Proceedings of the American Control Conference
Anchorage, AK May 8-10, 2002

Robot Soccer as a Culminating Design Project for Undergraduates

Randal W. Beard James K. Archibald Steven A. Olson
Department of Electrical and Computer Engineering
Brigham Young University
Provo, UT, 84602
{beard, jka, olsonsa}@ee.byu.edu
http://www.ee.byu.edu/ee/srprojects/robot

Abstract

For the past four years, the Electrical and Computer Engineering Department at BYU has sponsored a one-on-one robot soccer competition as one of several electives within the senior capstone design project. Teams of four students design and build autonomous robots that must fit within a seven inch cube. The soccer field is roughly the size of a ping-pong table and a golf ball is used as the soccer ball. An overhead camera provides sensory input for the robots. Much of the project involves the creation and debugging of software, ranging from image processing code to multi-level control software to allow precise movement and intelligent play. Using a custom simulator of the game environment, software is developed and tested in parallel with the creation of the robot hardware. This paper describes the complete program and explains how it has been tailored to become an exceptional capstone project for senior electrical and computer engineering students.

1 Introduction

The ABET accreditation criteria requires that all students be prepared through a curriculum culminating in a "major design experience;" Moreover, each student is required to demonstrate an ability to "function on multi-disciplinary teams" [1]. Good culminating design projects for teams of students can be difficult to develop. We have observed three different approaches to senior design projects. The first is to require students to define their own project, often with little guidance from faculty advisors. There are several problems with this approach. The projects are rarely scoped appropriately: they are either too easy and do not stretch the students, or they are too difficult to complete in the allotted time and with the available resources. Also, the projects seldom have a strong connection to the research interests and ongoing projects of the assigned faculty member. Faculty advisors are therefore more likely to view their involvement with the design project as a distraction and less likely to be effective mentors.

The second observed approach to the culminating design experience is to solicit projects from industrial sponsors. The primary difficulty with this approach is that soliciting industrial sponsors and following through on each of the projects requires an enormous investment of faculty resources. In addition, industrial projects are difficult to scope appropriately.

The third senior project approach is to enhance the labs in senior-level courses already offered by the department. This is probably the least-cost solution for faculty members. However, it seems to fall short of the objective of a culminating design experience.

During the past four years we have been experimenting with a novel culminating design project for undergraduate students based on an autonomous robot soccer competition. We have been motivated by the desire to provide an exceptional, well-scoped, multi-disciplinary design experience for our students that not only satisfies the ABET requirements for a culminating design experience and an ability to function on multi-disciplinary teams, but also prepares them to make an immediate impact in our own areas of research. These include control theory, multiple agent coordination, software architectures, and real-time computing. In addition, we have sought to establish an infrastructure that requires modest faculty resources to maintain the program from year-to-year.

Our solution has evolved into a one semester, four credit hour design course culminating in a one-on-one robot soccer competition held in a public venue. In order to design a robot that can successfully compete in robot soccer, students need to become at least somewhat familiar with a number of technical topics, including: mechanical design, microcontroller programming and interface, RF communication, signal conditioning, communication protocol development, software architectures, concurrent engineering, artificial intelligence, computer vision, system modeling, low-level control algorithms, path planning, trajectory generation and following, ball prediction, and state estimation from noisy, time-delayed signals. In other words, the project requires integration across a broad range of the Electrical and Computer Engineering discipline and therefore represents a truly *culminating* design experience. One of the advantages of robot soccer is that the design objectives are transparent, yet the design alternatives are innumerable. Another advantage is that the project captures the imagination of students and spectators alike and creates a great deal of enthusiasm.

There are a number of research groups around the country engaged in robot soccer, but we are unaware of any other university that offers a robot soccer design project for undergraduates. The biggest challenge faced in an open-ended project of this type is restricting its scope to the point that teams of typical students can complete it in a single semester, while not overly restricting the design space to the point that student creativity is stifled. The main objective of this paper is to describe how we have addressed this challenge.

In Sections 2 and 3 we describe our current course format and structure that brings students to the point that they can explore many design alternatives with confidence in their ability to assess the implementation difficulty of each. (In four years of implementation we have only had one team of 26 total that failed to produce a robot that could compete in the tournament.) In Section 4 we describe the rules and format of the competition. In order to scope the project for undergraduate students, we have invented our own competition rules rather than use the existing RoboCup [2] or FIRA [3] standards. In Section 5 we itemize the the hardware needed for the course and tabulate cost estimates of required equipment. In Section 6 we conclude with a discussion of the impact of this senior design program and of our plans for the future.

2 Course Structure and Philosophy

Robot soccer is offered as a one semester, four credit hour course, to be taken by electrical and computer engineering students in the senior year. The goal of the course is to introduce students to the "art of design" and then to have them practice those principles with a significant design project. The schedule for the course is shown in Figure 1.

As Figure 1 shows, the students meet four times a week during the first four weeks of the course. Two lectures a week are devoted to "business processes," and two lectures a week are devoted to technical lectures. The titles of the business process lectures are

Week	M	W	Th	F
1	BP 1	BP 2	TLL 1	THL1
2	BP 3	BP 4	TLL 2	THL2
3	BP 5	BP 6	TLL 3	THL3
4	BP 7	BP 8	TLL 4	
5				PC 1
6				DR 1
7				
8				
9				PC 2
10				DR 2
11				
12				
13				PC 3
14				Tournament
15				

Figure 1: Course schedule for robot soccer.

BP 1. Working in teams.

BP 2. Capturing the voice of the customer.

BP 3. Turning customer needs into product specifications.

BP 4. Product architecture and concept generation.

BP 5. Developing project schedules.

BP 6. Giving effective presentations.

BP 7. Overview of business processes.

BP 8. Engineering economics.

The goal of the business process lectures is to teach students a systematic design process that includes an evaluation of customer needs, product specification, and the design-test-debug cycle. These eight lectures are presented by an adjunct faculty member who has worked in industry for many years.

The seven technical lectures are organized around two concurrent tracks: a low-level track (TLL), and a high-level track (THL). The lecture titles for the low-level track are

TLL 1. Robot construction and design.

TLL 2. Computer vision and velocity estimation.

TLL 3. Low level control.

TLL 4. Motion planning.

The lecture titles for the high-level track are

THL 1. Simulator and software architecture.

THL 2. State machines and play construction.

THL 3. Artificial intelligence.

Students are organized into teams of four students per team, with two members of the team primarily responsible for "low-level" technology development which includes hardware, vision, and control algorithms, and two members of the team primarily responsible for high-level technology development which includes the software architecture and artificial intelligence. The course is organized such that during the first month of the course students design and build a prototype robot with prototype software. The mechanical structure of the robot is specified and particular software functionality is required. Toward that end, each of the technical lectures listed above has an associated laboratory assignment. After completing all of the lab assignments, each team will have built and programmed a robot that has nominal functionality.

During the next ten weeks of the course, the students are free to design their own robot, redesign the software architecture, and add functionality. The class does not meet together during these ten weeks, but two scheduled design reviews (DR 1 and DR 2) are held with each team individually. These design reviews allow us to assess the progress of each team and to offer suggestions and encouragement. In addition to design reviews we have three practice competitions (PC 1, PC 2, and PC 3). The final double-elimination tournament is held during the fourteenth week of the semester. The practice competitions allow students to gauge their progress relative to their peers and also give them additional incentive.

As reflected in Figure 1, the philosophy of the course is to lead the students through a prototype design during the first month of the course and then to "turn them loose" to be creative. The development of the prototype greatly increases the likelihood of success of each team, yet in our four years of experience not a single team has stuck with the original prototype for their final design. Every team elected to design and build a custom robot with its own unique features.

3 Labs and Lectures

Developing a prototype robot in one month is a nontrivial exercise. Over the years we have refined our lectures and labs to focus on the essential and the truly useful. We have strived to achieve a reasonable balance between exposing students to a variety of interesting material and the reality that this project is just one of several courses students are taking. In this section we describe the specific lectures and lab requirements.

3.1 TLL 1. Robot Construction and Design

The objective of the first technical low-level lab is to have the students build a prototype robot that can be remotely controlled from a computer keyboard. A photo of the prototype robot is shown in Figure 2.

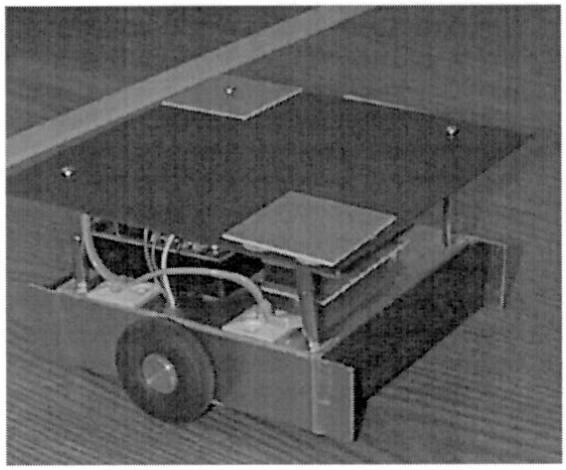

Figure 2: Prototype robot build in first low-level lab.

To complete the lab, the students must integrate the RF link, the microcontroller, and the DC motors. Unoptimized prototype software is provided to the students to enable them to get a basic system working. Most teams begin to optimize that software right from the outset.

Although, as noted earlier, the simple prototype robot is rarely used past the initial labs and practice competition, the experience gained in its creation is invaluable. The process of building the robot gives the team many ideas for improvement. Students become familiar with departmental resources which gives them the confidence to pursue more intricate and complex designs.

3.2 THL 1. Simulator and Software Architecture

This lab introduces students to a custom simulator developed to facilitate the development and testing of each team's control software. Written in C++, the simulator runs on the LINUX computers used in the class laboratory. It includes a graphical user interface displaying an overhead view of the playing field, and it models the movement and interaction of robots and ball. Based on a client-server model, the simulator communicates with each robot's software via sockets, allowing teams to play each other without having to recompile or make their source code available. Figure 3 shows how the simulator and the hardware robot are designed to have identical interfaces, allowing the same control software to run on both without modification. The figure also illustrates the software architecture introduced in this lab and developed in subsequent labs.

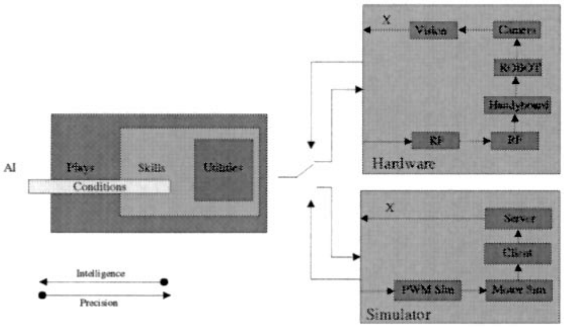

Figure 3: Software architecture.

For this lab students are required to familiarize themselves with the simulator and its operation by writing code to make their robot score on a simple play given a fixed starting position and an essentially immobile opponent. Students are given a collection of unoptimized low-level control functions (object code only) so they can focus on higher-level operations.

3.3 TLL 2. Computer Vision

The objective of the second low-level technical lab is to familiarize students with basics of computer vision and to help them develop a base-line vision system. As part of the lab, students are required to develop vision code that returns the position and orientation of their prototype robot, the position and orientation of an opponent, and the ball's position. Students are also required to estimate the variance of these measurements as well as the velocities and angular velocities.

As part of the lab, students are given a prototype vision system that achieves approximately 10 frames/sec. In addition, they are given software that helps them to visually threshold the colors on the field. With some effort, most teams are able to modify the prototype code to achieve sample rates of 30 frames/sec.

3.4 THL 2. Play Construction

In this lab, teams are required to make significant progress in the creation of their high-level intelligence code. In a six-game match on the simulator, they must beat a relatively unskilled opponent we provide. The software architecture shown in Figure 3 is discussed and recommended, but teams are free to choose any approach that is extensible and consistent.

In the suggested architecture, a robot's repertoire of simple actions are represented by *skills* or low-level control functions. Students are required to create at least four *plays*, or complex actions consisting of some sequence of skills. Students are given code templates for plays as a starting point. At the highest level, the *AI* or *artificial intelligence* selects a reasonable play for the current game situation. For this lab, a very simple AI that picks one of the available four plays is all that is required. (*Conditions* and *utilities*, the remaining classes of functions in the architecture, exist primarily to simplify the other function types by encapsulating state and implementation details.)

3.5 TLL 3. Low Level Control

The objective of the third low-level technical lab is to develop low-level control functionality. In the software architecture shown in Figure 3, motion commands to the robot are issued at the "skill" level. Skills developed in this lab include the following:

`sklMoveWheels(wrd,wld)`: This skill moves the robot wheels at angular speed `wrd` for the right wheel and `wld` for the left wheel.

`sklMove(v,w)`: This skill moves the robot at linear speed `v` and angular speed `w`.

`sklGoToPoint(p, ang)`: This skill moves the robot to the point `p`, pointing along the angle specified by `ang`.

`sklTurnToAngle(ang)`: This skill causes the robot to turn to angle `ang`.

3.6 THL 3. Artificial Intelligence

In this lab students are required to refine their software until they are able to beat a much more capable client than in the previous high-level lab. They are required to implement a more complete set of plays and a well thought-out and consistent AI mechanism. Students generally choose either a *reactive* approach in which robot actions are tightly coupled to stimuli, a *deliberative* approach employing planning based on current and projected state information, or a *hybrid* approach employing both reactive and deliberative aspects.

3.7 TLL 4. Motion Planning

The objective of the fourth low-level technical lab is to explore a few techniques for motion planning and control. In particular we discuss waypoint path planning, which can be used in conjunction with the deliberative AI approach, and we discuss the potential fields method which can be used together with a reactive AI approach.

4 Competition Format

The competition formation is a one-on-one competition loosely based on MLS shootouts. The soccer field, which is shown in Figure 4, is 60 inches by 108 inches, roughly the size of a ping-pong table. The field is green with white markings. A yellow golf ball is used as the soccer ball.

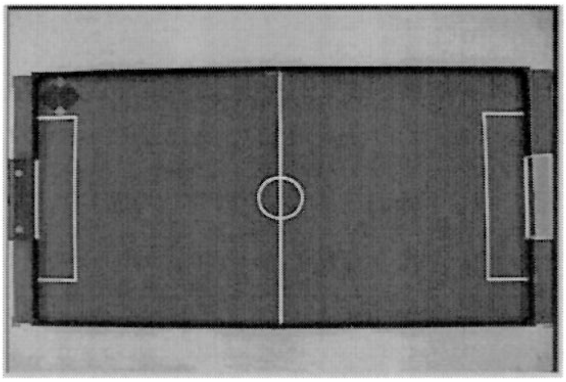

Figure 4: Robot soccer field at BYU.

Each team is supplied with a dedicated Linux workstation equipped with a frame grabber that receives image data from a camera suspended at a fixed height over the playing field. A single camera is used and the signal is distributed to all computers.

With the exception of special colored markers on the top of each robot, everything that is visible to the overhead camera must be black or dark gray. Square colored markers (provided to each team) are used on the top of each robot to distinguish between players and to allow the orientation to be determined. Each robot has a green marker in a fixed location. On offense, each team must affix a red marker; defensive players have a blue marker. The markers atop each robot must be visible at all times from the overhead camera, and they must be non-overlapping.

Each robot must fit in its entirety into a 7 inch cube. At no time during competition can a kicker or other movable part extend beyond this 7 inch limit.

The final double-elimination tournament consists of a sequence of matches, each a contest between two robots. Each match consists of three "innings", and each inning consists of two "plays". For each play, one robot is designated as offensive and the other as defensive. For the second play of each inning, the roles are reversed, with the corresponding "uniform" change. Thus, in every match, each player has three opportunities on offense and three on defense.

Each play begins with the offensive team placing its robot anywhere on its own half of the field outside the center circle. Once the offensive robot is in place, the defensive team places the ball anywhere in the center circle, and then it places the defensive robot so that at least part of it is in its goalie box. Once the robots and ball are in place, play begins when the referee says "go". This may be communicated to the robot by a key-press on the workstation keyboard, after which human interaction with the robot is prohibited.

To keep the tournament spectator-friendly, strict timing regulations are observed. Each play ends when a goal is scored or when 30 seconds has elapsed, whichever comes first. At the end of each play, the team on offense for the next play has 30 seconds to change uniform and place their robot. The defensive team then has 15 seconds to place the ball and their robot. Therefore, throughout an entire 6-play match, plays are separated by intervals that are 45 seconds or shorter. From the conclusion of any match, the participants in the next match have 3 minutes to set up.

Both offensive and defensive players may score. The offensive and defensive labels apply only to initial placement; there are no movement restrictions once play begins. The winner of each match is the robot with the most goals scored over the 6 plays. In the event of a tie, the match is extended with sudden-death overtime innings.

Robots are not allowed to "carry" the ball, defined as trapping or holding it in such a way that it cannot be taken away by the opponent. Robots may not have concave openings in the perimeter that allow unfair control of the ball; at least one-half the ball's diameter must be visible from all side views at all times. Similarly, at least one half the ball's diameter must be visible at all times from a view directly overhead the side of the robot in contact with the ball.

5 Equipment

The equipment supplied to the students and its approximate cost is listed in Table 1.

Equipment	No./Team	Cost/Item
DC motor	2	$150
Handyboard	1	$300
RF link	1	$300
Wheels	2	$2
Markers	1	$1
Framegrabber	1	$410
Total/Team		$1,315

Table 1: Equipment list and approximate cost.

In addition to the items shown in Table 1, each team is supplied a dedicated LINUX workstation with standard software and compilers. Students also have access to the machine shop in BYU's College of Engineering, where they can machine parts for their robots. The overhead camera is a low-cost camcorder (approximately $500).

6 Conclusion and Future Direction

In this paper we have described the current implementation of a robot soccer design course for undergraduate students. We feel that the course has been very successful. It receives enthusiastic support from students, faculty, and industrial partners. The level of play and the enthusiasm of the competition can be seen in videos that can be downloaded from our project website. We invite other universities to become involved and would welcome inter-collegiate competitions for our students.

The success of the senior project course has led to changes elsewhere in our curriculum. For example, in the past year we have reorganized our classical control course so that the labs focus on mobile robots. To do so, topics such as feedback linearization, path planning, and trajectory generation were added to the course. It is particularly noteworthy that the enrollment in this control class has approximately doubled since the introduction of the robot soccer course. (Students who want to take the course in the future pay careful attention to the suggested prerequisites; most of the increased enrollment in the control class comes from computer engineering majors.) We conclude that robot soccer is an excellent introduction to control theory that underscores the relevancy of the subject matter.

We are committed to continuing enhancements and refinements that improve our course. When possible we upgrade to more reliable hardware components and subsystems. We have reduced the overall robot size three times, from an initial 12 inches to the current 7 inches, and we may reduce it further in the future. More significantly, when the playing capabilities of the robots come close to their full potential for one-on-one competition, we plan to move to a two-on-two format, with a commensurate increase in the complexity of the higher levels of control software. A project of this type would expose more students to the problems in our own research area of coordinated control of multiple vehicles. We think the experience will make more of our students interested in our research projects and in pursing graduate study.

References

[1] http://www.abet.org/.

[2] "The robot world cup initiative." http://www.robocup.org.

[3] "Federation of international robot-soccer association (fira)." http://www.fira.net.