AUTONOMOUS SOCCER-PLAYING ROBOTS:
A SENIOR DESIGN PROJECT

Jed M. Kelsey
Department of Electrical & Computer Engineering
Brigham Young University

Faculty Advisor:
James K. Archibald

ABSTRACT

This paper describes the experiences and final design of one team in a senior design competition to build a soccer-playing robot. Each robot was required to operate autonomously under the remote control of a dedicated host computer via a wireless link. Each team designed and constructed a robot and wrote its control software. Certain components were made available to all teams. These components included wireless transmitters and receivers, microcontrollers, overhead cameras, image processing boards, and desktop computers. This paper describes the team's hardware and software designs, problems they encountered, and lessons learned.

KEY WORDS

Robot soccer, Senior Design Project, Autonomous Control, Nonholonomic Robots

INTRODUCTION

During the 1998-1999 school year, the Brigham Young University Department of Electrical and Computer Engineering sponsored a senior project program involving twelve students (four teams of three students each) in designing and building one-on-one soccer-playing robots [1]. The project was part of a pilot program initiated and advised by Dr. James K. Archibald and Dr. Randal W. Beard. The purpose of the pilot program was to implement a framework to administer various types of senior projects, meeting ABET 2000 requirements and providing a team-based multidisciplinary design experience to graduating seniors. A number of similar robot design projects have been used successfully at other universities to provide this multidisciplinary team-based approach (for example, see [5] and [6]). The BYU program was intended to reinforce important skills not normally emphasized in the curriculum, as well as polish technical, presentation, and writing skills.

One of the key objectives of a university senior design project is to emulate the design process and experience that engineers will experience once they leave the university. The design teams involved in the Robot Soccer project worked independently to create the soccer-playing robots, and a competition was held between the robots at the end of the final semester.

DESIGN CONSTRAINTS AND PROJECT FRAMEWORK

One focus of the senior project program was to require the teams to gather their own body of facts and make design decisions based on their findings. The four teams were essentially given free rein in the design, although a rigorous top-down approach to the design was recommended. A framework of initial design criteria and rules was developed, based on the RoboCup Small-Size Robot League regulations with a few modifications [2]. Robots were required to fit within a 10"x10"x10" cube, allowing somewhat larger robots than in the RoboCup small-robot competition.

Throughout the first of two semesters, a class was held once each week in which students were able to discuss various topics related to robot design. These topics included project management, RF communications, vision processing, construction materials, motor dynamics, and related items of interest. Design reviews were held at the end of the first semester to evaluate the teams’ progress and to give students the opportunity to present their final design proposal. Design reviews and checkpoints were held several times during the second semester.

To assist the students in the top-down design of their robots, several control documents were required at the design reviews. These documents included Functional Specification, Concept Design & Evaluation, and Product Features & Specifications documents, as well as project schedules.

A final competition was scheduled for the end of the second semester, at which time the robots competed against each of the other teams' robots one-on-one. The assistance of the International Telemetering Foundation was instrumental in the success of the project, and a cash prize incentive of $1000 to be shared among the designers of the winning robot was provided by ITF.

AVAILABLE RESOURCES

Each team was equipped with a Motorola[*] 68HC11 microcontroller on MIT's Handy Board, Linx RF transmitters and receivers with demo prototyping boards, and access to computers in the laboratory with Microsoft Visual C++ and GNU gcc compilers. In

---

[*] Motorola is a registered trademark of Motorola, Inc.; Linx is a registered trademark of Linx Technologies, Inc.; Microsoft is a registered trademark of Microsoft, Inc.; Cognachrome is a registered trademark of Newton Research Labs.

addition, two CCD cameras and two host computers (each with a dual-boot Linux/Windows 95 setup) were set up in the lab. Cognachrome Vision System image processing boards from Newton Laboratories were provided for each camera.

The overhead CCD camera and vision processor board were used to determine the positions of objects on the playing field. The camera transmitted an NTSC video signal to the vision board, which could track objects of particular colors on three separate channels concurrently at data rates of up to 60 frames per second. This data could be used to track the locations and trajectories of a robot, its opponent, and the "soccer ball" (a colored golf ball).

Although the above items were available to every team, individual teams could opt to obtain and use sensors and other devices for tracking and navigation. Any other materials needed for the construction of the robots were obtained as needed. The students were limited to a fixed out-of-pocket expense, but department funds were available for materials which could be reused for future robots.

The playing field was built by department technicians according to the RoboCup Small-Robot League regulations. The basic setup of the playing field and camera system is shown in Figure 1.

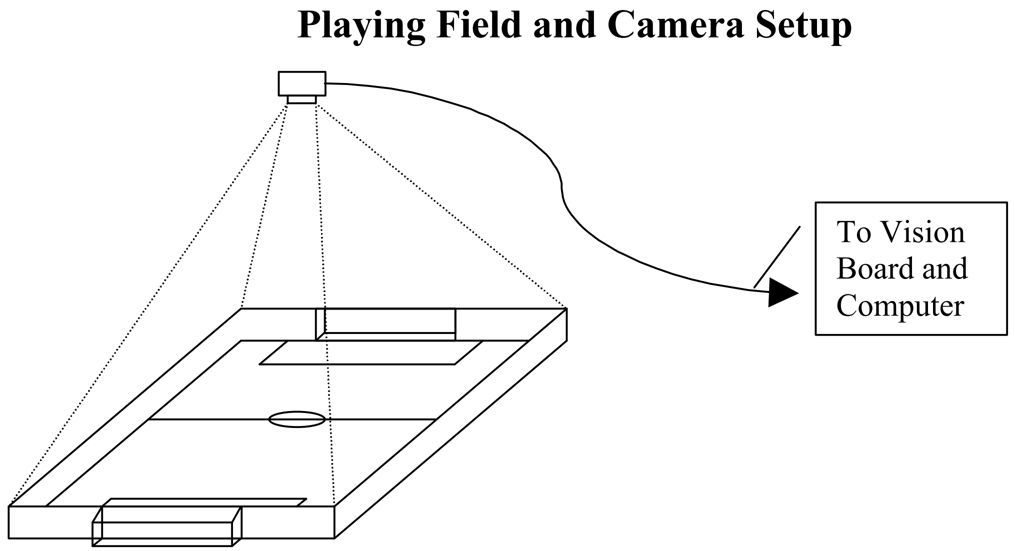


Figure 1.

SPARCC’S DESIGN PROCESS

SPARCC, whose name is derived from "Soccer-Playing Autonomous Robot with Camera Control," was one entry in the BYU robot soccer competition [3]. SPARCC’s designers—Jed Kelsey, Ron Ward, and Jonathan Waite—were all seniors graduating at the end of Winter semester 1999. Each brought a different background and set of skills to the project.

SPARCC’s design was completed in several stages. First its basic physical structure and functionality was designed, followed by the software algorithms. Both the hardware and software design processes proved to be iterative in nature. Regular design reviews helped refine and solidify initial design goals and target specifications. As the design matured, many of the original design goals were met, while others were discarded as being unneeded or no longer relevant. A working prototype was built in January 1999, and software for the host computer and embedded microcontroller was written later that semester.

The images in Figure 2 below show SPARCC’s physical construction, and Figure 3 describes the system data flow.

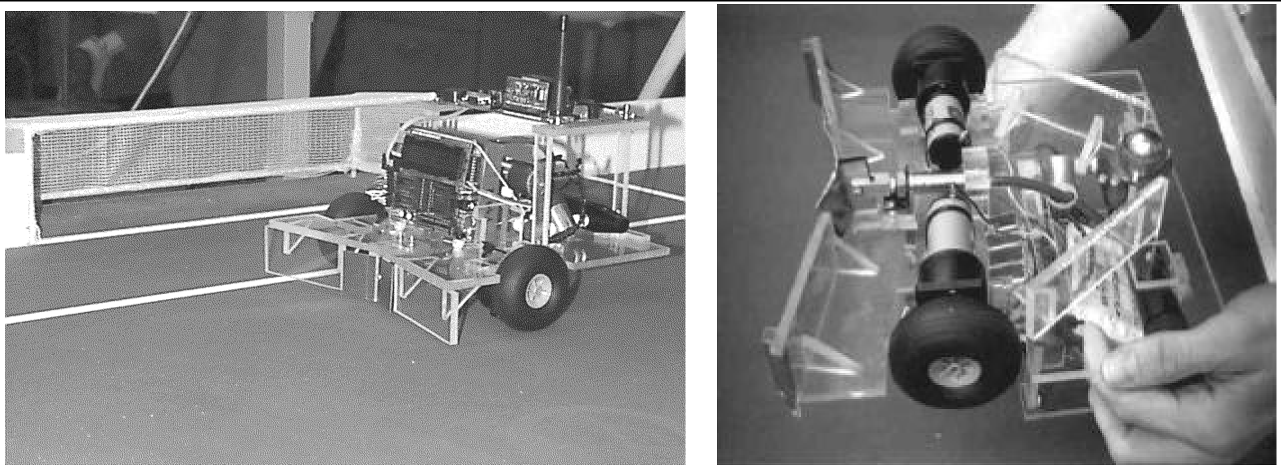


Figure 2.

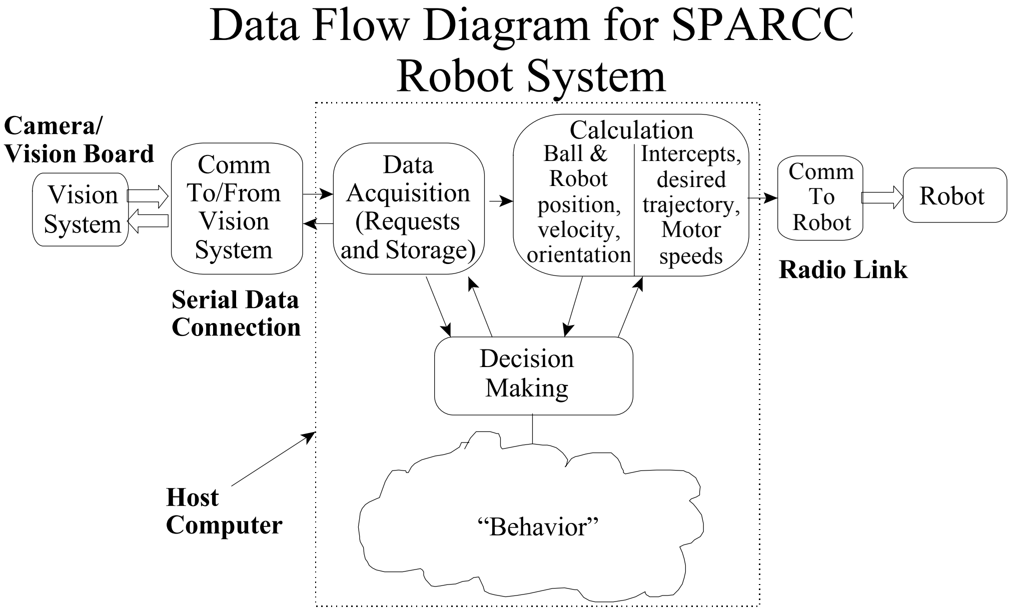


Figure 3.

HARDWARE

Because of its strength, unique appearance, and workability, Plexiglas was selected as the primary building material for the prototype, while aluminum was initially considered as material for the final robot. However, upon more research and testing of the Linx RF transmitter/receiver pair, it was discovered that proximity to flat sheets of a conductive material could cause communications disruption due to shielding and destructive interference. For this reason, the Plexiglas model was chosen for the final design.

The V-shaped front end of the robot was designed to help "funnel" the ball into kicking position. In addition, the V shape prevents the ball from sliding off of the front of the robot when pushing it. The depth of the V was limited by the requirement that at least half the ball must be visible from the side; the robot could not "trap" the ball and would lose it with a sudden turn. All of the teams utilized this basic V shape for the front end of their robots.

SPARCC's designers felt that the ability to kick the ball from across the field was an important factor in gaining an advantage over opponents in a soccer game. A feature unique to SPARCC was the pneumatic ram-operated kicker. A standard, inexpensive CO_2 cartridge such as those used in paintball guns was utilized to provide the thrust needed for kicking the ball. Testing showed that typically around 20 kicks were possible per cartridge.

The pneumatic kicker was chosen in preference to other considered options such as solenoids or servo motors. The CO_2 system was fairly lightweight and consumed very little battery power, an advantage over both of the other two options considered. In addition, it was easy to replace the cartridge. At the end of the final semester, SPARCC was the only robot with a kicker. The other teams opted to push the ball around with their robot rather than "kick" it.

The robot was designed with a differential drive, permitting a near-zero turn radius. High-quality geared Maxon motors were used to drive the robot. Camera feedback, rather than encoder data, was used to ensure straight travel by the robot. To prevent undesired overshoot by the proportional control and collisions with the walls or other robots, "brakes" were added to the robot. In essence, this was done by using relay contacts to disconnect the motor drivers and provide a short circuit across the motors. The back-emf of the spinning drive motors then quickly stopped the robot. Of the four robots in the competition, SPARCC was the only robot designed with brakes.

The overhead camera and vision processor board were used to gather global positioning data of SPARCC, its opponent, and the soccer ball. The data was then corrected for the camera's fisheye effect by SPARCC's software interface on the host computer, which processed the information and made control decisions for the robot.

Use of sensors in SPARCC was limited to this overhead camera. A Linx RF transmitter/receiver pair was used to implement a one-way link from the host computer to the robot. The one-way nature of the communication link effectively prohibited the robot from providing sensor feedback to the host computer. However, the camera provided updates at a high enough rate--and with sufficient accuracy--to ensure that additional onboard sensors, such as wheel encoders and microswitches, were not required for operation.

Using the Linx transmitter and receiver on the prototyping boards enabled one-way serial communications from the host computer to the robot with very few hardware modifications to the prototype boards. Upon testing the Linx boards, it was discovered that some characters were incorrectly transmitted if a header character was not transmitted immediately preceding it. However, without using interrupt-driven serial communications for the robot's microcontroller (on the Handy Board), the second character would be dropped if the header character was not removed from the serial input buffer soon enough.

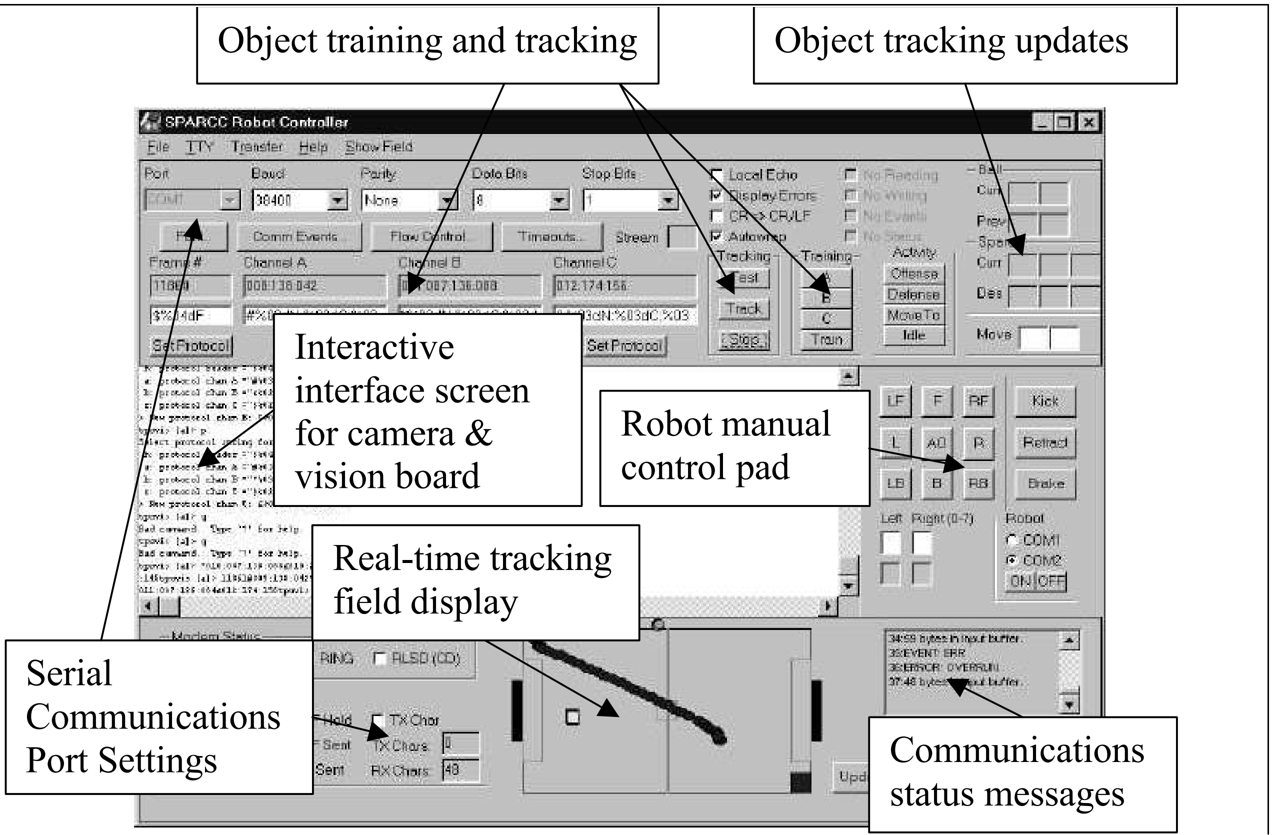
Due to lack of time, the interrupt system was not implemented. Instead, a simple protocol for transmitting information from the host computer to the robot was developed as a tradeoff between the Linx board's problems with transmitting single characters and the Handy Board's tendency to drop characters when simply polling the serial input buffer.

This tradeoff meant that certain characters were not used, as they were not transmitted correctly by the Linx transmitter without an appropriate header sequence. Using this protocol, all required control information for the robot could be transmitted in a single 8-bit character. A simple repeated transfer of the same character was used to detect transmission errors.

SOFTWARE

Each team had to make decisions as to whether major data processing and computations were to be done on their robot's microcontroller or on a host computer. In SPARCC's case, it was decided that all video processing and decision-making were to be done on the camera's vision board and on the host computer. The program executing on the robot's microprocessor simply handled data reception on its serial line and controlled motor speeds and kicking actions. Thus the program onboard the robot was a simple one which polled the serial buffer for new commands and executed them immediately.

The control software running on the host computer was entirely coded in C, and compiled with Microsoft Visual C++. The user interface to the control program is shown in Figure 4.



Baud: 38400
Parity: None
Data Bits: 8
Stop Bits: 1
☐ Local Echo
☑ Display Errors
☐ CR => CR/LF
☑ Autowrap
Channel A: 008:138:042
Channel C: 012:174:156
Frame # protocol: $%04dF
○ COM1
● COM2
TX Chars: 0
RX Chars: 48

```
34:59 bytes in input buffer.
35:EVENT: ERR
36:ERROR: OVERRUN
37:48 bytes in input buffer.
```

Figure 4.

The host computer program used a serial data connection with the vision processor board to download camera frame grab images and object tracking data. Tracked objects were displayed in a graphical user interface and were used to make control decisions based on selected behavior modes.

The host computer's control interface shown in Figure 4 was built as a multithreaded application. The graphical user interface and datapoint conditioning routines (for removal of camera fisheye effect and noise) constituted the main thread, while serial communication with the vision board had separate threads for reading and writing. Calculations, decision-making, and the robot's proportional control all worked within another thread of execution, as did the object tracking display.

The interface was designed to function at several levels of abstraction. At the lowest level, the robot could be controlled manually by setting the desired motor speeds and clicking on the control pad at the right to make the robot move, brake, or kick. In addition, the software allowed manual or automatic control of the camera and vision board. Behavior schemes were selected at the start of each round, and included offense, defense, and field positioning schemes.

Serial port settings for the camera system interface could be controlled manually for interfacing with different vision processor boards or at different speeds. By default, the settings for Newton Labs' vision processor board were selected. Serial communications status messages for buffer overruns and other problems were displayed in a separate box. The serial interface portion of the control application was based on a multithreaded TTY communications example program distributed with Microsoft Visual C++.

Communications protocols for the vision system were built in to the interface and could be modified onscreen by the user. Each of the three object-tracking channels was individually trained to a desired color by pressing the appropriate button on the interface screen. Two tracking modes could be selected: a test mode in which object tracking information was displayed textually in the interactive interface screen, and a tracking mode for real-time tracking and control of the robot. Tracked objects were displayed in real-time in the field display at the bottom of the screen.

Once tracking was initiated, the application utilized a simple hybrid proportional control to maneuver the nonholonomic robot around the field. The control featured decoupled angular and translational proportional control during travel to minimize angle and distance errors. Behavior was based on distinct states or modes of operation, depending on the selected behavior scheme. Modes included stationary orientation, braking, and linear movement. Switching between these modes of operation was based upon SPARCC's proximity and relative positioning to the walls, opponents, and the ball.

PROBLEMS ENCOUNTERED

One of the most difficult challenges our team faced was inconsistency in color training and tracking with the vision system. The vision system did not work as well as hoped without near-perfectly uniform lighting conditions (which are difficult to maintain), and it had problems differentiating between and tracking certain colors. Camera parameter variations, nonuniform lighting conditions, and limitations in the vision board's color lookup table caused the board to lose track of certain objects in portions of the field—especially the corners. With more time, a better lighting scheme could be implemented to help eliminate these "blind spots."

Another challenge we faced was with the late arrival of much of the hardware to be used in the project (particularly the camera system.) With more time to work with the actual hardware, implementation would have been completed more quickly. In addition, beginning design and implementation of software control algorithms earlier would have helped us greatly. We also ran into problems with memory leaks in the initial design of the multitasking host computer program, which set us back somewhat in our project schedule. It would have helped to start software design earlier in order to relax time constraints due to software problems.

Given one or two more months, we would have finished implementing the camera frame grab functionality. This would enable us to automatically generate a lookup table for camera fisheye correction. In addition, we could have expanded and improved our behavior-defining algorithms and improved the interrupt-driven buffered serial communications by writing a better interrupt service routine for the robot microcontroller's serial port.

Interrupt-driven serial communication on the robot would enable multi-character or binary stream transmissions, and permit the use of a larger command set. In addition, it would allow faster transmission of data on the serial line. We spent a large portion of time testing and adjusting the Linx transmitter and receiver, and multi-character transmission packets seemed to be more reliable and less prone to noise. If interrupt-driven communications had been established, we could have transmitted data more efficiently and quickly.

Control logic, behaviors, and the hybrid proportional control logic would have been greatly improved if we had a little more time to spend testing and debugging. Two-way communication between the robot and the computer would be a great asset, since it would allow more sophisticated error checking and feedback from additional sensors. Encoders on the wheels would further stabilize the robot's movement, and an infrared phototransmitter/receptor would help us to determine with better accuracy when the ball was in the appropriate position for kicking.

RESULTS AND CONCLUSIONS

The Robot Soccer senior project was a great introduction to the field of robotics and autonomous control. The project incorporated many different fields of expertise and experience within the design teams and demonstrated the importance of teamwork and top-down design methodology. It also emphasized the importance of efficient, reliable data collection and the need for precision and efficiency in communication, both between system components and system designers.

Design of the SPARCC's hardware took less than 30% of the total design time required for the project. Most of the remainder of design time was spent programming the host computer's graphical user interface and developing supportive control behavior, and in testing and tuning the program. Although SPARCC did not win the competition ending this year's Robot Soccer project, it was very competitive.

The breadth of experience gained designing, testing, and running SPARCC in competition with the other teams' robots was invaluable to each of the designers, as were the insights gained from the other teams' ideas and systems. In addition, the experience gained from developing SPARCC should prove invaluable in the author's graduate research work in BYU's Multi-AGent Intelligent Coordinated Control (MAGICC) laboratory [4]. He will also be working as a teaching assistant for next year's robot soccer senior project program.

The team-based, structured design methodology required for this senior project was essential for SPARCC's successful completion. The faculty's emphasis on control documents was a great help in focusing our efforts and solidifying design concepts. Design tasks and responsibilities were divided among the three members of SPARCC's design team, and regular design reviews helped keep us on track. Classes during the first semester were very informative, although much of the lecture material was theoretic in nature and not directly applicable to the project.

Based on the responses of the faculty, SPARCC's design team members, and the other students involved in this year's robot soccer senior project, the program was successful in providing the students with an excellent multidisciplinary, team-based engineering experience. The senior project was very time-consuming, with most students working well over the 100 hours required for senior projects. Most of the groups found they were running short on time at they end of the final semester. Subsequent senior projects will be based on a more aggressive schedule to assist students in meeting design goals earlier in the project timeline. In addition, completion of the project will be worth more than the 1 credit hour which was given to students completing this year's project.

REFERENCES

Informative Links:

[1] 1998-1999 Robot Soccer Senior Project (BYU Electrical & Computer Engineering)
http://www.ee.byu.edu/~beard/senior-projects/1998-1999/description.html

[2] RoboCup Information and Small-Size Robot League Regulations
http://www.robocup.org/
http://www.robocup.org/regulations/42.html

[3] SPARCC Soccer-Playing Robot (BYU Electrical & Computer Engineering)
http://www.ee.byu.edu/~beard/senior-projects/1998-1999/sparc/index.html

[4] BYU MAGICC Laboratory (Multi-AGent Intelligent Coordinated Control)
http://www.ee.byu.edu/ee/robotics/

References:

[5] Crisman, Jill D., "System Design Via Small Mobile Robots," IEEE Transactions on Education, Vol. 39, No. 2, IEEE, Piscataway, NJ, May 1996, pp. 275-280.

[6] Kruempelstaedter, Kerry, "Robocar: Unmanned Ground Robotics," Linux Journal, Issue 41, Seattle, WA, September 1997, pp. 16-22.