High School Robotics Course: The Teacher View
A. Scott Hagin
Math Teacher, Edwardsville High School, shagin@ecusd7.org

High School Robotics Course: The Teacher View

Introduction

This paper is intended to offer encouragement to those High School teachers who have ever thought, “Wouldn’t it be valuable for students at our school to have a robotics course?” You can do it! I also write this paper in humble appreciation to all those who continue to sacrifice for the education of our future engineers and scientists. The results of your imagination and efforts give a meaningful answer to the search by school boards and parents for student-motivating content. Thank you KIPR and SIUE!

In 2007, Edwardsville High School (EHS) developed an elective robotics course in the Math department that provides an alternative setting for learning and problem solving. The course was designed to supplement mathematics education for students that struggle at the Algebra 2 level and provide a pre-engineering laboratory course for students preparing for college computer science or engineering majors. While multiple course objectives review science and math concepts, or introduce programming and technology skills to the students, the primary focus of this course in our Mathematics curriculum is to improve student problem solving. This paper describes the course, called Introduction to Robotics.

Introduction to Robotics is a course that uses a curriculum pieced together from multiple resources. After describing the curriculum, the teaching approach is highlighted. Specifically, the paper addresses how students are placed in the role of an engineer and expected to complete tasks related to complex system development. A description of student assessment is also provided. The paper concludes with a course evaluation from the teacher’s and students’ perspectives.

For more complete student course evaluations for *Introduction to Robotics*, please read papers written by 2 students in the class.[1, 2]

What to teach

Deciding what content should be included in the course was directed by the primary course intent to engage students in problem solving. The author’s five years of Botball coaching experience and a previous career in engineering development largely influenced specific topic choices. As in any new course the decisions about content were also guided by the obvious constraints of expected student ability and time.

The prerequisite for Introduction to Robotics is completion of a semester of Algebra II. At EHS, that translates into a course open to 10th – 12th grade students. This prerequisite also creates opportunity for a class of students with a wide range of analytical thinking abilities and computer programming experience. The course is taught in a one semester (18 week) term with class meetings for 54 minutes, 5 days a week.

---

[1] Parker, Aaron. High School Robotics Course: The #1 Student View. Submitted to GCER 2008.
[2] Ray, Matthew. High School Robotics Course: The #2 Student View. Submitted to GCER 2008.

The course curriculum is based on the KISS Institute for Practical Robotics (KIPR) approach to learning about robotics, as taught during Botball professional development workshops[3]. It is organized into 6 units (Table 1).

Table 1. Introduction to Robotics Course.
Unit 1: Introduction to Robots and Robot Building
Unit 2: Sensors & Programming Part I
Unit 3: Robot Structures and Mechanics
Unit 4: Sensors and Programming Part II
Unit 5: Engineering and the Systems Engineering Process
Unit 6: Course Final Project

Primary sources for course lesson and activities include the following:

1. KIPR Botball Professional Development Workshop tutorials
2. NASA *Robotics with the XBC* course[4]
3. Carnegie-Mellon University (CMU) developed Robotics Educator 2.5[5]
4. CMU developed C-based Programming Educator[6]

These resources are used to meet a substantial number of the planned desired learning outcomes for each unit established in developing the course (attachment 1). Before commenting on which parts of the resources are applied in the course, I will describe the robot equipment and learning approach.

How to teach

The robot equipment available to students in the class includes an XBC Robot Starter Kit[7] for each project team, 2 iRobot® Create™ robots (with connecting cables for the XBC version 3)[8], and equipment from the previous 5 years of Botball teams. Interactive C is used for programming robots. The course meets in a computer lab with 28 standard desktop pc's using windows XP professional, networked to provide each student password protected server storage (teacher accessible), a common (student read only, teacher read/write) server space, and internet access. The lab contains adequate work table space for robot construction and a storage cabinet for team robot kits.

The general course learning approach is discovery based, organized around a 2 or 3 member project team. Since a major objective of the course is learning the engineering process that involves practicing the communication skills required for teamwork, team selection is done very deliberately. Project team formation is based on a survey given to students the first week of class to understand each student's analytical ability (level of Math completed), computer programming experience, and communication ability (attachment 2). An attempt is made to pair students with complementary ability in the three areas (Math, Programming, Communications).

[3] http://botball.org/about-botball/overview.php
[4] See NASA's Robotics Alliance Project course archive at http://robotics.nasa.gov/courses/summer06/index.php
[5] See http://www-education.rec.ri.cmu.edu/roboticscurriculum/index.html
[6] See http://www-education.rec.ri.cmu.edu/roboticscurriculum/index.html
[7] https://botballstore.org/catalog
[8] Provided through a technology grant from the District 7 School Board Citizen Advisory Council.

Grade level was not used as a factor this year. Seventeen students formed 8 project teams in the class – 1 team of three students, 6 teams of two students, and a team of two teaching assistants.

Specifically, the approach used for teaching each unit was to introduce a topic through presentation or demonstration, give students an activity to reinforce the topic, provide direction on accessing the KIPR and CMU tutorials, examples, and code templates related to the topic, and finally, assign the project teams a robot building task with a specific grading rubric and timeline. Attachment 3 gives examples of how the primary resources are used in each unit. Table 2 outlines the robot project tasks for each unit.

Table 2. Project Tasks	
Unit 1:	**Ping – pong robot using basic motor commands and touch sensors.**
Unit 2:	**Line following using basic motor commands and light sensor.**
Unit 3:	**Build 243:1 gear box; "fastest" brick-pushing robot with gear box of choice.**
Unit 4:	**4A: Find and grab orange ball using servos, ET sensors, and back-EMF libraries.** **4B: Find and grab ball with color camera.**
Unit 5:	**Write a Design Synthesis Document that includes a requirements and functional analysis.**
Unit 6:	**Botball 08 game challenge.**

One of the advantages in starting this course after participating in Botball for several years is availability of students who have already learned how to design and build robots. Two of the most experienced members from the EHS Botball team were students in this year's course, so they were assigned teacher assistant duties. Their tasks in the class included building demonstration robots, writing template programs for projects, and helping to plan the unit project tasks. While the TA's were not involved in assessing other students' work, their tutoring and observations gave good information to help gauge difficulty of course content and pacing for the inexperienced students. Unit project complexity was tailored to reflect ongoing learning progress. Each unit's project task also provided the focal point for assessing student learning.

Student Assessment

Table 3 outlines the grading scale used as the structure for student assessment. Note that the grades received for projects comprise most of the course grade (70%). At the start of each project, students are provided a rubric that describes the specific task. Attachment 4 provides one example of a project rubric. The semester exam required students to present the class a 10-20 minute overview of the engineering design, the hardware and the software concepts applied in their final robot project (attachment 5).

Table 3. Grading Scale		
Grade Scale %	**Grade Criteria**	**Semester Grade Criteria**
A 90.0 - 100 B 80.0 - 89.9 C 70.0 - 79.9 D 60.0 - 69.9 F Less than 60	Quizzes 15% Daily Work 15% Projects 70%	1^{st} Quarter Grade 40% 2nd Quarter Grade 40% Semester Exam 20%

Course Evaluation

Teaching a robotics course, like any teaching, is hard work. The challenge in teaching this course is compounded by the limitless outcomes that are possible when giving students a box of many technologies and saying, "build a robot to complete this task." Since the class finished just last week, an evaluation of what was good and bad in the course demands more reflection time; however, the following thoughts give some insight into what seemed to work and what could be improved in terms of the course curriculum, the teaching approach, and the assessment of students. Student comments included below came from a feedback discussion with students on May 30, 2008.

What worked

course curriculum

The majority of students were successful in meeting the course objectives. Students commented that it was beneficial to start with basic robot construction, sensor, and programming concepts before getting the more advanced tasks later in the course. Due to limited time, three initially planned objectives dealing with complex subjects were not presented: use of proportional robot control, use of arrays and random numbers in IC, and introduction of systems engineering software tools (attach 1).

The curriculum resources available (KISS, CMU, NASA) were easy to tailor for specific lessons. They were also well suited to introducing students to topics and provided accessible references for student use. Accessibility to tutorials and detailed explanations was important in helping the more capable students stay motivated, by enabling them to explore topics not yet introduced by the teacher in the course.

A significant supplement to the curriculum was a field trip to the SIUE School of Engineering where students were introduced to college-level robotics challenges. After their visit to SIUE's robotics lab, students frequently referred to what they saw -- some design feature or applied technology (PSoC)[9].

the teaching approach

Team formation generally worked well and students agreed that the team approach to learning was superior than working alone. After the first project, it was clear most of the teams established primary and secondary roles for individuals. For example, almost every team

[9] http://roboti.cs.siue.edu/projects/

established one student as the primary programmer and one student as the builder. Student consensus seemed to be that the team size of 2 students was ideal; this was confirmed from the 3-person team comment that sometimes they didn't need a third person.

One of my initial concerns was the ability of students to learn the required programming fundamentals in order to be successful at controlling a robot for more complex tasks. The programming templates provided for each project task were sufficient, in most cases, to allow students to write successful programs. Students were expected to understand how to read and interpret introductory IC – but they were not required to create original software, as long as the team could modify a template for their specific robot's task. The TA's were great in explaining and demonstrating the use of more advanced programming techniques, such as creating custom movement or camera libraries of functions.

The teams with experienced programmers required less time for basic project tasks, so each project requirement was specified with "plus ups" for either a higher grade or extra credit (attach 4 as an example). One of the advanced teams was required to use a create robot as the platform for their final project. This team's achievements made it clear that the opportunity for such differentiated learning should be encouraged in a robotics course. One of their initiatives was to write a Java application that generated an output file of IC code to play music on the create. While the create team had no problems in programming the XBC/create, they had quite a challenge in building structures for adding sensors and a camera on the create.

The hardware available to students held together for daily classroom use. At the outset there was uncertainty in the physical robustness of an non-encased robot controller. Students packed and unpacked robot kits every class day, so frequent opportunities for accidents or mishandling existed. The XBC kits tolerated the classroom environment. The only hardware casualties were a few malfunctioning sensors and one broken XBC serial connector. Previous years Botball kits provided the supplies for replacement parts.

assessment of students

The ability of the team to design and build robots that performed increasingly complex tasks provided the evidence for how well individuals on a team improved their problem solving ability. Use of project task rubrics was a reasonable attempt at quantifying application of knowledge in each unit. But, a more meaningful insight for understanding any improvement in problem solving was what students said during their oral project reviews. Such project reviews included impromptu (1 day's notice) brief presentations to the class and the more formal presentation such as the course semester exam (attach 5). Improvement in problem solving was indicated by students discussing different kinds of challenges in subsequent project reviews. Teams that reported the same challenge over and over seemed to indicate less progress in understanding in how to solve a problem.

What could be improved

course curriculum

There are at least three major curriculum improvements to consider for this course. One of the resources available through Botball that would benefit students in this course is use of the simulator as a way to develop and debug code (see Botball 2008 workshop slides). Use of the simulator could give students more opportunities to practice writing and using code before adding the complexity of hardware integration. Another shift in the course would be to integrate the systems engineering concepts earlier in the course. For example, a consistent deficiency in the student approach to many of the projects was a lack of effort in collecting and documenting test data. More deliberate use of design and test documentation procedures early in the course might help establish systematic practices. Finally, one of the TA's recommended an advance robotics course be developed that could better challenge the students experienced in programming. Perhaps the curriculum could be two-tiered with more programming intense projects for those that need a greater challenge.

the teaching approach

According to the students, if they could make only one improvement to the course it would be to eliminate the daily engineering journals they were required to keep. One student idea was to keep a team journal rather than individual journals. The challenge is to integrate this documentation (and reflection) tool into their daily work process. Perhaps a preformatted, online or server based journal might be more appealing, realistic, and save student time.

As mentioned above, the team size of 2 seemed about right; except, the 3-student team did point out that it was very beneficial for them to have an extra brain during team brainstorming, design and planning sessions. An alternative team structure might be a "super team" assignment, where a 4-student team would be used for planning and design, then students paired up for prototype development, building, and testing. Students gave a favorable head nod to that idea. Such a "super team" might also allow students to dedicate more time for documentation tasks and better absorb the effect of student absences.

One final improvement to teaching approach would be to more fully use the create. It serves as a stable platform to learn how to use a camera for object searching and tracking. Such a combination would be useful for teaching camera applications even to students who are inexperienced programmers.

assessment of students

A key improvement to assessment might be to have more formal quizzes with the purpose to motivate better student retention of fundamental programming and sensor definitions. On more than one occasion during project designs, a student would puzzle over a problem that could be easily solved had the student recalled a basic concept from earlier in the course. "More quizzes" was also one of the observations made by the senior TA.[10]

Another open assessment question is whether or not shorter, more frequent projects might better prepare students for the final complex project, rather than single major projects for each unit.

In conclusion, the EHS experience verified that an introductory High School Robotics course is a meaningful addition to curriculum that needs to prepare students to solve real-world problems. Experienced Botball coaches and mentors have much of the background and resources right at their fingertips. The bonus for teaching such a course is participating in a class with motivated students focused on learning, whether the students realize it or not. You can do it!

---

[10] Parker, Aaron. High School Robotics Course: The #1 Student View. Submitted to GCER 2008.

Introduction to Robotics
Desired Learning Outcomes
(* items not met during initial Spring 2008 course)

Unit 1: Introduction to Robots and Robot Building

Students will know the key historical milestones of robot development.
- The robot idea in early history
- Robots in fiction (literature and movies)
- Robots in practice

Students will know the major functions that Robots perform today.
- Robot terms and vocabulary
- Industrial and commercial robotics
- Research robotics

Student will know how to use the XBC Robot controller (computer).
- Understand safety issues and risks in using test board electronics.
- Identify all major parts of XBC
- Connect to PC, download programs, maintain battery charge
- Test motors and sensors

Students will know the basics of Team work and Project Management.

Students will build and test a ping-pong robot.

Unit 2: Sensors & Programming Part I

Students will know the function and demonstrate use of analog sensors.
- Visible light, IR “top hat”, and range finder “et “.

Students will know the function and demonstrate use of digital sensors.
- Small & large touch and lever sensors.

Students will control motors through use of back EMF program commands
- Discriminate between basic and precise movement.
- Become familiar with mathematics of angular velocity.

Students will understand the IC programming language
- Recognize syntax of the IC language.
- Know how to use variables and recognize data types.

- Use Booleans to represent logic flow.
- Know how to use conditionals and looping statements (if/then, for/next, while).
- Know the structure and procedures for functions and libraries.
- How to and why use the Preprocessor.

Students will modify IC program templates to control robot sensors and motors.

- Know procedure for designing and writing a program.
- Use commenting & documentation.
- Modify and create new variables, functions, and libraries
- Modify and create sensor and motor control procedures

Students will design, build, program & test a Robot Maze or Bot-Trot 4Bottle Race robot.

- Mounting sensors with hot glue gun.

Unit 3: Robot Structures and Mechanics

Students will know how to use a servo motor with IC

- Know the difference between a servo motor and a standard DC motor

Students will know how to design and build reliable sound structures with Lego's

- Mounting servos and motors
- Differences between connector pegs
- Lego vertical dimension relation

Students will know how to build drive trains

- Understand speed and torque tradeoffs.
- Know how gears can provide speed or torque.
- Use a worm gear.

Students will design, build, program & test a Robot that uses a shoulder and gripper arm.

Unit 4: Sensors and Programming Part II

Students will know the different vision approaches robotics researchers have used
- Know the robot "vision model"

Students will know how to use the CMU Camera and color vision options on the XBC controller
- Understand how to use the GBA color vision software
- Calibrate color models
- Understand how to work with and properties of color blobs (tracking, bounding boxes, etc.).
- Program a robot to follow a colored object

Students will know how to program all of the XBC buttons for custom use.
* Students will understand the difference between and use proportional and bang-bang robot control.
- Students will use casting (mixed data type computations) in proportional control.

Students will be familiar with multitasking in IC through use of process management functions.
* Students will be familiar with use of arrays and random numbers in IC.
Students will design, build, program & test a Robot that chases and grabs an orange ball.

Unit 5: Engineering and the Systems Engineering Process

Students will know the Systems Engineering Process
- Apply a process model that includes planning, requirements, design, prototype, test and iteration until a working system is complete.
- Identify the main focus of each engineering discipline.

Students will know how to use design reviews for refining designs and prototypes.

* Students will understand the availability of software tools to help systems engineering tasks.
- Familiar with GANTT & PERT charts for managing project tasks.
- Familiar with Modeling & Simulation as a tool to test requirements.
- Use LEGO Design Software to create designs.

Students will prepare and present a project plan for final course project.

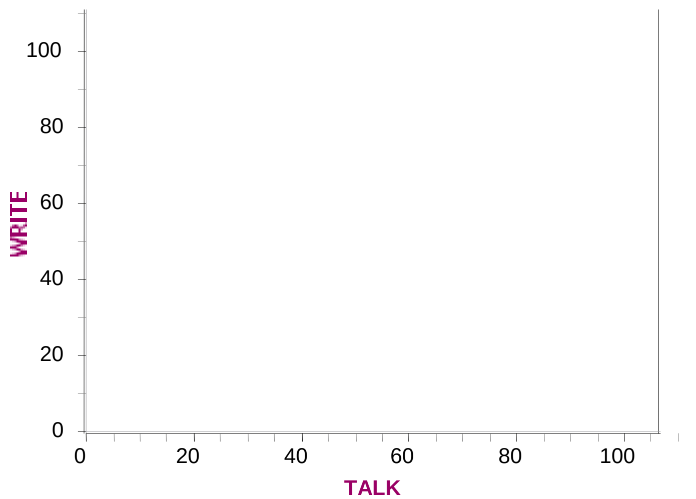
Unit 6: Course Project

Students will complete their planned robotic project, present an overview of the team effort, and demonstrate the robot.
- Project will be on the difficulty level of the current year's Botball competition.

Robotics Student Pairing Survey

NAME ___________________________

1. How do you like to communicate? Place an X (and label the coordinate) on the graph that shows the way you PREFER to communicate and how much you LIKE to communicate with people.
    a. Example 1. An X at (100, 100) means I really like to communicate with people and I don't care whether I talk or write.
    b. Example 2. An X at (10,10) means I really don't like to communicate with people and I don't care whether I talk or write.
    c. Example 3. An X at (90,10) means I really like communicating with people by talking, but I don't prefer to communicate by writing.
    d. Example 4. An X at (10,90) means I really like communicating with people by writing, but I don't prefer to communicate by talking.



2. My computer programming experience is (circle a number):

not much		some		a lot
1	2	3	4	5

   Names of computer languages you have used: ___________________________________
   ____

3. What is your last Math course and grade?
   ____

4. My experience in building with LEGO's is (circle):

not much		some		a lot
1	2	3	4	5

5. If asked, I could explain the idea of a gear ratio (circle one):

   YES MAYBE NO

Attachment 3 Table. Primary Course Resource Use Examples		
Unit 1	**KIPR**	Basic Robotics Element (#7-16) from Botball 2007 workshop
	KIPR	XBC starter-kit plan ("black" steps 1-16 & 22-24)
	CMU RE	Lego Building Tips slideshow
	NASA	Session2: Motor, sensors, pong program
Unit 2	**CMU CPE**	Interactive C Programming Tutorials
	NASA	Session 3: Analog sensors, Variables and data types, For – next loops, library functions
	CMU RE	Sensors for Robotics slideshow
	KIPR	IC programming slides from Botball 2007 workshop
	KIPR	IC Help file examples
Unit 3:	**CMU RE**	All about gears tutorial and slideshow
	KIPR	Mounting Servos and Black Gear Motors in Botball Robots (Botball 04 supplemental document)[11]
	KIPR	Calibrate Shoulder, Arm and Gripper (#187-188) from Botball 2007 workshop
Unit 4:	**KIPR**	2006-2008 Botball workshop slides for ET Sensor, servo motors, and color camera
	NASA	Session 4: XBC Buttons
Unit 5:	**Other**	*Mowin in the Graveyard,* a Systems Engineering project story (www.gmu.edu/departments/seor/insert/story/story1.html)
	Other	Systems Engineering Fundamentals, Jan 2001, DAU Press, Ft. Belvoir, VA[12]
Unit 6:	**KIPR**	2008 Botball game description and rules

[11] Martin, Fred G. The Art of Lego Design, 3/15/95, also provided as a Botball 04 supplemental document.

[12] http://www.dau.mil/pubs/pdf/SEFGuide%2001-01.pdf

Unit 3 Project:

Your robot must do at least the following:

Push the brick weight the length of the table over the end of the table.

Plus ups

1. (5 points): After pushing the brick, return to the starting box.
2. (5 points): While returning to the starting box, the robot demonstrates cliff detection using the tophat sensor.
3. (10 points) Write a one page description of your gear train that includes a diagram or sketch of the geartrain and discusses your gear ratio.

Project Grade: 100 possible points

Tasks	Max Points
Robot that completes basic project task	80
Gear train paper	10
Returns to starting box	5
Cliff detection demonstration	5

Introduction to Robotics

Semester Exam
Presentation Grading Rubric

(100 points possible, each item's points in parenthesis)

Team Names _______________________________________

(30) On Time:
Teams with Seniors:

- Presentation file due to me on or before: Wednesday, 5/21/08.
- You must be prepared to present on Thursday, 5/22/08.

Team 2 & 7:

- Presentation is due to me on or before: Wednesday, 5/28/08

(10) Length: Minimum of 10 and maximum of 20 minutes duration

(10) Presenters: All team members must discuss some portion of the semester project.

(10) Format: Must use PowerPoint and must **engage the entire class** in presentation.

(40) Content: Your presentation must include the following:

1. Design: how did my team use systems engineering concepts to come up with a design strategy for the robot?
2. Hardware: What were the key decisions that the team made related to hardware?
   For example:
   a. Why or why not use gears for motion?
   b. Arm design & structural integrity of chassis
   c. Servo/motor use
3. Software: What were the key decisions that the team made related to software?
   For example:
   a. What template did the team start with for our task?
   b. How did we collect data to test software and fix problems?
   c. How did we use a flow diagram to plan the logic of our program?
4. Sensors: Describe how sensors were used to meet the team's strategy.
5. Creativity: Use of photographs, drawings, &/or video to help explain your team's work.