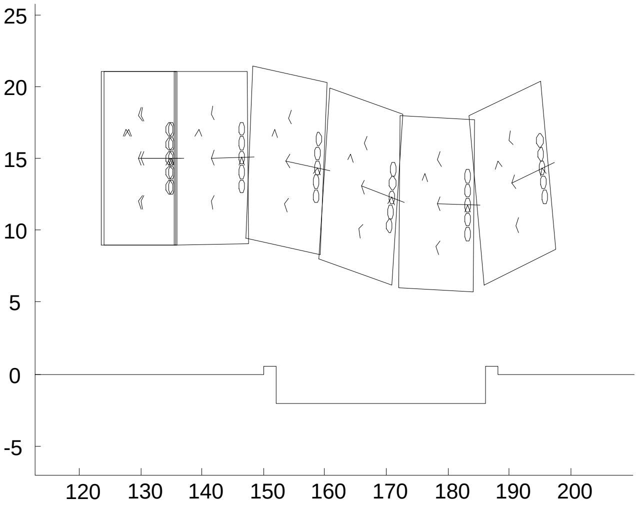

Figure 5: Travel path of a simulated robot following a wall past a closed door.

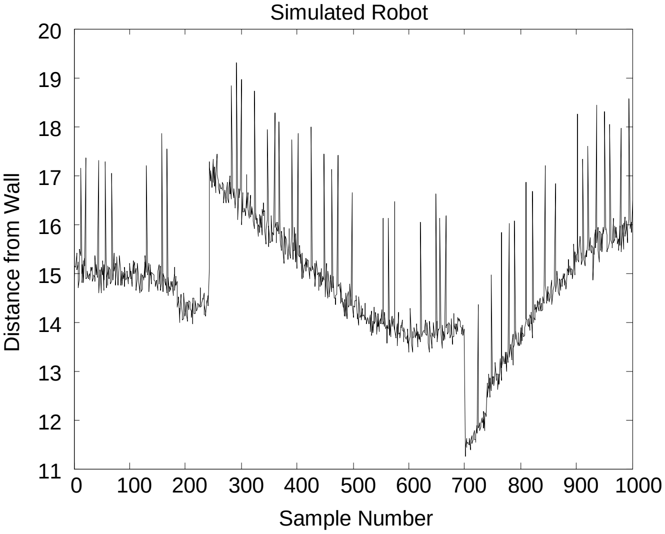

Figure 6: IR range data from a simulated robot following a wall past a closed door.

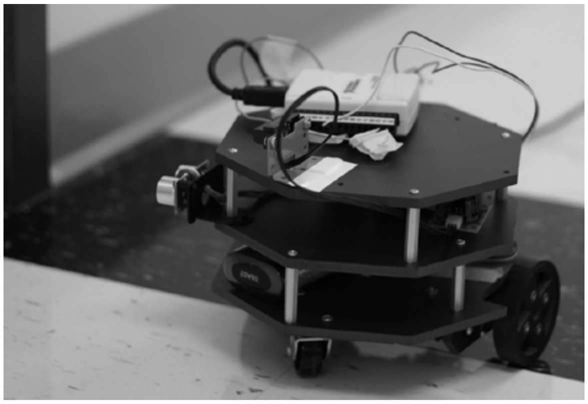

Figure 7: Target robot following a wall past a closed door.

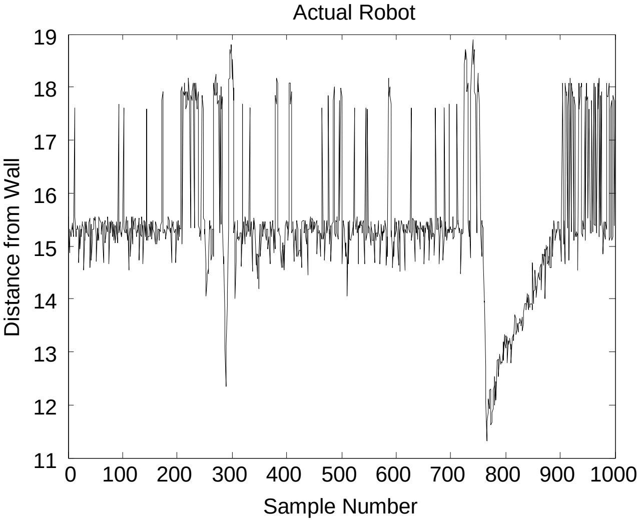

Figure 8: Sensor data from the real robot driving past a closed door.

By creating an environment where students can interact with simulation in the same manner as physical hardware, students can gain knowledge of the underlying system by experimenting in the simulation environment before being burdened with the difficulties of real hardware.

Results

In Junior Design the students were given a project where they were to design a drive algorithm where a robot drives down a hall looking for the third door. If the third door is open then robot is to turn in to the room otherwise it is to stop. Once they get their algorithm working in software they then transfer it to target hardware. They were graded on their ability to use the simulated robot to develop an algorithm and then on their ability to make minor modifications to their algorithm to get the target hardware to perform in a similar fashion.

In Robotics Design the students were given a series of projects that demonstrated different linkages and abilities of the technology and then given two exam questions where they were required to control a simulated robot to perform desired functions.

- The students calibrate the response of two physical distance sensors using laboratory equipment. Then they calibrate the response of the simulation of the same two distance sensors using the simulation environment and compare and contrast the calibration.
- The students are given a homework assignment to develop an automated controller that will drive the physical robot down a straight hallway, around a support pillar, and around a 90^o corner.
- The students are given an exam question where they take the knowledge gained from developing autonomous drive algorithms and asked to maneuver a robot down a simulated hall having the same characteristics of straight, round, and corner sections.
- The students are given a simulated arena with flashing beacons distributed around the area. They are asked to develop an autono-

mous controller algorithm that will park their robot in front of the beacon that is flashing the fastest.

The students performance in using the RIE was observed through homework exercises, project reports, project presentations, and examination questions. The results from analysis of their performance indicate:

- The developed technology performs well at introducing students to sensor control topics of robotics without requiring a detailed knowledge of embedded systems.
- The RIE can successfully facilitate many different scenarios without requiring the construction of a physical environment. The time required characterizing and modeling different electro-mechanical systems is similar to the time required to construct physical environments, but the storage and maintenance requirements are much smaller.
- The students performed tasks that required easily transferring knowledge from the simulation domain to the physical domain. The RIE shows that while simulations can be used to gain knowledge and experience in a domain, they are only as good as the models they are executing.

Extending the Environment

The simulation is driven from a timer that is created in the constructor for *analoginput*. The timer is set to update the environment 250 times per second. During each update, previous calls to *getdata* are updated with samples from the previous 4ms. The robot is moved according to the motor set speeds and the *drive_robot* function, and the display is updated every 100th time the timer function is called.

To attach a new instance of an existing sensor model, the sensors location is added to the attachment vectors in the RIE constructors. Analog input attachments are made by modifying the in vector in the *robot_analoginput* constructor. Analog output attachments are made by modifying the out vector in the *robot_analogoutput* constructor. Digital attachments are made by modifying the wire_p1 and wire_p2 vectors in the *robot_digitalio* constructor.

To reposition the sensors on the robot, the sensor position vectors must be modified in *robot_config*. The vectors *wall_sensors*, *wall_acc_sensors*, *flashing_sensors,* and *floor_sensors* control the location and direction of the infrared distance triangulation sensors, acoustic distance sensor, optical illumination sensors, and floor line tracking sensor, respectively. The sensor vectors are ordered sets of coordinates in robot frame coordinates. All the sensor coordinates have an x, y pair. The sideways facing sensors also contain a view angle.

To integrate a sensor model into the RIE, a function that interacts with the simulation and returns a time index vector of sensor values must be written. The simulation environment provides the current attitude of the robot, a vector of wall vertices, and a vector of floor line vertices. The current functions are called with a sensor coordinate set and a time vector. There is a minor issue with not having a vector of robot positions that correspond to the time vector.

To integrate the new sensor model function into the environment, a new sensor attach letter must be chosen and used to attach the sensors in the appropriate constructors. Then *robot_update*, *start*, *getsample*, *getdata*, *getvalue* and *putvalue* must be modified to interact with the sensor model. *Start*, *getsample* and *getdata* are part of the *robot_analoginput* class and handle analog input calls. *Getvalue* and *putvalue* are part of the *robot_digitalio* class and handle digital IO calls. *Robot_update* must be modified to provide timely data gathering functionality when running a multipoint analog data gathering event.

To model a different robot drive train, all that is required is a rewrite of the file *drive_robot*. *Drive_robot* contains a function that takes a left wheel and right wheel control voltage. The function calculates where the robot will be one time step in the future using a global constant as the length of one time step.

Future Work

The developed simulation environment is currently being used in a senior elective in autonomous mobile robotics and a junior design course. In the autonomous mobile robotics course the simulation environment is used as a platform for students to develop skills in programming robots to maneuver in real world situations with realistic sensors and control response. In the junior design course the simulation environment is being used to facilitate group projects in a realistic setting providing experience in group dynamics, project partitioning, and handling open ended questions and design.

Use of the dual hardware / simulation environment leads to direct measurable evidence for the following ABET outcomes: [18]

- An ability to design and conduct experiments, as well as to analyze and interpret data.
- An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
- An ability to identify, formulate, and solve engineering problems.
- An ability to use techniques, skills, and modern engineering tools necessary for engineering practice.
- Knowledge of mathematics and the basic sciences, computer science, and engineering sciences necessary to analyze and design complex electrical and electronic systems which may include hardware and software.

The authors are currently working to leverage the developed simulation environment in other courses in engineering at Saint Louis University. Lessons are in place for Electrical and Computer Freshman Engineering that will use the simulation environment to introduce students to topics of signal processing, controls, and robotics while giving them a real world exercise to learn a new programming environment. The department is planning a laboratory course for the Linear Systems class to utilize this environment. The departments of Biomedical Engineering and Mechanical Engineering are looking at modification to the capstone project in their Measurements course to include this environment to allow students to develop programming skills for sensor interrogation and integration.

Conclusion

Students express the thought that lecture material without tangible experimentation is difficult to synthesize. By creating this simulation environment students can interact with robots in both simulation and real hardware in the same manner. This facilitates students gaining knowledge of the underlying system by experimenting in the simulation before being burdened with the difficulties of real hardware. Observation of this environment being used in course activities show that once students see the linkage between the simulation robot and the real robot they lose their misconceptions about lessons utilizing the simulation not being grounded in practice application.

A website containing the current simulation environment along with a test case example and several example projects can be found at:
http://csss.slu.edu/~mitchekk/courses/robot_sim/

References

1. S.F. Barrett, D.J. Pack, P. Beavis, M. Sardar, A. Griffith, L. Sircin, G. Janack, “Using robots to Teach Complex Real Time Embedded System Concepts,” Computers in Education Journal, OCT-DEC, 2006.

2. Griffith, S.F. Barrett, D.J. Pack, “Verilog HDL Controlled Robot for Teaching Complex Systems Design,” Computer in Education Journal, Jan-March, 2008.

3. Christopher Kitts, Neil Quinn, “An interdisciplinary field robotics program for undergraduate computer science and engineering education” Journal on Educational Resources in Computing (JERIC) Volume 4, Issue 2, June 2004

4. Seung Han Kim, Jae Wook Jeon, “Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms” IEEE Transactions on Education, Volume 52, Issue 1, pp. 99 - 108, Feb. 2009.

5. Maja J Mataric, “Robotics Education for All Ages” Proceedings, AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education, Palo Alto, CA, Mar 22-24, 2004

6. Martin Grimheden, Martin Torngren, “What is embedded systems and how should it be taught?---results from a didactic analysis”, ACM Transactions on Embedded Computing Systems (TECS), volume 4, Issue 3, pp. 633 - 651 August, 2005

7. National Instruments, http://www.ni.com/ data acquisition

8. The MathWorks, http://www.mathworks.com

9. M. N. Horenstien, “Design Concepts for Engineers, 2nd Ed.”, Prentice Hall.

10. 2009 IEEE Region 5 Conference, Lubbeck, Texas http://www.ewh.ieee.org/ reg/5/ index.html

11. http://www.budgetrobotics.com/shop/ ?shop=1&cart=777144&cat=58

12. http://www.dimensionengineering.com/ Sabertooth2X10.htm

13. http://sine.ni.com/nips/cds/view/p/lang/en/ nid/14605

14. http://www.phidgets.com/products.php?category=2&product_id=2008

15. http://www.robotshop.us/devantech-ultra-sonic-range-finder-srf05-1.html

16. http://www.taosinc.com/productdetail.aspx?product=115

17. http://www.lynxmotion.com/Product.aspx?productID=58&CategoryID=

18. http://www.abet.org

Biographical Information

Dr. Kyle Mitchell is an Assistant Professor in Electrical and Computer Engineering and a Research Associate in the Center for Sensors and Sensor Systems at Saint Louis University. He received his Ph.D. from the Missouri University of Science and Technology. His research interests include embedded systems design, data acquisition systems design, system modeling, and data feature extraction.

Dr. William J. Ebel, a senior member of the IEEE, is an associate professor of Electrical and Computer Engineering at Saint Louis University and currently serves as the director of the Center for Sensors and Sensor Systems. He received the PhD from the Missouri University of Science and Technology in 1991. Previously he held faculty positions at the Virginia Polytechnic Institute and State University and Mississippi State University. His research interests include robotics, digital signal processing, and digital/cellular communications. He has been involved in over $750,000 of research funding and has over 20 published papers in conferences and journals.

Dr. Roobik Gharabagi is an Associate Professor and the Chair of the Electrical and Computer Engineering Dept. at the Saint Louis University. He received his Ph.D. from the University of Pittsburgh, PA. His research interests include Semiconductor Devices modeling and simulation, Integrated Circuits, and Undergraduate Education.