DEVELOPMENT OF A SENIOR LEVEL ROBOTICS COURSE FOR ENGINEERING STUDENTS

C. N. Thai[1], J. M. Mativo[2]
[1]Biological and Agricultural Engineering Department
[2]Workforce Education, Leadership and Social Foundations Department
University of Georgia

Abstract

For Spring 2011, a senior-level robotics course (first taught in Spring 2010) had been revised according to principles for "Smart Teaching" described in the book "How Learning Works". Homework, laboratory sessions and anchor projects had been redesigned to provide better scaffolding for students with 2 different but complementary engineering backgrounds, and also for a better flow towards the theme of humanoid robotics. The e-portfolio tool EMMA was integrated into this course as a collaboration and feedback tool between instructor and students to help improve student algorithm development work, but EMMA was not found to be responsive enough nor useful for this kind of use. Student responses had been very positive to the project-based approach of this new course, but they had found challenging the subject of Zigbee communications programming between robots and were not confident in C++ programming techniques as they should.

Introduction

In the Summer 2010, the Computers in Education Journal published a selected survey of novel approaches to robotics education for high school and engineering undergraduate levels which indicated that currently in the U.S.A. Worcester Polytechnic Institute is probably the only university that currently offers a stand-alone B.S. degree in Robotics Engineering [1], while other universities such as Rose-Hulman Institute of Technology adopted the approach of a multidisciplinary robotics minor for students majoring in Computer Science, Electrical, Computer, Mechanical or Software Engineering [2]. With the recent approval by the University System of Georgia Board of Regents for new B.S. degrees in Mechanical Engineering and Electrical and Electronics Engineering to come on line in Fall 2013 at the University of Georgia, and when combined with the existing Computer Systems Engineering B.S. degree, an emphasis area in Robotics is looking very viable to be developed for undergraduate students enrolled in the above three degrees at UGA. As an initial step, we are looking at using robotics as an instructional approach to integrate hardware, software and communication technologies at a senior-year level course [3,4] whereas students would already have taken courses on Microcontrollers, Sensors and Transducers, Kinematics, Dynamics, Machine Design and Control Systems. The goal is to provide students with a basic practicum in Embedded Robotics wherein the students will learn about the programming of embedded controllers, the actuation of servo motors, the interfacing of sensors, inter-computer serial communications, and the control of autonomous as well as remotely piloted systems. At this point in time, as the potential students taking this course would be from Computer Systems Engineering (CSE) and Agricultural Engineering (AE), we chose an experimental analysis approach whereas students would work on existing robots instead of a creative design approach such as the "Robot Diaries" approach developed at Carnegie Mellon University [5] and also their NSF-sponsored program FIRE (http://fire.cmu.edu/) In the near future when more ME and EE senior students are available we are expecting to shift towards a more "creative mastery" approach. After extensive research into commercial robotics systems from the cost point of view as well as from the ability to expand hardware and software sophistications into future graduate robotics courses, we opted to go with the Bioloid systems from Robotis (http://www.robotis.com). The most attractive feature of the Bioloid systems is their potential for link-based locomotion, allowing us to go beyond wheel-based systems [6].

This project-based course in Robotics was implemented for the first time in Spring 2010 with 3 students and was designed around 3 projects with lectures and laboratory demonstrations performed by the instructor to provide necessary background materials for students to carry on successfully with their chosen projects. Students were asked to keep design notebooks (paper-based) that were reviewed and assessed weekly. Mid-term and final rubric-based course assessments were also performed. The 1st project goal was for students to practice combining Remote Control and Autonomous Behavior programming in one application resolving a "tail-gating" situation between 2 car bots whereas the rear car bot had to ignore user commands and performed autonomous maneuvers to help it avoid from colliding into the front car bot which had stopped suddenly. The 2nd project challenged the students to create appropriate gait solutions for a simple bi-pedal (7 servos) robot (called GERWALK) to negotiate going up and down stairs steps. For the 3rd project, 3 humanoid robots (18 servos each) were built with different capabilities and tasks to be performed: a) Humanoid A was equipped with a gripper and 2 NIR distance sensors (one on the tip of its left arm and one forming its head), its task was to use its left arm sensor to locate a dowel bundle, turned an appropriate amount to face the bundle, approached it within a proper distance, then grabbed and lifted up the bundle; b) Humanoid B's task was to use its wireless video camera to locate a blue dowel (its beacon) and walked to it, however it had to avoid the red dowels that were placed at random blocking its path towards the blue dowel; c) Humanoid C was to use its 3-axes balance sensor to help it maintain balance as it walked up a ramp, however the inclination angle of the ramp could be varied at will by the user. For Spring 2010, only the Humanoid A & B projects were performed successfully. During Summer 2010, new projects were designed and tested successfully: a) Using a PC base station running on LabView, 3 Car bots were sent out to travel autonomously as far as they can from the PC while maintaining wireless communications so that the Car bots could report to the PC readings from their NIR sensors; b) to combine 2 simple biped robots in one 4-legged robot so as to illustrate coordination issues on a robot that had 2 independent microcontrollers capable of wireless communications, and also to demonstrate the enhanced stability and maneuverability obtained; c) to compare stability and range of motions achievable between 2 humanoids balancing on 1 leg, one using a 3-axes IMU-based sensor, while the other used 4 pressure sensors mounted beneath its balance foot (videos at http://www.engr.uga.edu/~mvteachr/RobotVids/).

The objective of our current paper is to describe the revised curriculum of this course to be used in Spring 2011 in terms of new student project goals and our experiences in implementing an e-portfolio tool called EMMA (http://emma.uga.edu/) as a collaboration and feedback tool between instructor and students. EMMA was originally designed for English Composition but through its many iterations it is now a very versatile e-portfolio tool with an XML editor with multimedia storage facilities making it suitable for engineering design notebooks, except for the lack of scientific symbols perhaps.

Bioloid Robotics Systems Description

Currently, there are 3 Bioloid robotics systems commercially available - all based on the Atmel AT mega microcontroller (see Fig. 1):

1. The Comprehensive kit uses the CM-5 controller which can interface with actuator modules AX-12+ and sensors modules AX-S1 (NIR and sound) connected in a daisy chain fashion using TTL serial protocols rated at 1 Mbps (called Dynamixel

bus). It can also communicate via RS-232 and ZigBee protocols at a recommended rate of 57,600 bps to standard PCs and other Bioloid controllers.

2. The Premium kit uses the CM-510 controller with additional capabilities beyond the CM-5's as it can also interface with user-created sensors using memory-mapped I/O ports. It also has similar RS-232 and ZigBee communications capabilities.

3. The CM-700 became available in mid 2010 as a bare-bone controller having all the capabilities of the CM-5 and CM-510, and also RS-485 interfaces to the more advanced servo motors from Robotis series RX and EX. The CM-700 is not yet used in our robotics course.

Figure 1. Current Bioloid controllers CM-5, CM-510 and CM-700.

For communications between a PC and the various CM-5/510/700 controllers, there are a variety of options (see Fig. 2):

1. Plain RS-232 9-pin cable between PC COM ports to a mini-jack port on the CM-5 or CM-510.

2. For newer PCs with only USB ports, ones can use the USB2Dynamixel module to connect the above 9-pin cable to an available USB port. The USB2Dynamixel module also allows ZigBee communications between the PC and various CM controllers when used with a Zig2Serial module and a Zig-100 daughter board.

3. The CM-700 can only use the USB-based LN-101 module for either program development tasks from the PC or during run-time uses the Zig-110 for ZigBee communication between itself and the PC or with other CM controllers, but not both at the same time as only one physical port is available for external communications.

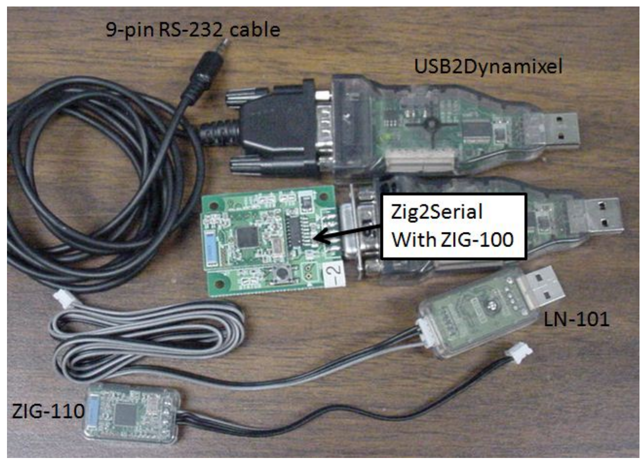

Figure 2. Communication options between PC and various CM-5/510/700 controllers.

Regarding software programming tools on the PC side, the student can start from a beginner IDE called RoboPlus Task and later go to direct API programming using Linux (gcc) or MS Windows (Visual Basic, C++, C#, LabView and MATLAB). Recently Robotis had also provided embedded C programming libraries for its Dynamixel modules and ZigBee communications at its support web site (http://support.robotis.com/en/). The CM-5/510/700 controllers are based on the Atmel AT mega microcontroller at 16 MHz, with either 128 KB (CM-5) or 256 KB (CM-510/700) of Flash memory. The useraccessible memory area is divided into 3 independent but cooperating sections:

1. The main user logic resides in the TASK section which has standard features such as "main" and other user-defined functions. Familiar selection and repetition structures are available, but no parameter arrays can be defined by the user currently. A special function named CALLBACK can also be defined here but only once.

2. The CALLBACK section is executed every 7.8 ms which is also the refresh time period for all servo motors. Limited commands are allowed in the CALLBACK section to prevent collisions with the other commands from the TASK section.

3. The MOTION section contains the definitions of various static "poses" that the robot can take. Each pose is essentially a data structure representing coordinated absolute positions of the relevant servo motors used to build a given robot. These static poses can be further modified by applying JOINT OFFSETs which can be computed during run-time from user-defined algorithms responding to changes in selected servos.

However these RoboPlus tools will not be available when Embedded C applications are used as the original Bioloid firmware is effectively overwritten by the Embedded C application.

Evaluation of Spring 2010 Course Implementation

In Fall 2010, we performed an evaluation of the Spring 2010 implementation of this robotics course using the principles for "Smart Teaching" from a recent book titled "How Learning Works" [7] and from a second book titled "Integrating Differentiated Instruction and Understanding by Design"[8]. Some of the main lessons learned and possible solutions to be implemented for Spring 2011 are listed below:

1. The paper-based design notebooks created an inefficiency bottleneck during the transfer from students to instructor and vice-versa. Students were also not keeping up with their design notes regularly. We hope to resolve these issues by using the EMMA tool for its ubiquitous web access which should improve the frequency of submissions and reviews.

2. Student problem-solving skills such as algorithm development and applications of mechanics knowledge could not be captured previously and thus timely and targeted feedback could not be done in an organized manner. Two changes are planned for Spring 2011:

    a. The formal class times will be changed from MWF (55 minutes each time) to a MW schedule with 2 back-to-back class periods on Monday and a single class period on Wednesday to allow more continuous discussions and hands-on opportunities on Monday.

    b. With the use of EMMA, when students work on their assignments outside of formal class times, for example they would be able to just use their cell phones to record video clips of the problems encountered and post them via EMMA. The instructor in turn can review these videos and offer possible solutions asynchronously.

3. More scaffolding needs to be done in our lecture materials (such as course concept maps and realignment of the projects towards a certain theme), and also in the project requirements (i.e. require students to show planning of their projects and to

specify how they used previous knowledge (from previous courses and/or previous projects) and instructor feedback in subsequent work. More details are given in later sections describing the revised or new course projects.

Overall Course Objectives and Project Descriptions

As this is a first course in Robotics, the integration between controller programming, actuators control, sensors interfacing and communications design was of first importance in our selection of instructional materials as well as in our design of the scaffolding projects. We also have interests in the area of multi-controller communication and control applications, as wireless robot to robot communications are becoming important issues to consider in robotics [9,10], and lastly in humanoid robotics which will be developed further into a future split-level course within a few years.

At present, the students taking this course will either come from the CSE curriculum or from the Electrical and Electronics emphasis area of the AE curriculum. The AE students only formal exposure to software programming was a 2-credit freshman course taught using Excel and MATLAB, while the CSE students software training was much more extensive as one third of the courses in the CSE curriculum were in the Computer Science department, thus we will be expecting difficulties from AE students in handling Embedded C applications towards the end of the semester. On the other hand, the AE students had more exposure to core engineering courses in Kinematics, Dynamics, Mechanical Design and also in Microcontroller Programming at the assembler level which should give them further insight into link-based locomotion robots. Regarding the Control Systems area, both types of students should have the same training opportunities, thus overall the synergy implication between the two groups is clear. As a consequence of the above issues, we decided to use the higher-level RoboPlus IDE environment in delivering most the instructional materials and to show that C programming (whether embedded or PC side) would allow students to drill down into the structural components of the same concepts already explained at the RoboPlus level, and that they could exploit that capability to go beyond what the RoboPlus environment can offer.

In order to achieve a common knowledge level for both types of students, foundational materials were presented in Chapter 1 of the course for the following concepts developed around a car bot system leading into the Project 1 (these materials mostly unchanged from Spring 2010):

- Description of main functional blocks for typical robotics systems (sense-think-act paradigm).
- Details of Bioloid systems. Hardware capabilities. RS-232 communication concepts. Software development tools (RoboPlus suite, V. 1.0.20.0).
- Hands-on practice using the "MANAGER" tool - direct hardware observation tool and the "TASK" tool – main IDE tool:
  - Programming Servo Motors for the "Continuous Rotation" mode. Interfacing NIR and Sound Sensors integrated into car bot.
  - Autonomous obstacle avoidance programming for car bot (2 approaches - Reactive Control and Behavior Control). Extension to maze navigation.
  - Wireless user control via Zigbee remote controller (RC-100).
- **Homework 1:** Student to add new "speed level" commands using the "numbers" keys to existing TASK code that was made to accept Up-Down-Left-Right commands from user (via the RC-100). Approach and intermediate results to be documented in EMMA.
- **Project 1:** <u>Automated Car Bots Collision Avoidance</u>. Starting from the homework 1 code, this project goal was for students to

practice combining Remote Control and Autonomous Behavior programming in one application resolving a "tail-gating" situation between 2 car bots (both going forward under remote control by separate students using RC-100s). When the front car bot suddenly stopped, the rear car bot using its NIR sensors would trigger an autonomous response (i.e. ignoring further user commands from the RC-100) to help it avoid from colliding into the front car bot. After the rear car bot performed successfully a passing maneuver, user remote control commands would once again be accepted and acted upon by this car bot (see Fig. 3). Approach and intermediate results to be documented in EMMA. Videos of student demonstrations of Project 1 will be recorded and published on a public website to showcase student mastery of the subject (http://www.engr.uga.edu/~mvteachr/RobotVids/).

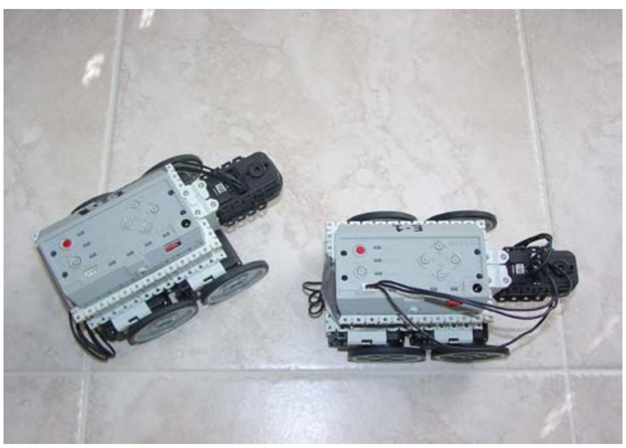

Figure 3. First Project: Rear Car Bot Avoiding a Tail-Gating Situation.

Chapter 2 had been designed to introduce students to bi-pedal locomotion concepts leading to Project 2 whereas a GERWALK robot had to negotiate stairs steps (see Fig. 4). During the Spring 2010 course implementation and other implementations of the same project with freshman engineering students, we noticed that some student gait solutions made the front leg rotate inwards while the rear leg was moving from the lower stair step to the upper one, thus resulting in a misalignment of the feet with respect with the upper step afterwards, and occasionally would make the robot fall off the stairs (please refer to Fig. 4). When these specific students were asked if they could come up with an explanation of this phenomenon, nobody could. Thus we decided to modify Chapter 2's instructional materials so that students in Spring 2011 would have the opportunity to explore experimentally the conditions that would result in this auto-rotation of the front leg (the actual explanation involves the concepts of Center of Pressure and Zero Moment Position which are quite beyond the scope of this course) [11].

Figure 4. GERWALK robot going up stairs.

Thus in the "new" Chapter 2, programming concepts for the "Position Control" mode for servo motors and for advanced sensor interfacing (3-D IMU and Foot Pressure Sensors) were further explored using two types of bipedal robots, the GERWALK and the BIPEDWALK (see Fig. 5). Essentially the GERWALK robot (7 servos) had 1-dof hip and ankle joints but it could swing its main weight (i.e. center of mass) left and right when walking, while the BIPEDWALK robot (8 servos) had 2-dof ankle joints which were used to shift its center of mass while walking.

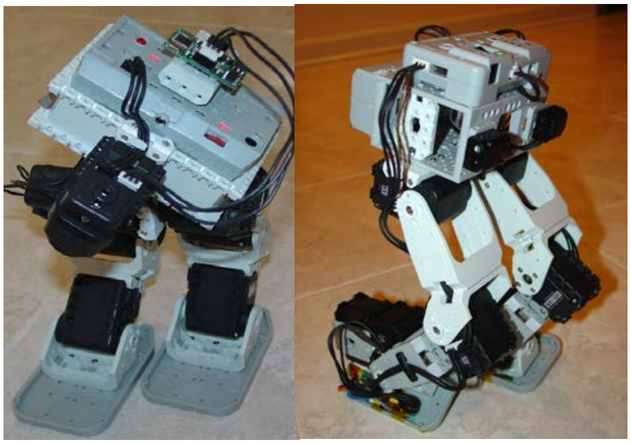

Figure 5. GERWALK with AX-S20 IMU Sensor (left)
BIPEDWALK with Foot Pressure Sensors (right).

Chapter 2 materials would be presented as follows:

- How to achieve "smooth" servo motions from "start" to "end" servo positions. Concepts of "Margin" and "Slope" parameters for a target servo position.
- Motion Programming using RoboPlus Motion tool as applied to GERWALK raising and lowering 1 leg one after the other.
- Procedure to interface with GERWALK equipped with AX-S20 (3D-IMU sensor able to record XYZ acceleration data at 20 Hz). Students are to design a data acquisition interface using the RC-100 buttons to mark up special events.
- **Lab 1:** Using above data acquisition tool to study two provided gait solutions for negotiating the last step of the stairs (see Fig. 4) – one would rotate the front foot, while the other one would not. Students are to experiment with their own gait solutions (yielding different results for the front foot) while recording the respective XYZ acceleration data and video clips of their trials (for later uploads to EMMA).
- **Homework 2:** To analyze data collected in Lab 1 to determine XYZ acceleration data boundaries when front foot was to rotate or not.
- Demonstration contrasting walking/ turning gait solutions between GERWALK and BIPEDWALK robots.
- **Project 2:** GERWALK or BIPEDWALK (student choice) negotiating stairs (see Fig. 4). The student could choose between two solution approaches. In the first approach, the bot would go forward and up the stair steps and then it would back down the steps. In the second approach, the robot would keep going forward during the up-stairs and down-stairs phases of this task. All students gait solution should not make the front foot rotate or at least keep it to a minimum. Students approach and intermediate results to be documented in EMMA. Videos of student demonstrations of Project 2 will be recorded and published on a public website (http://www.engr.uga.edu/~mvteachr/RobotVids/).

In Spring 2010, Chapter 3 introduced one-to-one Zigbee communications between 2 robots whereas students practiced shaping a 16-bit message into a standard Zigbee packet and how to send/receive and process such packets between a Master and Slave robot. For Spring 2011, Chapter 3 would be expanded to additionally cover broadcasting mode with the PC acting as base-station and with several car bots acting as mobile sensor nodes within a wireless sensor network:

- Open-loop control between 2 GERWALKs (Master-Slave configuration). Demonstration of TASK code that would send via Zigbee current ID and Present Position of each of the 7 servos used on the Master to the Slave which would then set its matching servos with these received values as their Target Positions. Students to experiment with varying physical distances between Master and Slave and document the subsequent Slave performance in mimicking the Master moves (in EMMA).

- Closed-loop control between Master and Slave GERWALKs. Demonstration of provided TASK code that would allow the user to manually set the Master into random poses while the Slave would repeat these poses via ZigBee (as before with the open-loop version). However if any of the Slave's servos are "constrained" for some reasons (i.e. their "Present Load" parameter will increase over a threshold of 512), the Slave should send back "appropriate" information about this situation over to the Master which would then "stiffen" up its corresponding servo(s) so as to inform the user that there are some restrictions on the Slave's motion at that time and that the user should not continue with his/her current operations.
- **Homework 3:** Students to use existing MOTION and TASK files for GERWALKs to create a new program that would allow the remote control (via RC-100) of a TWIN-GERWALK robot (see Fig. 6). The TWIN-GERWALK should be able to go forward, backwards and turn left or right given appropriate user remote commands (approaches exploiting inherent symmetry in twin robots and results to be documented in EMMA).

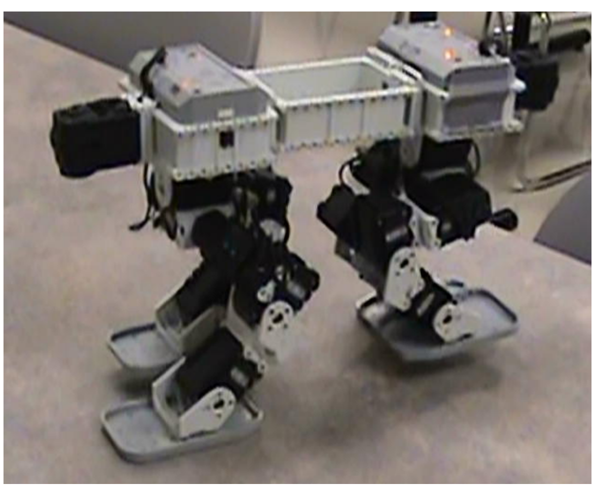

Figure 6. TWIN-GERWALK robot.

- Demonstration of C/C++ applications on the PC side for ZigBee communications between a PC and one CM-5 controller using the combination of 3 hardware modules USB2Dynamixel, Zig2Serial and Zig100 (see Fig. 2). TASK-type of programs will be used on the CM-5 side. Demonstration of the usage of C/C++ IDE (Visual C++ V. 6) including linkage to ZigBee library (provided by Robotis) and example codes (C++ and TASK files) for one-to-one and broadcast modes.
- **Project 3:** Mobile Wireless Sensor Network. Students to extend example C++/TASK codes to make the PC a base-station communicating via ZigBee to 3 car bots which are sending back the current values of their center NIR distance sensors, as they disperse from the base station. The goal is for the car bots to configure themselves into a relay system to "triple" the normal useable ZigBee range allowed by Robotis hardware. Using EMMA, students are to document their planning for the project such as choice of communication modes (1-to-1 or broadcast), packet design, carbot dispersal strategy as well as their enfolding implementation works. Videos of student demonstrations of Project 3 will be recorded and published on a public website.

In Spring 2010, Chapter 4 revisited servo control concepts presented in Chapter 2 and added the Torque Limit parameter to be used in a Force Control algorithm for controlling a gripper, and Chapter 5 introduced students to the AX-S20 sensor and the Robotis gyro-rate sensor that can be used for balancing humanoid robots. For Spring 2011, the plan is to remove the gripper and gyro-rate sensor materials from instruction to make room for the Foot Pressure Sensors (see Fig. 5):

- Demonstration of interfacing procedure with Foot Pressure Sensors (FPS) installed on a BIPEDWALK robot. Introduction to CALLBACK function.
- **Lab 2:** Students are to design a data acquisition interface with a running-average

function to smooth out FPS data which tend to be noisy. Students are to assess the pros and cons of putting this averaging function in the main program or in the CALLBACK section which is triggered every 7.8 ms (student's approaches and results to be documented in EMMA).

- Demonstration of example TASK code showing how a Callback Function could be used to read acceleration data from the AX-S20 and use them in computing and set the proper Joint Offset values to selected servos so that a humanoid robot could maintain its original balanced position even though the platform where it stood was being moved to different random angles with respect to the ground surface (see Fig. 7).

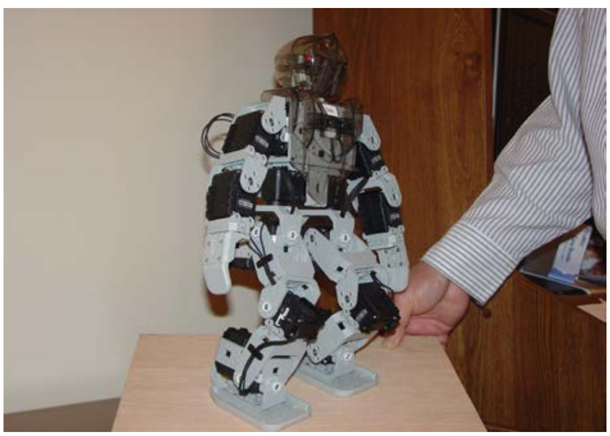

Figure 7. Humanoid robot balancing on a tilted platform using the AX-S20.

- **Project 4:** 1-Leg Balance for Humanoid or BipedWalk robot. Students to extend the approach used in the previous demonstration TASK code into solving this 1-leg balance challenge (see Fig. 8). The student could choose between the AX-S20 or the FPS as the sensing platform and the goal was to evaluate the effect (if any) of sensing platform choice on the overall performance of the balance process (how fast the platform tilt angle could be varied, what is the maximum tilt angle before balance would be lost, etc…). Students approach and intermediate results to be documented in EMMA. Videos of student demonstrations of Project 4 will be recorded and published on a public website.

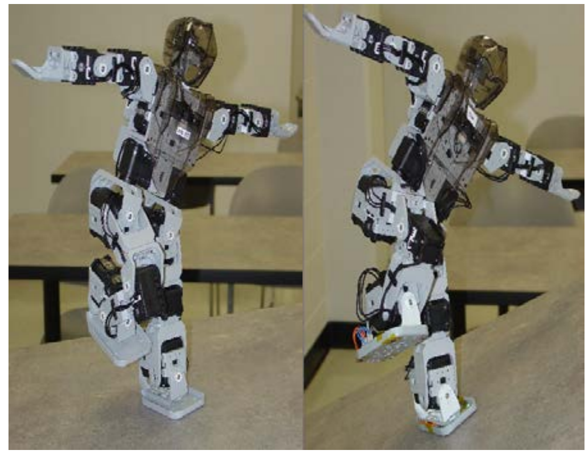

Figure 8. 1-Leg Balance using the AX-S20 (left) and the FPS (right).

Although some students had been exposed to PID control in their previous Control Systems course, it was for a single input to the system, while the 1-leg balance challenge involved the proper activation of several inputs (i.e. servos), thus students would have to use a trial-and-error approach. It would also be interesting to see how students would respond to the overload of the "knee" servo if they happened to start with a "bended knee" solution as a human would do.

Student enrollment for Spring 2011

The Spring 2011 semester started out with 7 students (4 from Agricultural Engineering and 3 from Computer Systems). One Ag. Eng. student dropped out after 2 weeks of class for lack of progress. All students had taken courses on Circuits and Sensors/Transducers. The Agricultural Engineering students had taken the first course in Microcontrollers while the CSE students did not, but the CSE students had attended many more courses on software engineering than the Ag. Eng. students. However all were currently taking "Feedback Control Systems", thus some problems are expected when students take on the last project about robot balancing on 1 leg which will be based on Proportional Control.

Implementation of EMMA tool

The plan was to organize EMMA into 5 sections, the first one being a student self-assessment for their knowledge and skills related to robotics, and the other 4 sections would correspond to the 4 chapters/projects. During this course implementation, we added 2 more sections on Wireless Data Acquisition and Master-Slave GERWALKS. The goal was for students to document their solution approaches and algorithm development activities so that the instructor could adjust lecture materials and assignments to emerging learning issues.

The student self-assessment data showed the Ag. Eng. students were not confident in their software skills as expected. The student EMMA submissions for the car bot maze project were useful to the instructor as they showed that students did not appropriately use selection structures such as “IF-ELSE-IF” and sequential “IFs”. These submissions also showed that students were not familiar with the “data-flow” approach to programming (i.e. involving a main endless loop). A quick check revealed that they took Sensors/Transducers with an instructor who did not use LabView as the programming environment. Subsequently more materials were presented to students to alleviate these problems, and in the next Project 1 we could see definite improvements in the CSE students programming performance, but unfortunately not with the Ag. Eng. students.

However, the instructor noticed that the EMMA submissions were fewer and fewer and getting more terse in content. The mid-term survey (2/18/2011) and the final survey (4/22/2011) confirmed that students did not like using EMMA for this purpose and their comments were:

- EMMA is not very effective and is a waste of time.
- EMMA is a helpful tool, but the interface needs some work.
- EMMA is a pain in the butt, I much prefer eLC (*UGA course management software*). I use eLC everyday, EMMA only when forced.
- EMMA is pointless and should not be used.

Another contributing factor for students could be that documenting their algorithm development in EMMA required that they had to “slow down” when working on their assignments, and the students did not see enough benefits in this behavior. Thus after the mid-term review, the instructor modified some of the homework assignments to become more lab-oriented activities whereas student problem-solving and code-generation problems could be ascertained and solved rapidly by the instructor as he visited each student workstation.

Results from the mid-term and final student surveys (5 responses out of 6 students)

This robotics course was taught in a special A Dual-Mode/Dual-Workspace classroom environment with dual displays and pen tablets available to students during lectures[12]. In the mid-term survey, students expressed that they like the dual-display feature as it allowed them to multitask (i.e. have the instructor desktop on 1 display, and work on their code on the other display), but they preferred to type than to use the pen-tablet to keep notes.

Camtasia Studio was used to capture lectures which were then available via UGA eLC within 2 hours. Students liked this feature a lots, some reviewed the lecture recordings once a week, some 2 to 3 times a week. One student commented “I missed 1 class period. The recorded lecture allowed me to attend class even when I was sick”, and throughout the semester no students had skipped class on purpose.

The mid-term (2/18/2011) and final (4/22/2011) surveys had 5 general questions and student responses were as follows:

1. What have you liked about the course this semester?
    a. Very hands-on, lots of example code. Team work.
    b. Very heavily project based.
    c. The programming of the bots.
    d. Very interesting subject matter. Excellent instructor.
    e. No tests. All project based.
    f. Projects were well designed.
    g. I liked lots of lab time.

2. What aspects of the course have been valuable for your learning this semester?
    a. It ties into other courses and explains new materials well.
    b. Understanding the differences of software/hardware programming.
    c. The use of code and value of professor.
    d. Hands on lab time are fun and valuable teaching tools.
    e. The different programming techniques of the bot.
    f. The actual projects helped understand software to hardware programming.
    g. In-class lab sessions.

3. What have you done that had helped you learn effectively in this course?
    a. Study examples and “play” with robots.
    b. Come to class every day.
    c. Complete outside work for the course.
    d. Study online lectures intensively.
    e. Spent time working in lab on the projects.
    f. Used slides when stuck on programming.
    g. Invested considerable time outside of class.

4. What had the teacher done that had helped you learn?
    a. Answer questions & provide feedback.
    b. Provide large amounts of sample code and reference.
    c. Build powerpoints.
    d. Very energetic and well organized lectures. Online materials are very helpful.
    e. Providing bots and example code. Upload resources.
    f. Frequent interaction with students.
    g. Letting us take the bot home.

5. What suggestions do you have for improvement?
    a. More gradual introduction of concepts/code.
    b. Less time spent on the basics on the hardware.
    c. The class would improve if the bots could be taken home for homework.
    d. More advanced robots would be interesting to observe/experiment with.
    e. Let the class learn the (zigbee) ID setting for setting up the bots.
    f. More examples of fundamentals before advanced.
    g. More in-class projects.
    h. Random partner pairings.
    i. More smaller simpler projects or assignments between projects.

To measure the effectiveness of instructional materials used in-class and outside-of-class, both surveys also asked students to respond to the following 7 questions using a 6-point Likert scale where "StD" meant "Strongly Disagree", "D" meant "Disagree", "SlD" meant slightly disagree, "SlA" meant "Slightly Agree", "A" meant "Agree" and "StA" meant "Strongly Agree":

1. In-class course materials delivery methods were effective.

2. I understood the materials presented during in-class lectures.
3. In-class materials presented via the second display were effective.
4. Recorded classroom lectures were useful.
5. Pre-recorded narrated tutorials were useful.
6. I felt comfortable going through multimedia presentations on eLeaning Commons.
7. I understood the materials presented in recorded lectures and narrated tutorials.

Student responses are shown in Table I (mid-term survey data in italics and final survey data in bold):

TABLE I. In-class & Outside-of-class materials effectiveness survey results. (mid-term survey in italics and final survey in bold)

Question #	**“StD”**	**“SlD”**	**“D”**	**“SlA”**	**"A"**	**"StA"**
1					*2*/**3**	*3*/**2**
2				*1*/**1**	*1*/**1**	*3*/**3**
3					*3*/**1**	*2*/**4**
4			*0*/**1**	*1*/**0**	*2*/**1**	*2*/**3**
5				*0*/**2**	*3*/**1**	*2*/**2**
6					*1*/**0**	*4*/**5**
7				*1*/**0**	*2*/**1**	*2*/**4**

Question 3 showed a slight increase in perceived effectiveness of the dual-display system, and Questions 6 and 7 also showed a slight increase in perceived effectiveness of the Camtasia lecture capture system. There was no substantial change in perceived effectiveness for in-class materials and delivery methods, but there was a light decrease in perceived effectiveness of the recorded lecture and pre-recorded tutorials as students were required to find their own solutions to the last 2 projects by applying and integrating on their own the materials already presented.

The students were also asked to evaluate the 3 course learning objectives:

1. Analyze a robotic problem description and conceptualize a solution based on computer systems engineering principles.
2. Have a good understanding of the functions of embedded robotic controllers and their wired/wireless communication programming.
3. Interface and control sound/light/ vision/acceleration sensors and servo motors to embedded controllers.

These student evaluations are shown in Table II (mid-term survey data in italics and final survey data in bold):

TABLE II. Student evaluations of 3 Course Learning Objectives(mid-term survey in italics and final survey in bold)

CLO #	**Not met**	**Met**	**Exceed**
1		*2*/**2**	*3*/**3**
2		*3*/**1**	*2*/**4**
3		*2*/**1**	*3*/**4**

Table II showed that for the later part of the semester students felt that they had learned more about the functions of embedded controllers and the interfacing of sensors.

Instructor Evaluation of Student Performance and Curriculum Issues

Videos of all student projects can be viewed at http://www.engr.uga.edu/~mvteachr/RobotVids/ whereas the readers can also evaluate on their own the student performances for the 4 projects required in this course.

In addition to the initial issue of proper usage of selection structures mentioned in a previous section, students still had not kicked the habit (since the freshman year) of not thinking through a solution first using flowchart or other directed graph tools, they still preferred to jump directly into coding, and as a result spent more time on any assignment than they should.

We had some equipment failure for Homework 2 (data acquisition on the 3-D IMU sensor AX-S20) thus we had only one GERWALK equipped with the AX-S20 for the whole class thus it became an unintended group project. In future years, these materials will have to be presented as a demonstration as the AX-S20 is no longer available from Robotis.

The "closed-loop Master-Slave GERWALKS" assignment turned out to be rather challenging for the students as this was the first time that they had to deal with communications programming concepts such as message shaping, coding and decoding, and especially closed-loop programming algorithm whereas Master and Slave bots behavior influence each other. Considering these difficulties for the students, a curriculum change was made for Project 3 by letting the students choose between "Twin-Gerwalks" (TG – originally planned as Homework 3) or "Mobile Wireless Sensor Network" (MWSN – originally planned as Project 3), as both deal with the "broadcast" ZigBee mode and the same level of complexity in shaping message packets. The MWSN project would require more skills in C++ programming on the PC side which are lacking in the Ag. Eng. students, while the TG project could be achieved using only the RoboPlus programming environment which all students are familiar with. The students were paired for Project 3 and at the end all 3 teams chose the TG project as they were not confident about their command of the C++ language. Although all 3 teams came up with a working solution, only 1 team achieved all project requirements while the other teams could not achieve reliable coordination of motions between the Leader and Follower bots.

As mentioned before, all Spring 2011 students happened to be taking the "Feedback Control" course concurrently with this Robotics course, thus they may not have mastered Proportional Control concepts by the time we get into Project 4 (1-leg bot balance control on a varying inclined plane), another curriculum change was made by letting the students use the "simpler" BIPEDWALK bot equipped with FPS (with 8 servos as legs only) over the Humanoid bot which is a more complex system with 18 servos. Students were also paired for this project and were allowed to choose between 2 possible balance configurations: a) counterweight leg bent and b) counterweight straight up (see Fig. 9). At first one team chose the straight-leg configuration but soon they had found that a high CG configuration was harder to maintain balance, thus at the end all 3 teams chose the bent-leg configuration and achieved the goals of this project.

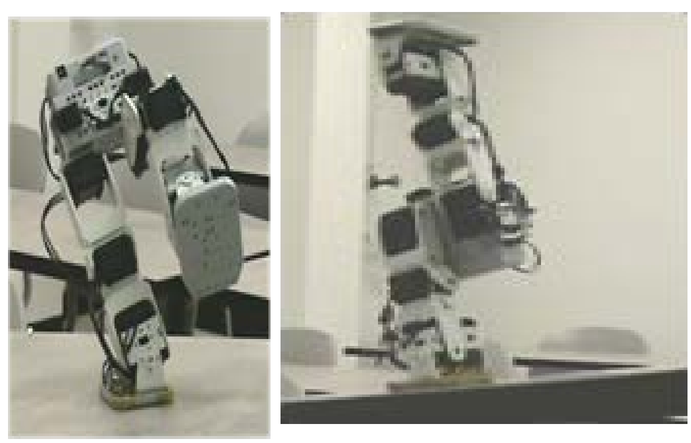

Figure 9. 2 possible configurations for the BIPED bot doing a 1-Leg balance on a varying inclined plane.

Interestingly, we had found some anecdotal evidence that Agricultural Engineering students had a better "sense" of mechanics (very much needed in bipedal robotics) such as CG location and dynamic flow between motions than the Computer Systems Engineering students who were, on the other hand, definitely better at software engineering.

Conclusions

Applying principles for "Smart Teaching" from the book "How Learning Works", we believed to have obtained a more focused and more challenging senior-level robotics with better scaffolding features to help students master robotics concepts in the area of sensor interfacing and actuator control, as well as wireless communications aspects in a multiple controller's environment. Results from our mid-term and final student surveys showed good acceptance from our Spring 2011 students.

Our particular engineering students were not confident in C++ programming techniques as they should, thus in future implementations of this course we will implement more C++ assignments in the first part of the semester so as to review and build up C++ skills for our students to get them ready for the later projects.

We had found that communications programming concepts were particularly hard for students to understand and integrate into robotics programming. Although we used Robotis equipment exclusively (because we just happened to have them), we hope that interested readers can see beyond our current equipment usage to extend our approach to their own needs. The FIRE project from Carnegie-Mellon University is releasing soon their Arduino-based multi-robot development system which could be more economical than the Robotis system (http://www.education.rec.ri.cmu.edu/fire/multi-robot/index.php).

And finally, we were not successful in using EMMA as a tool for collaboration between instructor and students to document and help improve student algorithm development skills as students found EMMA not effective and not responsive enough (some students preferred Google Docs).

Bibliography

1. T. Padir, M.A. Gennert, G. Fischer, W.R. Michalson, and E.C. Cobb, "Implementation of an undergraduate robotics engineering curriculum", Computers in Education Journal, vol. I, no. 3, pages 92-101, 2010.

2. M. Boutell, C. Berry, D. Fisher and S. Chenoweth, "A multidisciplinary robotics minor", Computers in Education Journal, vol. I, no. 3, pages 102-111, 2010.

3. C.A. Berry, "Mobile robotics: A tool for application-based integration of multidisciplinary and undergraduate concepts and research", Computers in Education Journal, vol. I, no. 3, pages 67-80, 2010.

4. N. Correll and D. Rus, "Peer-to-peer learning in robotics education: lessons from a challenge project class", Computers in Education Journal, vol. I, no. 3, pages 60-66, 2010.

5. I. Nourbakhsh, "Robot Diaries: Creative technology fluency for middle school girls", IEEE Robotics & Automation Magazine, vol. 16, no. 1, pages 16-18, 2009.

6. B. Bishop, J. Esposito, and J. Piepmeier, "Moving without wheels: educational experiments in robot design and locomotion", Computers in Education Journal, vol. I, no. 3, pages 41-49, 2010.

7. S.A. Ambrose, M.W. Bridges, M. DiPietro, M.C. Lovett, and M.K. Norman, How Learning Works. San Francisco, CA: Jossey-Bass, 2010.

8. C.A. Tomlinson, and J. McTighe, Integrating Differentiated Instruction and Understanding by Design. Alexandria, VA: ACSD, 2006.

9. I. Mezei, V. Malbasa, and I. Stojmenovic, “Robot to robot: communication aspects of coordination in robot wireless networks”, IEEE Robotics & Automation Magazine, vol. 17, no. 4, pages 63-69, 2010.

10. N. Correll and A. Martinoli, “Multirobot inspection of industrial machinery: from distributed coverage algorithms to experiments with miniature robotic swarms”, IEEE Robotics & Automation Magazine, vol. 16, no. 1, pages 103-112, 2009.

11. M. Vukobratovic and B. Borovac, “Zero-moment point – thirty five years of its life”, International Journal of Humanoid Robotics, Vol. 1, no. 1, pages 157-173, 2004.

12. C. N. Thai, ““A Dual-Mode/Dual-Workspace classroom environment”, Computers in Education Journal, vol. XVIIII, no. 1, pages 12-23, 2009.

Biographical Information

C. N. Thai is an Associate Professor at the Biological & Agricultural Engineering department of UGA. He teaches courses in Robotics, Machine Vision and Systems Simulation. His main research area is in Spectral Imaging for plant health and quality characterization of agricultural products.

J. M. Mativo is an Assistant Professor at the Workforce Education, Leadership and Social Foundations Department of UGA. His instruction duties are in both workforce education and engineering. His research centers in engineering education and mechanical engineering. Dr. Mativo joined the University of Georgia in 2007.