

# Cornell BIG RED

## Small-Size-League Winner

*Raffaello D'Andrea and Jin-Woo Lee*

- The Cornell RoboCup Project was created to teach systems engineering concepts and practices to students to prepare them for designing, integrating, and maintaining highly complex systems. Another objective of the project is to explore the interplay between AI, dynamics, and control theory. This article describes the Cornell RoboCup team, which won the RoboCup-99 small-league championship in Stockholm, Sweden.

The Cornell RoboCup Project was created to teach systems engineering (Blanchard and Fabrycky 1997) concepts and practices to students to prepare them for designing, integrating, and maintaining highly complex systems. Another objective of the project is to explore the interplay between AI, dynamics, and control theory. This article describes the Cornell RoboCup team, which won the RoboCup-99 small-league championship in Stockholm, Sweden. The article is organized as follows: We first describe the overall system, followed by the AI architecture used in the competition. We then discuss the high-fidelity simulation environment used throughout development, followed by our conclusions.

### Overall System

The Cornell University team consists of two mechanical designs, one for the field players and one for the goalie (figure 1). All robots have a unidirectional kicking mechanism powered by a solenoid (two for the goalie). The kicking mechanism can impart a maximum velocity of two meters a second relative to the robot. The field players have a design mass of 1.5 kilograms, a maximum linear acceleration of  $5.1 \text{ m/s}^2$ , and a maximum linear velocity of 2.5 meters a second. The goalkeeper has a different design from the field players; it is equipped with a holding and kicking mechanism that can catch a front shot on goal, hold

it for an indefinite amount of time to find a clear pass to a teammate, and execute the pass.

The main function of the on-board electronics is to receive left and right wheel velocity signals by wireless communication and implement local feedback control to ensure that the wheel velocities are regulated about the desired values. The on-board electronics also control the robot kicking mechanism.

The global vision system runs at a speed of 35 hertz with a resolution of  $320 \times 240$ . A dedicated vision workstation identifies the ball and robot locations as well as the orientation of the robots. The basic algorithm used is blob analysis (Gonzalez and Woods 1992). To determine the identity of each robot and its orientation, the robots have color patches on top as well as the team color marker (blue or yellow Ping-Pong ball). The role-based AI is implemented on a dedicated workstation.

### Artificial Intelligence

In this section, we discuss the AI subsystem and trajectory control.

#### Role-Based Strategy

The AI subsystem is divided into two parts: The high-level AI takes the current game state (robot and ball positions, velocities, and game history) as input and determines role of each robot, such as shooter, defender, midfielder, and goalie. The reader is referred to Veloso et al. (1999), and the references therein, for a thorough description of the application of role-based systems in robotic soccer. Once the role is determined, desired final conditions such as time to target, robot orientation, and robot velocity are computed from the current state. More than 20 roles are preprogrammed. The low-level AI subsystem resides on each role and generates the trajectory to the target point and computes the wheel velocities to transmit to a robot.



Figure 1. Cornell BIG RED.

## Trajectory Control

The function of low-level AI is to generate trajectories and control the robot to follow the trajectories. It takes as input the current state of the robot and the desired ending state. The current state of a robot consists of the robot  $x$  and  $y$  coordinates, orientation, and left and right wheel velocities. A desired target state consists of the final  $x$  and  $y$  coordinates, final orientation, and final velocity as well as the desired amount of time for the robot to reach the destination.

Two position feedback loops are used for the robot's trajectory control. The first is a local feedback loop, the second a global feedback loop. The *local feedback loop* resides on the microcontroller of each robot and is responsible for controlling the motor position. The *global feedback control loop* also has a position feedback loop by way of the global vision system and ensures that the robot follows the desired trajectory. See Jones and Flynn (1993), and the references therein, for a general description of this two-degree-of-freedom decomposition of the control strategy. The desired velocity of each of the robot wheels is generated and then transmitted to the robots through the radio frequency communication link at every sixtieth of a second. Figure 2 shows the schematic diagram for the entire trajectory control loop.

The low-level AI needs to be efficient and robust to imperfections in robot movement. Currently, our algorithm can run more than 60 times a computation cycle, which is sufficient considering that it only needs to be run at about 4 times a cycle (for 4 robots excluding the goalie).

This complex problem is solved by breaking the problem of trajectory generation into two parts: The first part generates a geometric path. The second part calculates wheel velocities such that the robot stays on the path and completes it in the allocated time.

**Generating a Geometric Path** Our geometric path is represented by two polynomials in the  $x$  and  $y$  coordinates of the robots. The  $x$  coordinate polynomial is a fourth-degree polynomial, and the  $y$  coordinate polynomial is third degree.

$$x(p) = \sum_{k=0}^4 \alpha_k p^k$$

$$y(p) = \sum_{k=0}^3 \beta_k p^k$$

The task is to solve for the nine polynomial coefficients for a particular path requirement. The nine constraints on the polynomial path are (1) initial  $x$  coordinate, (2) initial  $y$  coordinate, (3) initial orientation, (4) initial curvature (determined by the initial left and right wheel velocities), (5) initial speed, (6) final  $x$  coordinate, (7) final  $y$  coordinate, (8) final orientation, and (9) final speed. It might seem strange that initial speed and final speed are constraints because they are not geometric features of the path, but we can actually transform them into the geometric feature of "path stiffness." For example, if the initial robot speed is slow, the path beginning can change curvature quickly (that is, low initial path stiffness), but if the initial speed is high, the path beginning shouldn't change curvature too quickly (that is high initial path stiffness).

**Generating Wheel Velocities** Every point on the geometric curve has a *curvature value*, which defines a relationship between the left wheel velocity  $v_l$  and the right wheel velocity  $v_r$  at that point in the curve. This relationship is as follows (Thomas and Finney 1996):

$$v = \frac{v_l + v_r}{2}$$

$$v_l(1 + \kappa \cdot r) = v_r(1 - \kappa \cdot r)$$

where  $\kappa$  is the curvature of the path,  $r$  is the half distance between the two wheels, and  $v$  is the forward moving velocity of the robot (figure 3). Thus, we simply need to choose a forward-moving velocity of the robot to solve for  $v_l$  and  $v_r$  at every point on the curve, which can then be sampled at the cycle rate of our AI system. Obviously, the forward-moving velocity is constrained by the time to target as well as mechanical limits of the robot.

Even though each run of this algorithm generates a preplanned path from beginning to

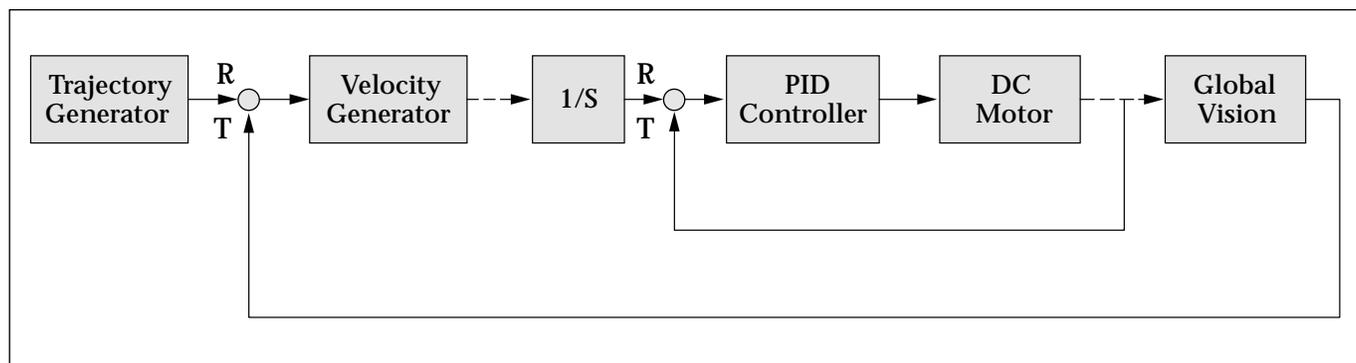


Figure 2. Schematic Diagram for Robot Control.

end, it can be used to generate a new path after every few cycles to compensate for robot drift. The continuity of the paths generated is verified through testing. However, this algorithm breaks down when the robot is near the target because the polynomial path generated might have severe curvature changes. In this case, the polynomials are artificially created (and not subject to the previous constraints) on a case-by-case basis, and these polynomials are guaranteed to be smooth.

## High-Fidelity Simulation

To provide a realistic testing platform for our AI system, we constructed a simulation of the playing field that models the dynamics of our environment (figure 4).

The dynamic modeling of our system is performed by a working model two-dimensional rendering of the complete playing field.<sup>1</sup> The model includes two teams of five individual players, the game ball, and the playing field. Real-world forces and constraints are modeled, including the motion of the tires and the inertia of the robots and ball. Additionally, the physical interactions between the players and between each other, the ball, and the playing environment are all modeled in the two-dimensional environment.

The simulator accepts external input and output every one-sixtieth of a simulated second, the rate at which the information from the actual vision system is updated, and robot commands are issued. To simulate the time lag and noise we encounter in our real-world simulation, the working-model parameters are passed into MATLAB,<sup>2</sup> where random noise, error, and delay are introduced to model the limitations of the vision system. This information is then passed to the AI module. Information that is typically fed back from the AI mod-

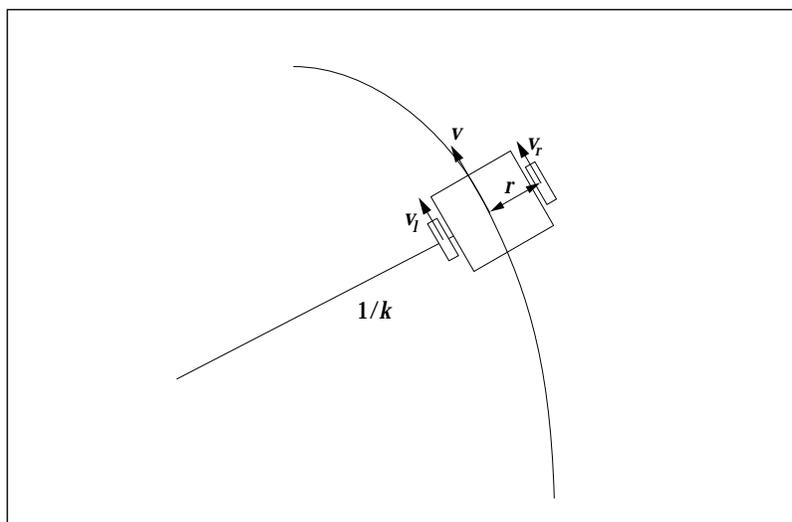


Figure 3. Trajectory Generation.

ule into our real-world robots is then delayed and interpreted in MATLAB before it is applied to the model of our system, to simulate the lag associated with our real-world system.

The simulator provided a means of testing the AI play by play even before our robots were fully constructed. It highlights the real-world problems that exist in a dynamic system and provides insight into solving these problems within an accurate representation of our playing environment. The simulator is a convenient, easy to use, and fairly accurate rendering of our real-world system.

## Conclusion

Even though our team performed well at the competition last year, there are many subsystems and components that need to be

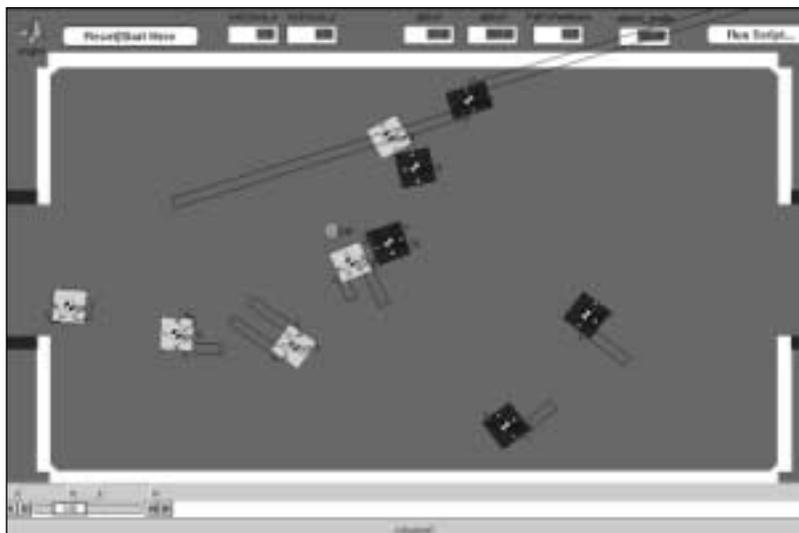


Figure 4. Simulation Platform.

improved. First, we need a more robust vision system. The current vision system performs well when operational but is very sensitive to the environment. In addition, it takes a long time to calibrate the system. One of our objectives for next year is to construct a reliable and robust vision system that can be set up in less than 30 minutes. Second, we need role coordination, which will allow us to implement set plays. Third, we need more refined trajectory generation, obstacle avoidance, and trajectory control, including analytic results relating to convergence and stability of the algorithms. Fourth, we need to reduce the system latency. Finally, we need innovative electromechanical designs.

### Notes

1. WORKING MODEL. Knowledge Revolution, San Mateo, California.
2. MATLAB. The Mathworks Inc., Natick, Massachusetts.

### References

- Asada, M., and Kitano, H. editors. 1998. *RoboCup-98: Robot Soccer World Cup II*. New York: Springer Verlag.
- Blanchard, S., and Fabrycky, W. J. 1997. *System Engineering and Analysis*. 3d ed. New York: Prentice Hall.
- Brookner, E. 1998. *Tracking and Kalman Filtering Made Easy*. New York: Wiley.
- Gonzalez, R. C., and Woods, R. E. 1992. *Digital Image Processing*. Reading, Mass.: Addison-Wesley.
- Jones, J. L., and Flynn, A. M. 1993. *Mobile Robots: Inspiration to Implementation*. Reading, Mass.: Addison-Wesley.
- Thomas, G., and Finney, R. 1996. *Calculus and Analytic Geometric*. Reading, Mass.: Addison-Wesley.

Veloso, M.; Bowling, M.; Achim, S.; Han, K.; and Stone, P. 1999. The CMUnited-98 Champion Small Robot Team. In *RoboCup-98: Robot Soccer World Cup II*, eds. M. Asada and H. Kitano, 77–92. Berlin: Springer Verlag.



**Raffaello D'Andrea** received a B.A.Sc. in engineering physics from the University of Toronto and an M.S. and a Ph.D. in electrical engineering from the California Institute of Technology. Prior to graduate school, he was employed as an electrical engineer at Bell Northern Research, where he designed packet-switching hardware. He has been with the Department of Mechanical and Aerospace Engineering at Cornell University since 1997, where he is an assistant professor. His research and teaching interests include the development of computational tools for the robust control of complex interconnected systems and systems engineering. D'Andrea has been the recipient of a Natural Sciences and Engineering Research Council of Canada Centennial Graduate Fellowship (1991–1996), the 1995 American Control Council O. Hugo Schuck Best Paper Award, the 1996 IEEE Conference on Decision and Control Best Student Paper Award, the 1999 Mechanical and Aerospace Engineering Shepherd Teaching Prize, and the National Science Foundation CAREER award.



**Jin-Woo Lee** received a B.A.Sc. in mechanical engineering from Seoul National University and an M.S. and a Ph.D. in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST). He was employed as a visiting lecturer at Cornell University from 1998 to 2000, where he instructed the Cornell RoboCup team. He has been with the Department of Mechanical Engineering at KAIST since 2000, where he is a member of the research staff. His research interests include robotics and optimal control.

# Seventeenth International Joint Conference on Artificial Intelligence

Washington State Convention & Trade Center  
Seattle, Washington • USA  
August 4 - 10, 2001

I J C A I ♦ O 1

*Sponsored by*  
The International Joint Conferences  
on Artificial Intelligence, Inc. (IJCAII)  
and American Association for  
Artificial Intelligence (AAAI)

*Conference Chair:*  
Hector J. Levesque, University of Toronto • hector@cs.toronto.edu

*Program Chair:*  
Bernhard Nebel, Albert-Ludwigs-Universität, Freiburg • nebel@informatik.uni-freiburg.de

*Local Arrangement Chair:*  
James Hoard, Boeing, Seattle • james.e.hoard@boeing.com

*Secretary-Treasurer:*  
Ron Brachman, AT&T Labs - Research • rjb@research.att.com

*General Information:*  
American Association for Artificial Intelligence  
445 Burgess Drive • Menlo Park, CA 94025, USA  
650-328-3123 • 650-321-4457 (fax)  
Email: ijcai@aaai.org

W W W . I J C A I . O R G

# Computers & Thought



Edward A. Feigenbaum  
& Julian Feldman

EDITORS

ISBN 0-262-56092-5 560 pp., index.

The AAAI Press • Distributed by The MIT Press  
Massachusetts Institute of Technology, 5 Cambridge Center, Cambridge, Massachusetts 02142  
To order, call toll free: (800) 356-0343 or (617) 625-8569.  
MasterCard and VISA accepted.