Computer Applications in Mechanical Engineering

Author: Estelle M. Eke

Professor of Mechanical Engineering

California State University, Sacramento

Introduction and Motivation

Juniors in mechanical engineering at California State University, Sacramento (Sacramento State) are required to take a 3-unit course titled “Computer Applications in Mechanical Engineering (ME 175)”. Prior to the fall 2000 semester, FORTRAN and MATLAB were the primary software packages used. The prerequisites are (i) any high-level programming language including C and C++, (ii) engineering materials, (iii) circuits and (iv) engineering mechanics – statics. The mode of course delivery is two 50-minute lectures and a 3-hour laboratory per week. Emphasis was on the introduction to numerical computation and assigned problems were solved on a PC/Workstation. Tests and final exams that rely heavily on computation were used to evaluate student performance; laboratory reports were used to assess writing skills. It was observed that a typical class was made up of two types of students; those who enjoyed programming, and students who considered programming as drudgery and were not motivated to do more than the minimum amount of work required to get a passing grade. The latter group also had difficulty relating the computer exercises in the textbook to real-world applications. After teaching this course a few times, the author decided to explore methods that might make the course more exciting to a greater number of students while remaining challenging. After some research it was decided that computer control of objects using microprocessors might be a good addition that will allow the students to test their programming skills, complement the techniques encountered in the numerical exercises, and at the same time lead to fun and challenging designs.

Objectives

The objectives for ME175 are to:

- Provide students with a basic exposure to numerical methods.
- Use MATLAB as the software environment to conduct numerical analysis.
- Perform simulations using SIMULINK (a MATLAB toolbox).
- Reinforce principles of computer science, electrical engineering, mechanical engineering through open-ended robot design with the Basic Stamp (a micrcontroller).
- Engage students in problem solving via team work.
- Provide a brief introduction to the design process.
- Give students an opportunity to demonstrate oral and written communication skills through oral presentations and final project demonstrations.

- Serve as a useful prerequisite for courses such as controls, mechatronics, modeling of dynamic systems, vibrations, and capstone design.

Course Structure

Beginning in fall 2000 the 16-week semester course was restructured such that 8 weeks are devoted to the theory of numerical analysis and problem solving in the MATLAB environment. The numerical techniques covered in this course spanned topics encountered in a typical numerical methods textbook[1-3]. The topics covered are: introduction to linear algebra, the solution of systems of linear equations, curve-fitting, interpolation, and the solution of ordinary differential equations. In the next 2 weeks a brief introduction to controls and/or vibrations is given. The accompanying laboratory exercises involve simulations via SIMULINK, and provide some insight to model-based design for dynamic systems. In the last 5 weeks programming in the Parallax PBasic language, an interpreter for the Basic Stamp microcontroller[4-5] is introduced. An open-ended robot design project is also assigned. The students present their projects in week 16.

The course syllabus shown in table 1 provides more details regarding course structure. Four 50-minute tests are administered on the MATLAB and SIMULINK portions; two of these tests cover the theory and the remaining two test the students' programming skills. An oral presentation is required by each group in the preliminary phase of the robot design. The final examination consists of a powerpoint presentation, a demonstration of each group's project, and a technical report. Every student in a group must write a portion of the report so that his/her writing skills may be assessed. Students evaluate their peers' presentations and demonstrations. Grade distribution (MATLAB and SIMULINK 60%; project 40%).

Table 1. Course Syllabus

Lecture 1: Introduction to Computing Environment (SacCT, UNIX, Voyager, Windows);
Review of Linear Algebra http://www.purplemath.com/modules/index.htm
Lab 1: Introduction to software (MATLAB) Driver, Plots, Conditional Statements; User-defined functions;Exercises with vectors and matrices
Lecture 2: Global variables; Data files: Read and Write
Lecture 3: Graphical User Interface (GUI) (Instructor notes)
Lab 2: Creating a GUI - Exercises with vectors and matrices
Lecture 4: Introduction to the PBasic Platform
Lecture 5: Review: Generating Plots in a GUI; Reading Data files
Lab 3: Completion of GUI Exercises with vectors and matrices; Introduction to the Basic Stamp Microcontroller
Lecture 6: Programming in PBasic; Subroutines (i.e. User-defined functions)
Lecture 7: Unavoidable Errors in Computing; Solving Systems of Equations
Lab 4: Microcontroller Basics with the Basic Stamp

Table 1. Course Syllabus continued

Lecture 8: Application of matrices – equations of motion for a robot arm Importing/Exporting MATLAB data to/from Excel
Lab 5: Microcontroller Basics with the Basic Stamp
Lecture 9: Least-squares Fitting of Curve to Data
Lab 6: Curve Fitting
Lecture 10: Additional examples in curve-fitting; Extracting equations using best-fit
Lecture 11: Interpolation
Lab 7: Fit curve and obtain equation for best-fit
Lecture 12: Numerical Integration of Ordinary Differential Equations ; Runge-Kutta Methods
Lecture 13: Analyzing non-stiff systems ODE45
Lab 8: Solving Initial Value Problems using ODE45
Lecture 14: Introduction to SIMULINK for solving Ordinary Differential Equations
Lecture 15: Application to Vibrations: Mass-Spring-Damper System using SIMULINK
Lab 9: Solving IVPs using SIMULINK
Lecture 16: Analyzing other systems using SIMULINK
Lab 10: Programming a microprocessor; constructing digital circuits
The Final Project (Lectures and self-paced labs) Technical Writing ▪ http://www.writing.eng.vt.edu/ ▪ http://www.calstatela.edu/library/guides/3mla.pdf ▪ Oral Presentation 1, Gantt Chart Week 16: Final Presentation (Oral presentation 2 & Demo)

A Review of Structured Programming via GUI Creation

The title *Computer Applications in Mechanical Engineering* encompasses a wide area and gives the instructor flexibility to choose from a variety of mechanical engineering applications. Since students at the junior level who take this course have already received exposure to various high-level programming languages such as C,C++, FORTRAN, JAVA and MATLAB, the first few lectures constitute a review of and/or introduction to MATLAB programming. Emphasis is placed on user-defined functions and the creation of a Graphical User Interface (GUI). Also Matlab requires that users have a good understanding of matrix operations. Thus a successful creation of a GUI such as that shown in Figure 1, demonstrates that the student (a) understands the importance of creating user-defined functions or modules that can be easily

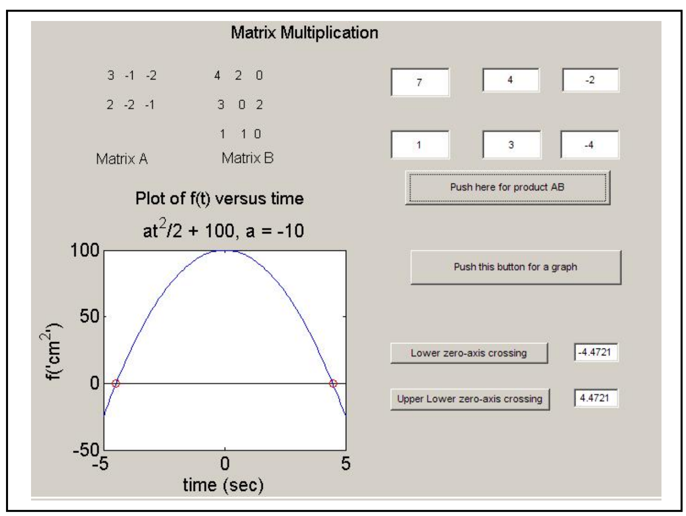

Figure 1. A Graphical User Interface generated in MATLAB

linked with other functions, and (b) can perform fundamental linear algebraic operations. Students are encouraged to use GUIs when presenting solutions to other laboratory exercises. About fifty percent of the students choose to use GUIs.

Basic Stamp Projects

The open-ended projects assigned in the last few weeks will now be discussed. The instructor presents the guidelines for the projects and the entire class provides inputs in the preliminary phase. In some semesters multiple microcontroller-based projects are provided by the instructor, and depending on the complexity of a project, teams may consist of two, three or four students. At the end of each semester every student is required to give feedback on the entire process. This feedback is used to improve the guidelines for subsequent semesters. Microcontroller-based projects require the use of the Parallax Basic Stamp microcontroller.

Background

Robots are used in many engineering design situations. In particular microprocessor control is basic to understanding how non-standard features such as servo control, programmable action, position sensing, response and PC-interfacing work. Microprocessor control is one of the foundation elements in mechatronics, a methodology used for the optimal design of electromechanical products. Mechatronics is multi-disciplinary, and allows today's engineering students to gain and use knowledge across the board in electrical, mechanical, and computer sciences, and in information technology.

In ME175, the students have been introduced to the Board of Education Basic Stamp 2 microcontroller and the Board of Education robot, Boe-Bot. A 5-week semester project allows students to demonstrate their programming skills by using computer control to maneuver two robots while performing a repertoire of actions. Videos of student projects completed since fall 2000 can be found at the author's web site[7]. Four examples of student projects are now presented. All projects may exhibit some of the additional features shown in Table 2 with a limitation that only materials supplied by or agreed upon by the instructor and class members are permitted.

Table 2. Other desirable features for each project

▪ Communication with PC or Cell phone ▪ Distance Detection ▪ Drop-off or Edge detection ▪ Infra-red detection	▪ Line following ▪ Light Sensitive Navigation ▪ Navigation with Infrared ▪ Obstacle detection ▪ Sound and/or light

Project#1 Fall 2008: Each group of students must design autonomous vehicles to scatter and collect plastic balls. A sample board for the competition is shown in figure 2; color may be white or brown.

Two Boe-Bots are required and time limit to complete all tasks is 3 minutes.

BoeBot1:

1. Starts at least 60 cm away from the triangular ball depository area carrying 6 plastic balls. Ball diameter is 3 cm.
2. Carefully deposits all 6 plastic balls in the triangular area whose boundaries are marked by tape. Balls must be contained within the triangular area situated on the elevated board (elevation is 15 cm).
3. Carries a manipulator to scatter the balls.
4. Scatters the 6 plastic balls so that each ball crosses a line located 30 cm from the edge of the board. Any ball that enters a hole or rolls off the board will be randomly placed in front of the line as shown in the figure.

Figure 2. 120 cm X 90 cm Competition Board and Specifications

BoeBot2:

1. Starts in a designated location (see figure).
2. Finds a ball without touching any other ball.
3. Deposits the ball in any hole.
4. Repeats steps 2 and 3 at least once.
5. Returns to start location.
6. Does not fall off the elevated board.

Project by Group 11[8] – Fall 2008

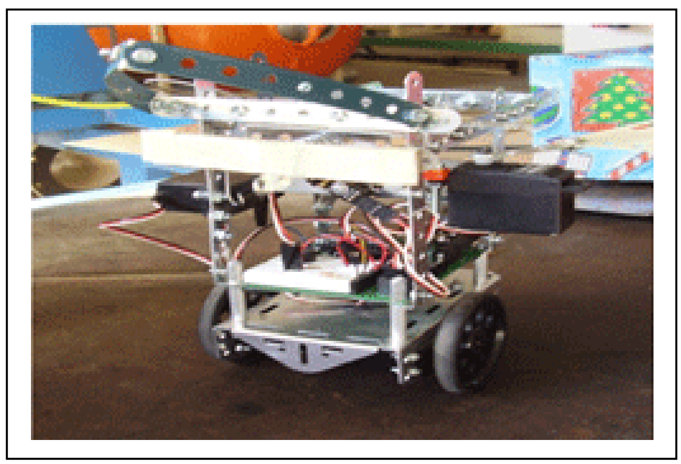

Figure 3. Boe-Bot delivers balls to the table and performs ball scatter

The design and programming of Boe-Bot #2 was the most efficient of all designs presented. The robot accurately accomplished ball retrieval and deposit of ball repeatedly. The sensors and devices below the image in Figure 4 are used for ball detection and ball capture and the sensors

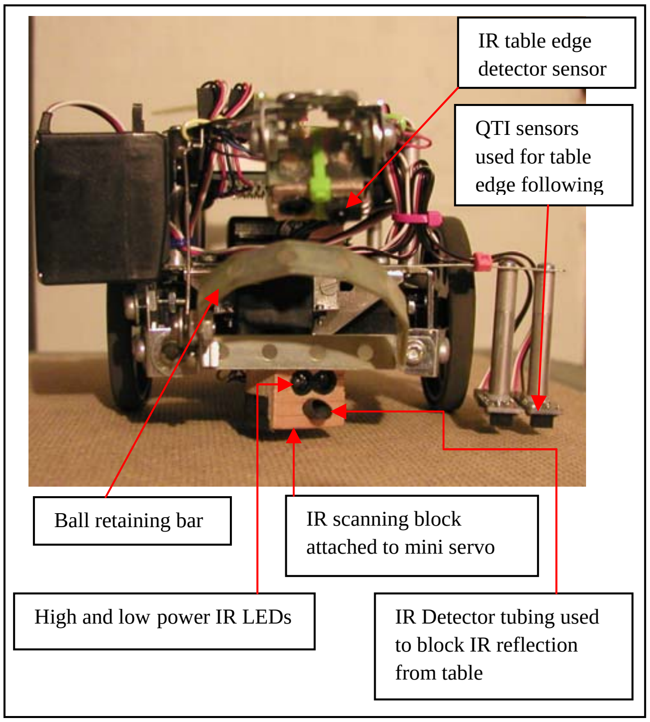

Figure 4. Boe-Bot #2 that detects and retrieves balls and drops the balls in designated hole.

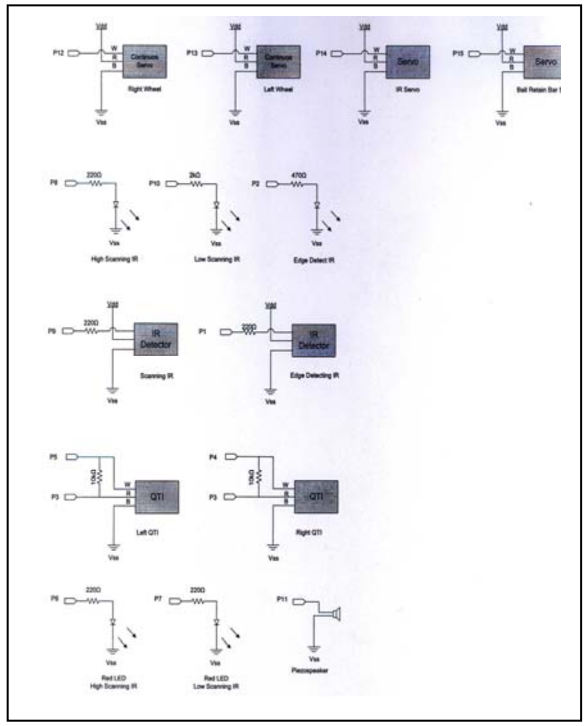

Figure 5. Schematic Diagram for Boe-Bot #2

on the right-hand side are used to navigate to the hole after ball capture. The schematic diagram for Boe- Bot #2 is shown in Figure 5.

Project#2 Spring 2008: Test-tube Retriever

This project was inspired by a robot workcell that consists of two robots Puma 560 and IBM 7575, and a conveyer system found at Professor Harry Cheng's[6] Integration Engineering Laboratory at the University of California at Davis. Some modifications were made as shown in the project guidelines. A team of four students worked on this project.

Goals: Boe-BotA should remove a test-tube full of beads off a rotating platform and pour the beads into another test-tube held by Boe-BotB. Boe-BotB then returns the test-tube back to the rotating platform. Repeat the process.

Guidelines:

a) Maximum project area: 48 inches x 48 inches
b) Elevation of rotating platform: between 4 and 6 inches

c) 3 test-tubes of beads equally spaced on rotating platform
d) Maximum time to complete process: 3 minutes
e) Parallax QTI sensors are not allowed

Project by Group 2[9] – Spring 2008

The group was able to overcome several challenges in order to successfully accomplish the project. The biggest challenge was navigation since QTI sensors were not allowed. A custom printed circuit board (figures 6 & 7) with photoresistors, light-emitting diodes and resistors was

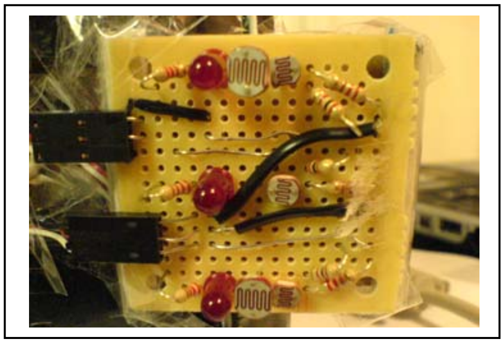

Figure 6: Custom line-follower Sensor

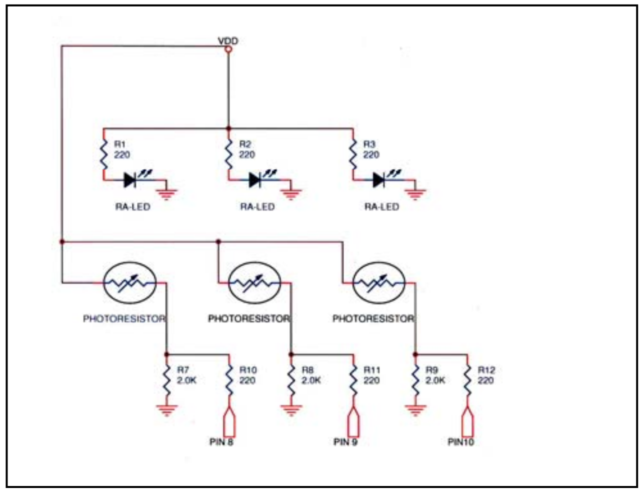

Figure 7. Line-follower Sensor Schematic

developed. The fabrication of the arm/clamp fixture for Boe-BotA (Figure 8), required utilizing one servo to satisfy both the vertical and rolling motions.

The process was successfully repeated and only one bead fell out during the transfer process (Figure 9). A video of this project can be found at the author's web site [7].

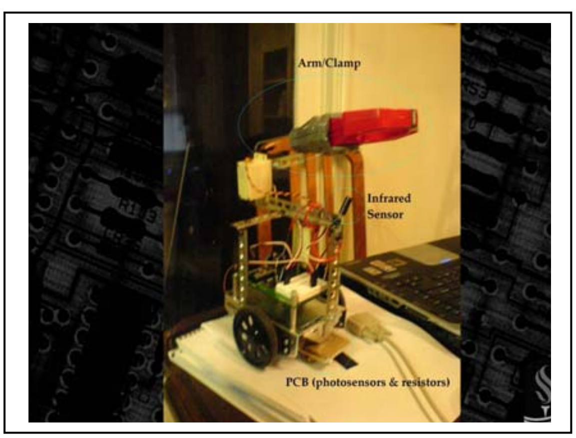

Figure 8. Robot #1 designed to remove test-tube of beads from a rotating platform.

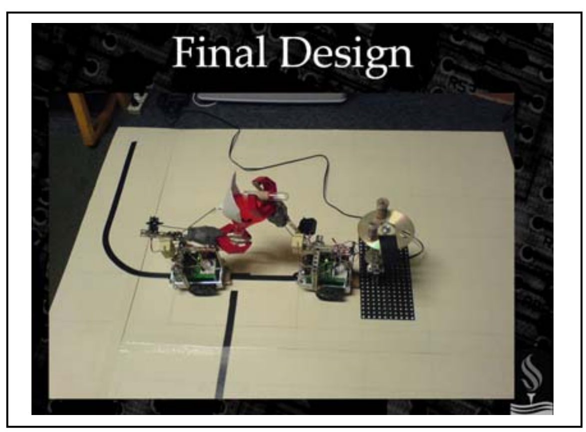

Figure 9. Robot #1 situated next to the rotating platform, transfers the beads from test-tube 1 through a funnel into test-tube 2 held by Robot #2.

In Fall 2007 two projects (#3 and #4) were assigned. Each group of students must select either the *basket ball shooting machine* or the *fire-fighting team*. Although the overall project is open-ended the main goals decided upon by the class are stated below.

Project#3 by Group 7[10]: Basket-ball Shooting Machine

Goals: A container with at least one ball is transported by Boe-BotA. Boe-BotA traverses the edge of the court and parallel parks between two obstacles. Boe-BotB acquires the ball from Boe-BotA and shoots the ball into the hoop. At least one ball should enter the hoop in a maximum of three attempts.

Guidelines:

1. Minimum basket-ball court dimensions: 48 inches x 24 inches
2. Elevation of basket-ball hoop: $\geq$ 10 inches
3. Maximum diameter of basket-ball hoop: 4 inches
4. Minimum diameter of ball (ping-pong) 1.5 inches
5. Minimum distance from hoop at which ball is released: 12 inches.

This project was designed to test the robot's ability to precisely and repeatedly launch an object.

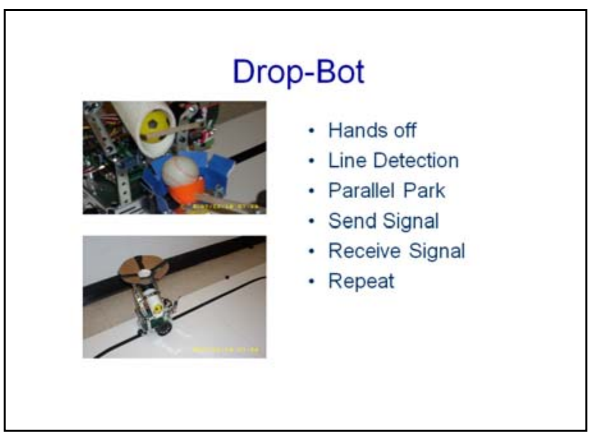

Figure 10. Delivery of ball from Boe-BotA to Boe-BotB and the basketball hoop elevated 12 inches above ground

Figure 11. A close-up view of the shooting mechanism attached to Boe-BotB

The navigation was accomplished using QTI sensors for line following. Boe-BotA (Figure 10) delivers the ball to Boe-BotB (Figure 11) and proceeds to the basket to catch the ball. This robot is capable of releasing one ball at a time. Boe-BotB receives a signal from Boe-BotA that triggers when it should move up to the shooting line that is located 3 ft from the 12-inch high basket. After launching the ball, Boe-BotB then signals to Boe-BotA that the ball has been launched before returning to the start position to receive another ball. The transfer of the ball from Boe-BotA to Boe-BotB was accurately and smoothly done. This team of three students programmed Boe-BotB to shoot the ball into the basket repeatedly resulting in a continuous cycle.

The trajectory generated by the projectile could be analyzed to obtain the time taken to travel from point of launch to destination. Future projects will include analyzing different speeds with which the ball can be launched and the longest range that can be achieved.

Project by Group 5 [11]: Fire-fighting Team

Goals: Boe-BotA detects fire in an area and Boe-BotB has to put out the fire. The area can be a city or a large office or dwelling place. Flame from a miniature candle represents the fire. The fire must be completely extinguished in the shortest possible time.

Guidelines:

1. Area dimensions: width > 4ft, 4ft $\leq$ length $\leq$ 10 ft
2. Fire may be from a miniature candle, a flame imitator, or an infra-red sensor
3. Area options (a simulated city, a cluster of buildings, a grid of houses along streets)

This project generated a lot of interest. Group #5 was able to develop a unique idea that involved using transmitters to send a signal from Boe-BotA to Boe-BotB. The computer code was quite advanced as the group members had to research how to use transmitters and receivers, a topic that was not covered in the course. The grid shown had to be accurately mapped to ensure that Boe-BotA knows how many intersections it encounters along the grid as it searches for the fire. It can sense the fire on both sides of the grid. Boe-BotB was programmed to take the shortest path to the fire. Boe-BotB is equipped with a small electric fan powered by a 9-volt battery that rotates until the fire is in its line of sight and then extinguishes the fire.

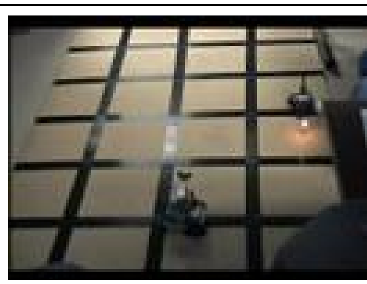

Figure 12. The fire-fighter team. Boe-BotB moves to put out fire after receiving signal from Boe-Bot A

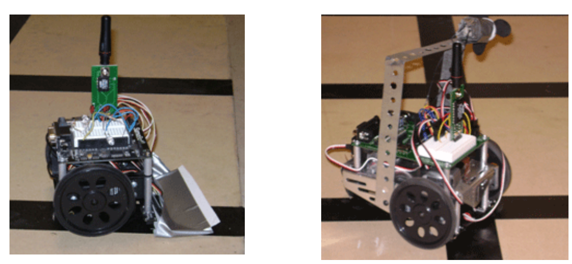

Figure 13. Boe-BotA and Boe-BotB

Some of the programming challenges included: (i) interference from the receiver which gave false directions, and (ii) complexity of code that made the Basic Stamp run out of Electrically Erasable Programmable Read Only Memory (EEPROM). The number of variables had to be reduced and the smallest size possible had to be used for the variables in order to make optimal use of available memory. Also subroutines had to be efficiently written.

This group's design was the most efficient of the four groups that attempted this project. The time taken to detect and extinguish the fire was less than two-minutes on a 4-ft by 6 ft area consisting of 16 grids.

Student Feedback

In this section some of the feedback provided by students at the end of the course is presented. Specific questions are not provided but rather students are asked to give comments and improvements that they think will be useful to incoming students and the instructor.

- Feedback 1: Start to work on the project immediately
- Feedback 2: Create a Gantt Chart and assign members to specific task at the start of the project; allow members to choose areas of expertise (the main areas are programming, design, build, powerpoint presentation creation)
- Feedback3: Keep the design simple and have a backup design
- Feedback4: Group members should be ready to collaborate and assist each other especially if one member is running behind schedule
- Feedback 5: Select materials that are inexpensive and will do the task
- Feedback 6: Use resources on the Internet and avoid re-inventing the wheel.

Student recommendations on areas that can be improved by the Instructor

- Feedback 7: Assign the project earlier than the last 5 weeks as instructors in other courses may also assign projects; the robot project is sometimes time-consuming
- Feedback 8: Assign projects that require one robot for a 2-member group
- Feedback 9: Spend an extra lecture on the software features such as pulse-width-modulation (PWM) and Electrically Erasable Programmable Read Only Memory (EEPROM)
- Feedback 10: Give the same project to the entire class, it makes evaluating the projects easier.

Implementation to Date – A Response to Student Feedback

The response to student feedback 7 – 10 follows:

- Response7: In the current semester, spring 2009, PBasic is introduced in week 3 (see Table 1 Course Syllabus). The project is assigned in week 12. It is preferable for the students to concentrate on the project in the last few weeks.
- Response8: After some discussions the conclusion is that the primary concern is the cost of the Boe-Bots. A Boe-Bot is available for loan to a group of two members. In extreme

circumstances the class may allow the use of one-robot but then the project has to be modified.

- Response9: As shown in Response7 above PWM and EEPROM are now being addressed and extra help will be provided as needed.
- Response10: This change was made last semester (Fall 2008) and it worked quite well. There was a lot of collaboration among groups even though each group's project maintained the unique characteristics presented in the preliminary phase.

Conclusions

The inclusion of projects dramatically increases students' interest in the subject. Even at the beginning of the course students express their anticipation in the hands-on robot designs that the course offers. Faculty from the college of engineering, students from other disciplines, friends and families frequently attend the end-of-semester presentations. The graphical user interface is used in other courses and students appreciate how they were developed. Some students have applied these GUIs in courses such as statistics. The overall passing rate has greatly improved. It has been observed that the focus on numerical methods as a means of providing a foundation to real-world problem solving definitely complements the project approach. Students now understand the notion of acceptability of solutions, and are aware of errors encountered in computing and how it relates to real-world designs. The team approach reveals to each member that the learning experience consists of frustration, compromise, and ultimately success. Future development already approved by the department of mechanical engineering includes offering a similar structure in an introduction to computer programming course so that students may appreciate at the onset why understanding programming concepts is essential for engineers. Emphasis will also be placed on communication between MATLAB and the Basic Stamp2. This approach establishes a most important link between theory and implementation.

Acknowledgments

Project examples presented in this paper were contributed by the following students:
Aman Bains, Rod Baybayan, Alan Camyre, Ryan Dunish, Ian Hellstrom, Robert Jolley,
Drew Mast, Cheng Moua, Matt Morrison, Garret Snedeker, Dustin Sutherland, Justin Pettenger, and Hung Tran.

References

1. Fausett, L. (2007). *Applied Numerical Analysis Using MATLAB.* Upper Saddle River, New Jersey: Prentice Hall.

2. Gerald, C., & Wheatley, P. (2003). *Applied Numerical Analysis*. New York: Addison Wesley Longman, Inc.
3. Recktenwald, G. (2000). *Numerical Methods with MATLAB: Implementation and Application.* Upper Saddle River, New Jersey: Prentice Hall.
4. *What's a Microcontroller: Student Guide.* Version 2.2. Retrieved February 16, 2008 from Parallax.com Web site http://www.parallax.com/Portals/0/Downloads/docs /books/edu/wamv2_2.pdf
5. *Robotics with the Boe-Bot: Student Guide.* Version 2.2. Retrieved February 16, 2008 from Parallax.com Web site http://www.parallax.com/Portals/0/Downloads/docs /books/edu/RoboticsV2_2.pdf
6. Cheng, H. (2007). *Faculty Page*. Retrieved June 18, 2007 from University of California Davis Web site http://iel.ucdavis.edu/people/cheng.html
7. Eke, E. (2009). *Faculty Page.* Retrieved February 16, 2009 from California State University at Sacramento Web site http://gaia.ecs.csus.edu/~ekee/
8. Jolley, R., Camyre,A., & Morrison, M. (2008). Undergraduate Students, Department of Mechanical Engineering, California State University at Sacramento: Sacramento, California.
9. Baybayan, R., Mast, D., Moua, C., & Tran, H. (2008). Undergraduate Students, Department of Mechanical Engineering, California State University at Sacramento: Sacramento, California.
10. Snedeker, G., Pettenger, J, & Hellstrom I. (2007). Undergraduate Students, Department of Mechanical Engineering, California State University at Sacramento: Sacramento, California.
11. Sutherland, D., Dunish, R., & Bains, A. (2007). Undergraduate Students, Department of Mechanical Engineering, California State University at Sacramento: Sacramento, California.

Biographical Sketch

Estelle Eke received a B.S. degree in aeronautical and astronautical engineering from Purdue University, an M.S. in mechanical engineering and materials science from Rice University, and a Ph.D. in aeronautical and astronautical engineering from Rice University. She worked for two and half years in the Spacecraft Navigation Section at the Jet Propulsion Laboratory in Pasadena, and then taught for two and half years in the Department of Aerospace Science Engineering at Tuskegee University before joining Sacramento State University. While at Tuskegee University, she received the Teacher of the Year award in Aerospace Engineering for two consecutive years. At Sacramento State, she was named Outstanding Teacher in the College of Engineering and Computer Science in 2000. She is currently Professor of Mechanical Engineering and teaches courses in the general areas of dynamics and control. Her research interests are in optimization and robotics. She also serves as a design judge for First Robotics competitions at the elementary and high school levels.