Teaching Robotics to Students with Mixed Interests

C. N. Thai
University of Georgia
Biological and Agricultural Engineering Department
Athens GA 30602-4435
E-mail: thai@engr.uga.edu
Web site: http://www.engr.uga.edu/~mvteachr

Abstract

An innovative and flexible course in Robotics was created to accommodate the interests of engineering and non-engineering students at UGA. It sought the integration of multi-threaded programming, advanced sensors interfacing (video camera and balance sensor), actuators control and wireless communications (Zigbee). It was based on the Bioloid Expert and Premium robotics systems from Robotis, Inc. Details of course design and materials, and of student projects are described herein.

Introduction

Educational interests in robotics at the University of Georgia are quite diverse:

- The Artificial Intelligence Center (Computer Science Department) emphasizes intelligent autonomous systems.
- The Department of Theatre and Film Studies is interested in using robots for dramatic productions [1].
- The College of Education wants to incorporate robotics in their graduate degrees in Technology Education (http://www.uga.edu/teched/).
- In the Engineering Education area, we are looking at using robotics as an instructional approach to integrate hardware, software and communication technologies at a senior-year level whereas students would already have taken courses on Microcontrollers and Sensors and Transducers.

Furthermore, the field of Robotics itself is also quite diverse as described in several robotics handbooks [2][3]. Thus our curriculum plan was to create a course that can serve as an elective for the B.S. degree in Computer Systems Engineering and also for the Mechanical Systems or Electrical & Electronics emphasis areas of the B.S. degree in Agricultural Engineering.

Currently there are several suitable robotics textbooks for university level students [4][5][6] and several other works more oriented towards pedagogy, for small children as well as for engineering students [7][8][9]. To obtain flexibility, this course goal was to provide students with a medium to advanced practicum in Embedded Robotics wherein the students will learn about the programming of embedded controllers, the actuation of servo motors, the interfacing of sensors (sound, light, acceleration, wireless color video camera), inter-computer serial communications (RS-232 and ZigBee), and the control of autonomous as well as remotely piloted systems. Also to accommodate different student knowledge and skills, the students would start programming with a high-level integrated (but somewhat constrained) programming environment, and then towards the semester's end, the students would practice lower-level programming using the C/C++ language depending on the term project chosen.

Course Design

Hardware/Software System Description

Balancing between potential sophistication of robot design and cost affordability, we decided on the Bioloid Expert and Premium systems from Robotis (http://support.robotis.com/en/). With the Expert system, we can build from basic car and bipedal robots to humanoid systems with wireless color video cameras (see Fig. 1). With the Premium system, we can add angular rate gyroscope, 3-D inertia measuring and magnetic heading sensors (see Fig. 2).

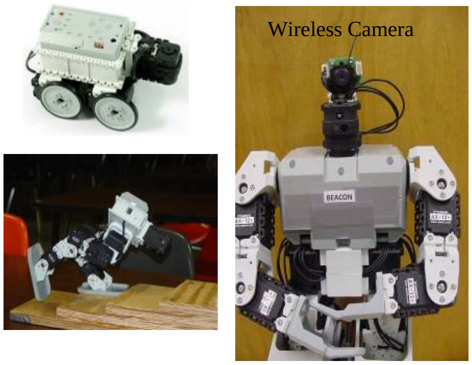

Fig. 1 – Bioloid Expert system configurations.

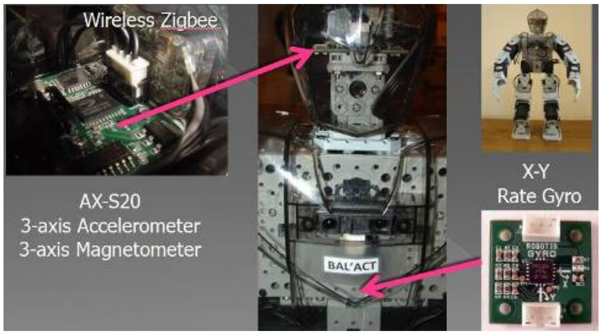

Fig. 2 – Bioloid Premium's advanced sensors.

Regarding software tools, 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).

The Bioloid embedded controllers are based on the Atmel AVR microcontroller at 16 MHz, with either 128 KB or 256 KB of Flash memory. The user-accessible memory area is divided into 3 independent and cooperating sections:

a) 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.

b) 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.

c) 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 (e.g. humanoid). 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 sensors. Up to 7 poses, along with transitioning time periods between these poses, can be combined into 1 motion page which is "played" during run-time producing dynamic robot gestures or maneuvers. A maximum of 127 motion pages can be defined with a 128 KB system, while a 256 KB system would allow up to 255 motion pages.

Instructional Design

Building upon other engineering educators experiences in teaching robotics [10][11][12], this semester course 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.

Foundational materials were presented in Chapter 1 for the following concepts developed around a Carbot system (see Fig. 1):

- 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. 1.0.17.0)
- Hands-on practice using "MANAGER" - direct hardware observation tool and "TASK" – main IDE tool:
  - Programming Servo Motors for the "Continuous Rotation" mode. Interfacing NIR and Sound Sensors integrated into Carbot.
  - Autonomous obstacle avoidance programming for Carbot (2 approaches - Reactive Control and Behavior Control). Application to maze navigation.
  - Remote user control via Zigbee controller (RC-100). How to deal with signal loss.
- Homework 1 – Add new "speed level" commands to existing TASK code that was made to accept Up-Down-Left-Right commands from user (via the RC-100).
- Project 1 – Automated Carbot Collision Avoidance. 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 Carbots (both going forward under remote control by separate students using RC-100s). When the front Carbot suddenly stopped, the rear Carbot 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 Carbot. After the rear Carbot performed successfully a passing maneuver, user remote control commands would once again be accepted and acted upon by this Carbot.

In Chapter 2, programming concepts for the (absolute) "Position Control" mode for servo motors were further explored:

- How to achieve “smooth” servo motions from “start” to “end” servo positions. Concepts of “Margin” and “Slope” parameters for a target servo position.
- Homework 2 - Independent (multi-threaded) control of multiple servos.
- Motion Programming using RoboPlus Motion tool as applied to simple bipedal bot (GERWALK) as shown in Fig. 1.
- Homework 3 – Wireless Remote Control of GERWALK bot combined with autonomous obstacle (wall) avoidance.
- **Project 2**: GERWALK negotiating stairs (see Fig. 1). 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 bot would keep going forward during the up-stairs and down-stairs phases of this task. Figure 3 displays both approaches.

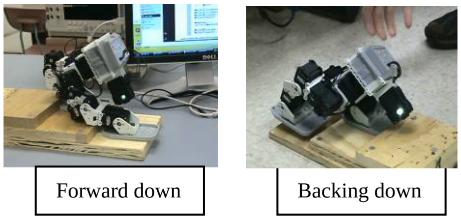

Fig. 3 – Two possible down-stairs gait solutions.

Chapter 3 expanded Zigbee communication concepts further into inter-bot communication packets manipulation whereas the students practiced shaping a 10-bit message into a standard Zigbee packet:

- This concept was demonstrated using an example TASK code that would send via Zigbee the current ID and Present Position of each of the 7 servos used on a Master GERWALK bot to the embedded controller on another GERWALK bot set in Slave mode. The slave controller would then set its matching servos with these received values as their Target Positions, i.e. the Slave bot would imitate/repeat the Master bot motions (with a small time delay – Zigbee packets were sent at 57.6 kbps). Essentially this was an open-loop control application between the 2 bots.
- Homework 4 – Starting from this open-loop sample code, the goal was for the student to design a closed-loop solution that would allow the user to manually set the Master GERWALK bot into various random poses while the Slave Gerwalk bot would repeat these poses via ZigBee communications (as before with the open-loop version). However if any of the Slave bot’s servos are “constrained “ for some reasons (i.e. their “Present Load” parameter will increase over a threshold of 512), the Slave bot should send back “appropriate” information about this situation over to the Master bot which would then “stiffen” up its corresponding servo(s) so as to inform the user that there are some restrictions on the Slave bot’s motion at that time and that the user should not continue his/her current bot operations.

Chapter 4 revisited servo control concepts presented in Chapter 2 and added the Torque Limit parameter to be used in a Force Control algorithm:

- Interactions between various servo parameter settings: Goal Position, Slopes, Margins, Punch, Present Position, Present Load, and Torque Limit.
- Gripper Design & Programming (see Fig. 4)
- Gripper Force Feedback & Joint Offset programming with Motion Page and Callback Function.

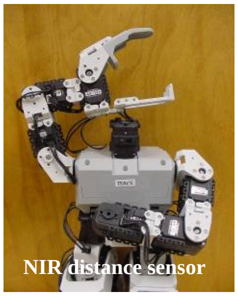

Fig. 4 – Bot “Teal’c” with Gripper (right hand) and NIR distance sensor (left hand).

Chapter 5 introduced students to sensors that can be used for balancing robots (see Fig. 2):

- The AX-S20 sensor can provide data for X-Y-Z accelerations and azimuth-roll-pitch angles at a 20 Hz refresh rate. As the azimuth angle was derived from a magnetometer, the AX-S20 was mounted in the bot’s head so as to avoid magnetic interferences from the servo motors as much as possible.
- The gyro sensor can only provide X-Y angular velocities (Pitch and Roll) but at a faster refresh rate of 30 Hz. It is installed in the bot’s belly (i.e. close to its C.G. as much as possible).
- An example TASK code was demonstrated to students 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 the bot 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 (slowly of course due to the limitation of the 20 Hz refresh rate of the AX-S20). At present, we are still working on developing code using the gyro to allow the bot to walk on uneven terrain.

For the rest of the semester, there were no more formal lecture given, as the students were working on the final Project 3 and only needed occasional consultations with the instructor on some specific issues. The pedagogical approach used for Project 3 was inspired by the work of Patricia Stokes "Creativity from Constraints" [13], thus 3 humanoid bots were built with different capabilities/constraints and tasks to be performed and the students could choose their bots to match their own interests and abilities:

- The "Teal'c" bot (see Fig. 4) 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 the dowel bundle, turned an appropriate amount to face the bundle, approached it within a proper distance, then grabbed and lifted up the bundle (see Fig. 5). This project could be fulfilled using the RoboPlus TASK and MOTION tools.

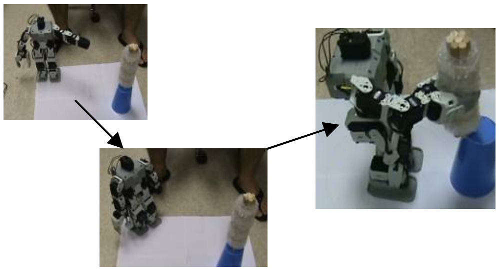

Fig. 5 – "Teal'c" bot locating and grabbing dowel bundle.

- The "Beacon" bot (see Fig. 1) 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 (see Fig. 6).

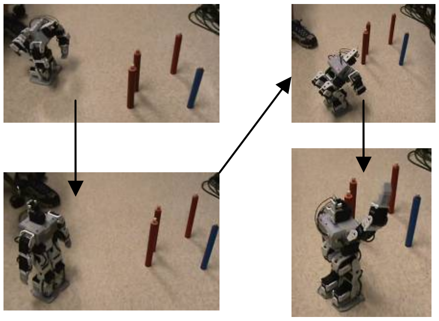

Fig. 6 – "Beacon" bot bending down to locate blue dowel, avoiding red dowels and finally reaching its destination.

  This project would require quite a bit of software integration work. As the wireless video camera was an analog USB device inputting into a PC, the student would have to create a Visual C++ program on the PC side to collect video frames in real time, process them to find the red and blue dowels, decide on the proper maneuvers for the bot, determine the equivalent bot commands to send through another USB to Zigbee device from the PC to the bot itself which would then decode and execute the commands received. Thus the "brain" was on the PC and the bot itself acted as an avatar. This project would require the RoboPlus TASK and MOTION tools, and also the Expert System binary C++ libraries for various function calls to the video camera and the USB-to-Zigbee communication device.

- The "Bal'Act" bot task (see Fig. 2) was to use the AX-S20 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. The student would be provided with an example TASK code that first would set the bot into a static ready pose on a horizontal platform and record the corresponding "balance" values of the Pitch-Roll (X-Y) accelerations. Later when the platform's angles were changed by the user, the Callback function would read the current X-Y accelerations, compute the required Joint Offsets to apply to the hip-leg-ankle servos in order to bring the X-Y accelerations back to the original "balance" values. This project could be fulfilled using only the RoboPlus TASK and MOTION tools.

Course Implementation

Spring 2010 was the first time that this course was offered, and 5 students enrolled for it: 3 senior-level engineering students, 1 graduate student from the Department of Mathematics and Science Education (MSE) and 1 graduate student from the Department of Theatre and Film Studies (TFS). During the first half of the semester, all 5 students were instructed together through Chapters 1 & 2, and Projects 1 & 2. From then on, each student pursued his or her own Project 3:

- The MSE student wanted to create a week-long short course for high-school teachers in robotics.
- The TFS student wanted to use a Premium Humanoid bot (see Fig. 2) and apply Disney animation principles [14] to the creation of "expressive" bot motions. He also wanted to compare the "Disney" results to those derived from standard inverse kinematics (which is a new tool provided inside the Robotis Motion tool).
- The 3 engineering students chose to work with the "Teal'c", "Beacon" and "Bal'Act" humanoid robots.

At this point in time, the MSE student had finished successfully with her final project. The "Beacon" project was also concluded successfully. The "Teal'c" project was achieved in 3 independent components: dowel bundle location with the left arm and appropriate body rotation to face the bundle, new walking gait to accommodate the extra weights due to the gripper and NIR sensor at the end of Teal'c arms, and in-place automatic body rotation to locate the bundle, grab and lift bundle *(video clips had been obtained of these projects and public web links for them will be created shortly)*. Several control strategies had been tried unsuccessfully for project Bal'Act and the latest approach was to apply the servo balancing algorithm only for 0.125 second at the end of each motion page.

The official student course assessment data are not yet available from the BAE department, but informal student feedback indicated that this course was well accepted in terms of materials delivered and the challenging projects. The one complaint was about the RoboPlus Task tool being too "hobby-ish".

Conclusions

Our first attempt at creating an innovative and flexible robotics course to accommodate a range of interests at UGA was mostly successful. For some of the final projects, we had underestimated the level of complexity required. At the next iteration of this course, the plan is to go over the foundational materials more quickly and expand more on wireless and embedded techniques using C++ language.

References

1. H. B. Segel, *Pinocchio's Progeny.* Baltimore, MD: The John Hopkins University Press, 1995.

2. B. Siciliano, and O. Khatib (eds), *Springer Handbook of Robotics*. Berlin, Germany: Springer-Verlag Publishers, 2008.

3. T. R. Kurfess (ed), *Robotics and Automation Handbook.* Boca Raton, FL: CRC Press, 2005.

4. T. Braunl, *Embedded Robotics*. Berlin, Germany: Springer-Verlag Publishers, 2006.

5. M. J. Mataric, *The Robotics Primer.* Cambridge, MA: MIT Press, 2007.

6. C. Chevallereau, G. Bessonnet, G. Abba, and Y. Aoustin (eds), *Bipedal Robots.* London, UK and Hoboken, NJ: ISTE Ltd. and John Wiley & Sons, 2009.

7. M. Umaschi Bers, *Blocks to Robots.* New York, NY: Teachers College Press, 2008.

8. A. Druin, and J. Hendler, *Robots for Kids.* San Diego, CA: Academic Press, 2000.

9. D. J. Ahlgren, and I. M. Verner, "Building Self-Efficacy in Robotics Education", *Computers in Education Journal*, vol. 18, no. 1, pp. 18-27, 2008.

10. I. M. Verner, D. J. Ahlgren, and D. P. Miller, "Robotics Olympiads: A New Means to Integrate Theory and Practice in Robotics", *Computers in Education Journal*, vol. 17, no. 4, pp. 11-21, 2007.

11. A. Reed, T. Creekbaum, M. Elliot, D. Hall, and D. Harbour, "Utilizing Robotics to Facilitate Project-Based Learning: A Student Perspective," *Computers in Education Journal*, vol. 19, no. 1, pp. 2-11, 2009.

12. R. Avanzato, "Assessment and Outcomes of Robot Competitions at Penn State Abington," *Computers in Education Journal*, vol. 19, no. 3, pp. 9-16, 2009.

13. P. D. Stokes, *Creativity from Constraints.* New York, NY: Springer Publishing Co., 2006.

14. F. Thomas, and O. Johnston, *The Illusion of Life: Disney Animation*. Rev Sub Edition, New York, NY: Disney Editions, 1995.