Proceedings of the 12th
KACC, October 1997

중앙집중 제어에 근거한 마이크로 축구경기 로봇

Micro Soccer-Playing Robot Based on the Centralized Approach

°홍선기*, 엄태덕*, 이춘영*, 김민성*, M.Sugisaka**, 이주장*

*Department of Electrical Engineering, KAIST 373-1 Kusong-dong, Yusong-gu, Taejon 305-701, KOREA
TEL:+82-42-869-3432 E-mail : jjlee@ee.kaist.ac.kr
**Department of Electrical Engineering, Oita University 700 Dannoharu, Oita 870-11, JAPAN
TEL: +81-975-54-7831

Abstracts This paper presents the design procedure for soccer-playing robots based on the centralized approach. Using a fast vision system, we obtain the configuration of each robot and then the host computer computes the desired motion and commands each robot directly via RF communication. The robot soccer game has a lot of problems such as obstacle avoidance, coordination between robots, dribbling the ball, and so on. To implement such motions, we think that the centralized approach seems to be more powerful than the distributed approach. We describe the technical tips for developing the robots in detail here and explain our strategy for getting the scores.

Keywords Micro Mobile Robot, Soccer Robot, RF communication, Multi-Agent Control

1. Introduction

A robot soccer game has been issued in the field of multi-agent systems since the first MIROSOT[1] (Micro-Robot World Cup Soccer Tournament) was held in 1996. A robot soccer game has a lot of challenging problems such as coordination between robots, motion planning of robots, visual recognition of objects, and so on. The orchestration of such functions is the key factor in winning the game.

In [1], the authors define three control methods of soccer-playing robots: a remote-brainless control, a vision-based control, and a robot-based control. Such categorization of control structure depends if each robot has its own CPU and who determines the robot's motion based on the visual information, as shown in Table 1. The designer may choose one method according to his/her research interest.

There has been a controversial discussion about which control method is more proper for a robot soccer game between a centralized control and a distributed one [2]. Although the robot-based control (i.e. the distributed control) is a very interesting field of research, we decide to use the remote-brainless control structure because it is very easy to make the robots and we have a very fast vision system.

In this paper, we describe the technical tips for developing the robots in detail and explain our strategy for winning the game.

1 http://www.mirosot.org

2. System Architecture

Our system consists of robots, a host computer, a one-way radio link from the host computer to each robot, and a vision system.

2.1 Robot Hardware

Our robot consists of very simple components: batteries, the hacked R/C (radio control) servo motors, the wheels, and the R/C receiver module. These components are very cheap and easy to obtain in the hobby shop. Using double-sticky tape, we fix these components on the robot's body together.

To drive the robot, we employ the differential drive

mechanism using a pair of Hitec HS-80 servo motors, which were modified to spin all the way around and to have a lower gear ratio. A key technique in the modification is to remove a potentiometer from the servo motor. Originally, an unmodified servo motor has integrated circuit and potentiometer to implement a closed-loop position control system. After removing the potentiometer, the closed-loop position control system is changed to the closed-loop velocity control system. Now, the velocity of the servo motor depends on a *pulse-code modulated signal* sent by the radio. In [3], you can see more explanations about how to make the R/C servo motor continuously revolvable.

The robot's wheels are made to have the diameter with *32mm* and standard rubber O-rings. The robot is powered by four 1.2 volt NiCd batteries wired in series. The power line is connected to two RC servo motors via the receiver. Fig. 1 and Fig. 2 shows the components of the robot and our soccer robot respectively.

2.2 Radio Link

We employ one-way radio link from the host computer to the robots using a commercial R/C transmitter and receiver. A conventional R/C transmitter has a trainer mode, which enables the student transmitter to control the radio control model. To control the robots directly, we set the trainer mode on and let the host computer send the *pulse-code modulated control signal* to the transmitter through a trainer jack.

Using the programmable timer 8253, we developed a interfacing board between the host computer and the R/C transmitter. All the modes of 8253 are used to generate a train of pulses of varying widths. These pulses are updated at a given period, typically set to *20ms*. This updating frequency matches the cycle frequency of vision system well so that a motion control of our robots can be possible based on the visual feedback.

2.3 Host Computer

The success of the remote-brainless control method depends on the performance of host computer, which is responsible for controlling and cooperating all the robots in our case. Our host computer receives the tracking data of the ball and the robots over the serial port from the vision board, computes the desired motion of the robots, and sends the control signal to the robots over the R/C transmitter. We chose a desktop personal computer with a 200MHz Intel Pentium-Pro microprocessor as our host computer.

2.4 Vision System

In our case, fast and accurate spatial sensing is very crucial to the success of controlling the robot's motion, because our robots have no processor to control their motion. The host computer, which is responsible for controlling all the robots, computes the desired velocity of the robot's each wheel and sends the corresponding control signal to the robots, using the visual information.

Our vision system consists of a video camcoder and the Cognachrome vision system[2]. The Cognachrome vision system is able to track several objects in the field of view at a full 60Hz frame rate, with performance degraded to 30Hz when the number of tracking objects is increased. The system can be trained to recognize up to three colors simultaneously, and there may be many objects of each color present in the field of view. For each object, the system calculates various statistics of the recognized bulbs such as centroid, area, and orientation of major and minor axes when desired. The user can obtain the tracking data in a user-defined format over one of the board's two asynchronous serial ports [4].

Although the Cognachrome vision system has a good performance in speed and accuracy, we have some problems of distinguishing each robot when two robots move very close to each other. We think that this problem can be solved by improving our software because the team from the Newton Labs won first place using the same vision system.

3. Control Architecture

Our software for controlling the robots has the hiearchical structure with two levels. The low level of the control software is responsible for controlling the speeds of two wheels of robot and implements the basic motions such as rotation, translation, kicking motion, and so on. The high level takes the current states of the robots and the ball, and specifies the desired motion of our robots, using basic motions. All these softwares run on our host computer.

3.1 Low Level Control with Basic Motions

Our robot with the differential drive belongs to the nonholonomic system that is known to have some

2 http://www.newtonlabs.com/cognachrome

difficulties in controlling a motion [5]. To minimize such difficulties, we use a simple rotate-and-go strategy to put our robots on a target position.

The basic motion control software functions are divided into *rotate()*, *goforward()*, *gotoxy()*, *kicka()*, *kickf()* in large. The *rotate()* and *goforward()* basically adopted the proportional gain controller structure and tuned up to the best status adjusting various parameters adaptively. The *rotate()* converges to the desired orientation generating the opposite velocity for each wheel with its absolute value proportional to an error in angle. Given desired position, *goforward()* calculates the distance error between the disired and the current and generates the average velocity of wheels. The velocity command of each wheel is determined by adding and subtracting the term proportional to the angle error between the orientation to the goal and the current angle. Merging above two functions, *gotoxy()* is implemented. When large angle error occurs, this function uses function *rotate()* so the robot stands still and spins around to tune up the angle. Otherwise, the robot reaches the goal with the curved path using *goforward()* function. Actually our robot's generic mobility is very high as we can see from the manual movement. But in case of computer control, the gain is diminished to prevent overshoot and obtain the accurate regulation. As a result, robot speed was much slower compared to other teams. This is not because our control algorithm is bad but the handmade robot body is troublesome. Despite appling the same velocity command to each wheel, the robot does not go straight. In backward motion, the problem becomes more serious due to the disappearance of casting effect. Depending on the situation, the robot changes the logical heading to optimize the reaching time.

The *gotoxy()* is easy to implement cause the mission becomes complete reaching the goal. However, kicking motion needs the regulation of the final orientation as well as the final position. To meet this requirement, the virtual goal algorithm(VGA) was devised. VGA generates the circle path which satisfies desired orientation at the final position and guides the robot to follow the path using *gotoxy()* function. The virtual goal is produced at every control instance. Although the curvature robot makes relies on the average velocity and the velocity difference, VGA performs well without the robot identification cause it mitigates orientation error all the way to the goal. VGA is implemented and named *kicka()*.

The *kicka()* bears a defect which comes form physical constraint. In some configuration, the circle path which leads the robot to the goal with the desired orientation does not exist. Moreover, tracking the generated circle path might be time-consuming though there exists the path. To overcome these problems, *kickf()* function was made. When the difference between the final desired angle and the angle toward the goal is greater than 45 degree. The *kickf()* introduces virtual goal with virtual angle behind the final heading and performs *kicka()*. The virtual angle is decided based on the distance to the goal not to intrude the forbidden region with high ball existence probability. It reduces the time to reach the goal and brings the additional effect of collision avoidance.

3.2 High Level Control

In the MIROSOT contest, strategy is the most important part for the victory because most hardware abilities are equal or not so much better than the others. Fast vision system enables most teams to control the robot agent in real time about more than 30 frames per second.

Our team's strategy is based on the regions which is made by drawing two lines crossing the center position of ball and spaced about some angle between two lines (See Fig. 3.).

These angles are chosen according to the attack direction. If attack direction is toward the opponent's goal area, the θ_1 is determined by the geometric configuration. From these regions and our robot positions, host computer determines how each agent should act. There are four regions which have strategic meaning with our robot agents configuration. The angles behind the ball, θ_2 and θ_3 are somewhat large so as to make large defense region. We name it *bravo-cone* strategy, which is valid when ball is not near the wall. We have all 10 configurations from 4 regions and 2 robot locations. Basic actions are as follows. When robot locates in region 1, this robot should move out of region 1 in order for the other robot to kick the ball toward the opponent's goal area. When robot locates in region 2 or region 3, this robot should take action for assistance the other robot or move to the region 4 for kicking the ball. When robot locates in region 4, this robot move to the attack line and kick the ball.

The bravo-cone has command set to each configuration and command each robot to move appropriately according to the command set. This strategy is simple to implement and has light computation load. This strategy showed good offense and defense when implemented in real game since

ball position plays the important role in this strategy. When ball is near wall we applied some heuristic commands since Bravo-Cone is not valid.

4. Conclusions

In this paper, we describe the technical tips for developing the remote-brainless controlled system in detail and explain our software architecture for implementing the control of our robots and our strategy. The remote-brainless control method has some advantages that it is very easy and cheap to make the robots. However, some drawbacks occur during the motion control of the robots such that the symmetry of back and forth motion is not guaranteed. We think the vision-based control seems to be the most appropriate for the MIROSOT contest.

Our software architecture has the hierarchical structure and using some basic motions, we can implement our *bravo-cone* strategy. We hope to participate in future contests with the vision-based robot team and to show the performance of our strategy.

References

[1] J. -H. Kim, H. -S. Shim, H. -S. Kim, M. -J. Jung, I. -H. Choi, and J. -O. Kim, "A Cooperative Multi-Agent System and Its Real Time Application to Robot Soccer," *Proc. of IEEE Conf. on Robotics and Automation*, pp. 638-643, Albuquerque, New Mexico, Apr., 1997.

[2] *Proceedings of Micro-Robot World Cup Soccer Tournament*, Taejon, Korea, Nov., 1996.

[3] Joseph L. Jones and Anita M. Flynn, *Mobile Robots Inspiration to Implementation*, A. K. Peters, 1993.

[4] Anne Wright, Randy Sargent, and Carl Witty, *Cognachrome Vision System User's Guide*, Newton Research Labs, 1996.

[5] Jean-Paul Laumond, "Controllability of a Multibody Mobile Robot," *IEEE Trans. on Robotics and Automation*, Vol. 9, No. 6, pp. 755-763, Dec., 1993.

Table 1. Categorization of Control Structure

	CPU	motion planner
remote-brainless	×	host computer
vision-based	○	host computer
robot-based	○	each robot

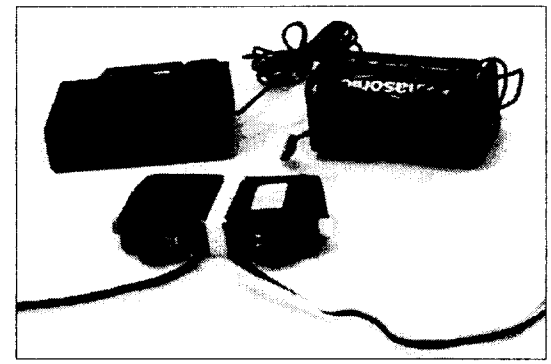

Fig. 1. The components of soccer robot

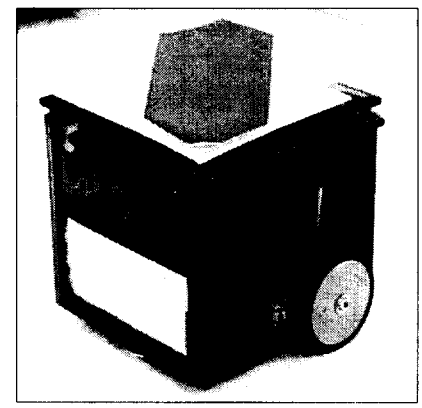

Fig. 2. Our soccer robot

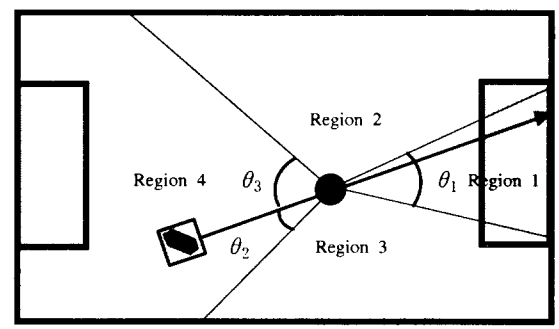

Fig. 3. Regions in bravo-cone strategy