AI Magazine Volume 21 Number 3 (2000) (© AAAI)

ARVAND

A Soccer Player Robot

Mansour Jamzad, Amirali Foroughnassiraei, Ehsan Chiniforooshan, Reza Ghorbani, Moslem Kazemi, Hamidreza Chitsaz, Farid Mobasser, and Sayyed Sadjad

■ ARVAND is a robot specially designed and constructed for playing soccer in the medium-size league according to RoboCup rules and regulations. This robot consists of a moving mechanism, motion-control hardware, software, and a wireless communication system. The motion mechanism consists of a drive unit, a steer unit, and a castor wheel. Motion control is carried out by a special control board that uses two microcontrollers to carry out the software system decisions and transfers them to the robot mechanics. The software system performs real-time object recognition at the rate of 16 frames a second. Playing algorithms are based on deterministic methods and communication among robots. We have constructed four such robots and successfully proved their performance at RoboCup-99.

The RoboCup initiative has initiated a new field of research in autonomous robotics by encouraging universities and robotics research centers to construct a team of robots that could play indoor soccer with another team according certain rules and regulations.[1] These teams of robots compete in robot world soccer games each year. RoboCup-99 was the third such games, held in Stockholm, Sweden, in August. Our team became champion among 21 teams in the middle-size–league robot competitions.

Our goal was to construct omnidirectional autonomous robots with high maneuvering and fast decision-making capabilities, which are two key issues in a soccer game. Therefore, we designed a special mechanics that provided a fast and flexible omnidirectional movement especially when looking for the ball and dribbling. At RoboCup, objects are identified by color. All robots on each team have one identical color mark that distinguishes them from other team robots. Therefore, object finding can be done by finding color regions in the field. Although infrared sensors and laser scanners can be used to calculate object distances, we calculated a metric distance from the moving robot to a specific color region to determine the distance between the robot and the object.

The special mechanical design of our robots, a simple color segmentation and distance calculation routine, gave our robots a fast and high performance in real competitions. Our group members designed and constructed all parts of the robots. These robots have a controllable maximum speed of 0.53 meters a second. In addition to the basic robot movement, the special design of its mechanics allows the robot to rotate around any point in the field. In practice, by calculating the distance between ball center and robot geometrical center, the robot is commanded to rotate around the ball center. Figure 1 shows a picture of our player robot.

The machine vision system uses a home-use movie camera and a frame grabber. Our fast image-processing algorithm can process as many as 16 frames a second and can recognize objects at this rate. Software design is based on object-oriented methods and written in C++ using a DJGPP (DJGUN plus plus) compiler in MS-DOS. Communications between robots are done using a wireless network under TCP (transmission control protocol) protocols.

Mechanical Architecture

Considering the motion complexity of a soccer player robot, a particular mechanics was designed and implemented that, together with the motor's current feedbacks, to a good extent simplified the playing algorithms.

Motion Mechanism

ARVAND consists of two motion units in front and one castor wheel in the rear. Each motion unit has a drive unit and a steer unit. The function of the drive units is moving the robot and

Copyright © 2000, American Association for Artificial Intelligence. All rights reserved. 0738-4602-2000 / $2.00

Figure 1. A Player Robot.

that of the steer unit is rotating the drive unit around the vertical axis of the wheel. The drive unit consists of a wheel that is moved by a DC motor and a gearbox of 1:15 ratio (Shigley 1986). The steer unit uses a DC motor and a gearbox of 1:80 ratio. For controlling the steer unit, the optical encoders are mounted on respective motor shafts, with resolution of 1 pulse representing 0.14 degrees of drive-unit rotation. Figure 2 shows a block diagram of the robot mechanical architecture.

This architecture for motion mechanism provides the following capabilities:

First, by rotating the drive unit around its vertical axis, the rotation center of the robot changes accordingly, allowing the robot to turn around any point in the plane. This point can be selected inside or outside the robot. It is necessary to adjust the speed of two drive units according to the following formula (Meriam 1993):

$$v_1 \cdot r_2 = v_2 \cdot r_1 \qquad (1)$$

v_1 and v_2 are the speed of the left and right drive motors, respectively. r_1 and r_2 are the distance of the left drive unit and right drive unit, respectively, from the rotation center. Therefore, the robot rotation center will not depend on the robot gravity center and the position of drive units in the robot.

Second, in our software system, we can set the drive units to be parallel to each other and also have a specific angle related to the robot front. This mechanism is useful for taking out the ball when stuck in a wall corner and also for dribbling other robots. However, if the drive units are set parallel to the robot longitudinal axis, then simultaneous movement of drive units with equal speed can move the robot straight forward or backward.

Third, the kicker mechanism uses a solenoid with controllable kicking power. The power of kicking is controlled by a 24 DC voltage applied to the solenoid.

Hardware Architecture

The goal of our hardware architecture is to provide a control unit independent from the software system as much as possible and also reduce the robots' mechanical errors. The ARVAND hardware system consists of three main parts: (1) image-acquisition unit, (2) processing unit, and (3) control unit.

For all robots, we used a PixelView CL-GD544XP+ capture card as a frame grabber. It has an image resolution of 704 3 510 and 30 frames a second. The image-acquisition system of the goalkeeper consists of a Topica PAL charge-coupled–device (CCD) camera with a 4.5-millimeter lens in front and 2 digital Connectix Color QuickCam2s for the rear view. For other robots, we used a home-use movie camera.

The robot-processing unit consists of an Intel PENTIUM 233 MMX together with a main board and 32 megabytes of random-access memory. Two serial ports on board are used for communication with the robot control unit. The software system boots from a floppy disk. Figure 3 shows a block diagram of the robot hardware architecture.

The controller senses the robot and informs the processing unit of its status. It also implements the processing unit commands. Communication between the controller and the processing unit is done with two serial ports with RS-232 standard (Mazidi and Mazidi 1993). Two microcontrollers, AT8952 and AT8951 (MacKenzie 1995), are used in this controller. They control the drive units, the steer units, the kicker, and the limit switches. Two limit switches are mounted on each steer unit. The microcontroller counts the number of pulses generated by the encoders mounted on a motor shaft and controls the drive unit rotation with a resolution of 0.14 degrees. The motors' speed are controlled by PWM (pulse width modulation) pulse frequency of about 70 kilohertz.

Software Architecture

The software architecture of ARVAND consists of four main parts: (1) real-time object recognition, (2) motion control, (3) communication, and (4) decision making.

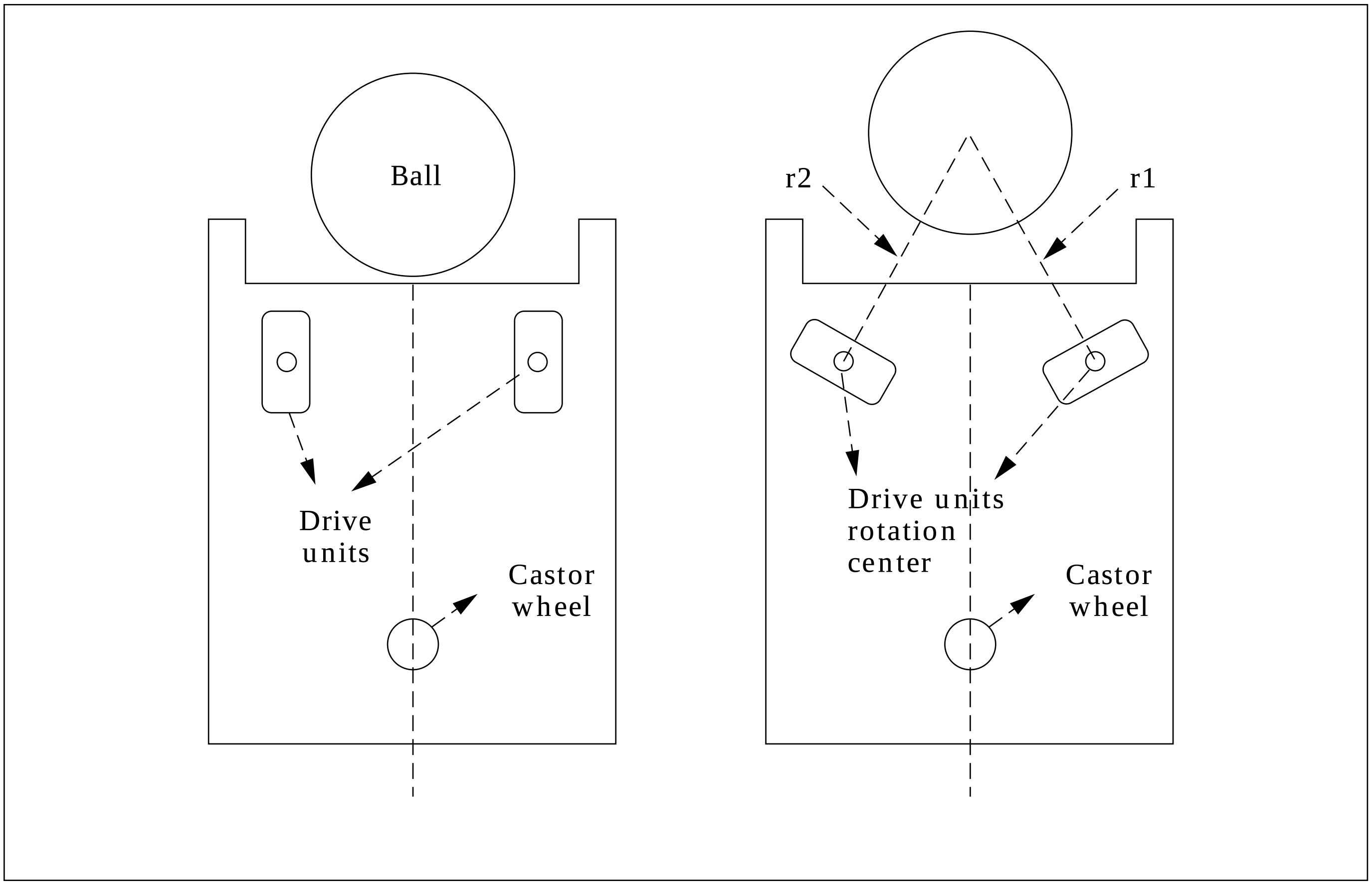


Figure 2. Player Robot Mechanical Architecture Block Diagram.

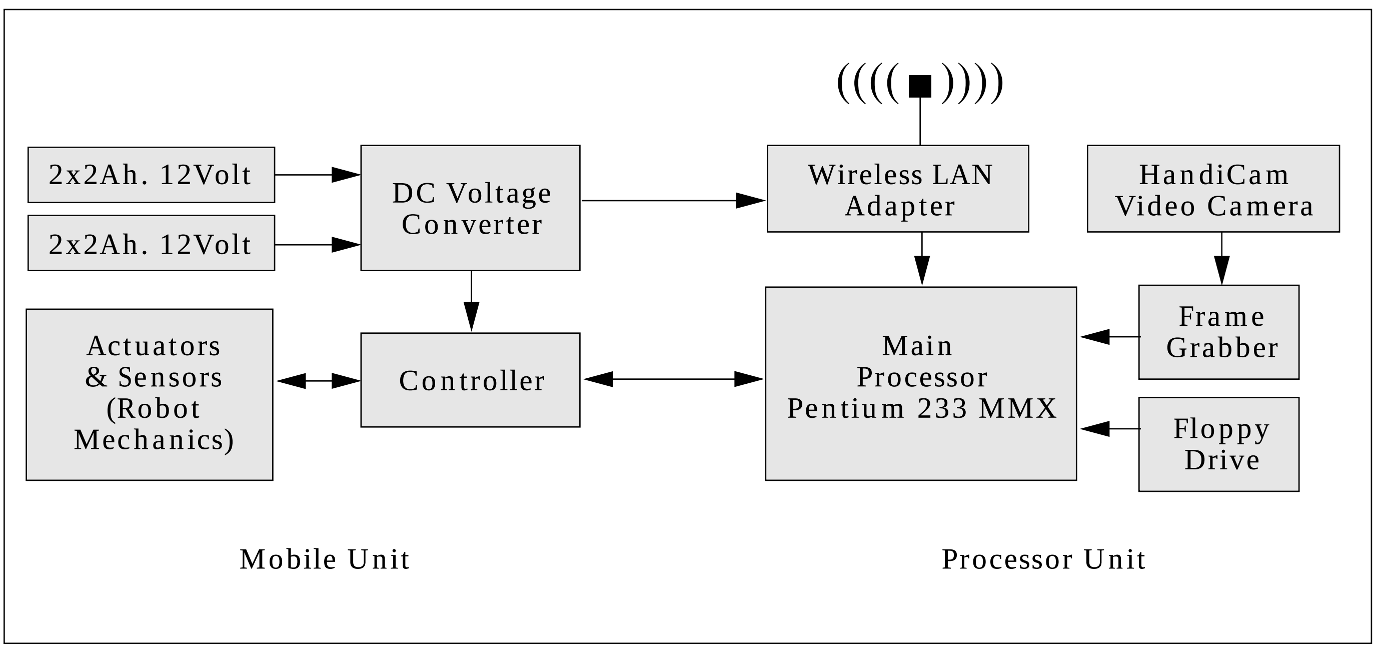


Figure 3. Hardware Architecture Block Diagram.

Real-Time Object Recognition

Object recognition is based on detecting a color using a hue, saturation, intensity (HSI) color model (Gonzalez and Woods 1993). In this model, a color can be detected by determining its domain in HSI space. To find all objects in a scene, the image array is processed from top to bottom only once. To speed up this routine, instead of examining each single pixel in the image array, only one point from subwindows of size $m_w \times m_h$ is tested.[2] If this point has the desired color (for example, red as ball), then to find the border point on this object, we keep moving in one pixel step upward until hitting a point with a different color.

From this border point, a clockwise contour-tracing algorithm is performed, and border points of the object are marked. If the object size is larger than a predefined threshold, then it is recognized as an object; otherwise, it is marked as a noise.

To find the next object, the search is continued from the start point where the previous object was found. In our search for the next object, the marked points are not checked again. At the end of this step, the small objects marked as noise are deleted; for the remaining objects, their size, distance from camera, and angle with respect to a line passing through a camera focal center are calculated.

To overcome the possible color error in the image-acquisition system during moving on the object contour, if we reach a pixel that has a color different from that of an object, but three of its four neighbors have the object color, then that pixel color is corrected to the color of the object and is added to the list of contour pixels. However, sometimes because of lighting conditions, one or more objects of the same type (not deleted with a noise-size threshold) are found within each other. In this case, only the object with maximum size is kept, and the remaining are neglected.

Motion Control

This module is responsible for receiving the motion commands from the decision-making module and putting the hardware to work. Communication between the processing unit and the robot controller is by way of two onboard PC serial ports. Some basic computations are done in this module, and commands are sent by serial ports to the microcontroller where they are executed.

For example, some commands are kick, go(forward), go(backward), rotate(left), rotate (right), and rotate round(left, 10) (this command stands for rotation around a point 10 centimeters straight from the robot geometric center).

Communication

Robot communication is done by wireless local area network (LAN) under TCP protocol.[3] Each robot has a wireless network card; a server machine outside the field coordinates messages between robots. The server's main responsibility is to receive the robots' messages and inform them about each other's status. For example, if one robot knows that another robot is holding the ball, it will not go for it.

Decision Making

The decision-making module is referred to that part of ARVAND software that processes the results of real-time object recognition, decides accordingly, and finally commands the motion-control software. We have taken a deterministic approach with these routines. This module is a finite-state machine whose input from changing state are machine vision results, motion-control hardware feedbacks, and server messages. The kernel of each robot-playing algorithm is finding the ball; catching it; finding the opponent goal; carrying the ball toward the goal; and finally, kicking the ball. However, a large number of parameters affect this main kernel and cause interrupts in its sequence because of dynamic change in the soccer field.

Our robots can avoid object collision by calculating the distance and angle from other robots and can change the speed of their motors to stay a certain distance apart from other robots. In addition, the robot ability to measure the current feedback of the motors allows it to determine the stuck situations, thus making the appropriate move to come out of that state.

Goalkeeper

In real soccer play, the players and the goalkeeper behave completely differently. We believe that the mechanics and hardware and software architecture of a player and goalkeeper not only should be different but also somehow should resemble the behavior of a human player. Our goalkeeper has a fast-moving sliding arm that can move left or right. It uses a rack-and-pinion mechanism.

In addition, we think the goalkeeper should be able to move straight toward left or right within the goal area; adjust itself in the goal center when displaced (which happens because of fast movements); and, of course, kick the ball when appropriate. These abilities are guaranteed by a combination of the following basic movements: move forward and backward, rotate around its geometric center, and move straight toward left and right. These movement abilities are guaranteed by installing four drive units in the goalkeeper's bottom such that each pair of front and rear drive units are rotated by one steer unit around their geometric centers.

Two digital cameras, installed at the left and the right sides of the robot rear, control its movement boundary within the goal area and also its ability to see the ball in corner areas. One CCD camera in front keeps track of the ball and other objects in the robots' front view.

Conclusion

One advantage of ARVAND is its mechanics capability to rotate around any point in the field, enabling it to rotate around the ball center while it looks for a goal position. In practice, this capability enabled us to implement special individual playing techniques in dribbling, coming out when stuck, and taking out the ball from a wall corner. Another advantage of our robot is its use of the MS-DOS operating system; it can be executed on a floppy disk, which is a cheaper and more reliable device compared to a hard disk, installed on mobile robots where there are many cases of bumping into the wall or other robots. Although using a solid-state hard disk could eliminate the problem of a hard disk crash in mobile robots, solid-state hard disks were expensive, and we could not afford to buy them.

Our robots made a good showing in the real games at RoboCup-99. We are improving our software algorithms mainly with regard to teamwork and multiagent systems. The wireless LAN enabled communication between the robots, which is key to the success of team-play algorithms as well as many individual techniques.

Notes

1. Please see the RoboCup official page at www.robocup.org.

2. m_w and m_h can be the sizes of the smallest object.

3. See www.geocities.com/SiliconValley/Vista/6552/l6.html.

References

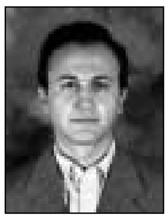
Gonzalez, R. C., and Woods, R. E. 1993. *Digital Image Processing*, Reading, Mass.: Addison-Wesley.

MacKenzie, I. S. 1995. *The 8051 Microcontroller.* Upper Saddle River, N.J.: Prentice Hall.

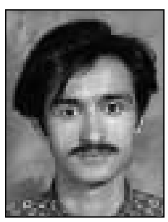
Mazidi, M. A., and Mazidi, J. G. 1993. *The 80 3 86 IBM PC and Compatible Computers, Volume 2.* Upper Saddle River, N.J.: Prentice Hall.

Meriam, J. L. 1993. *Dynamics.* New York: Wiley.

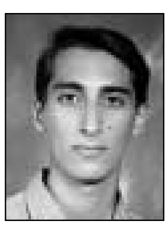
Shigley, J. E. 1986. *Mechanical Engineering Design.* New York: McGraw-Hill.



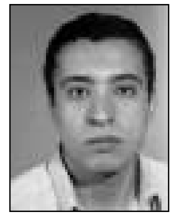
Mansour Jamzad is an assistant professor in the Department of Computer Engineering at Sharif University of Technology, Tehran, Iran. Jamzad received an M.Sc. in computer science from McGill University, Montreal, Canada, and a Ph.D. in electrical engineering from Waseda University, Tokyo. Jamzad's research interests include image processing, robot vision, computer graphics, and multiagent systems. Jamzad's e-mail address is jamzad@sina.sharif.ac.ir.



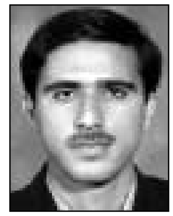
Amirali Foroughnassiraie is a graduate student in the Department of Mechanical Engineering at Sharif University of Technology, Tehran, Iran. Foroughnassiraie's research interests include design, manufacturing, and control of intelligent robotic systems, especially mobile robots. Foroughnassiraie's e-mail addrass is aafn@mech.sharif.ac.ir.



Ehsan Chiniforooshan is a software undergraduate student in the Department of Computer Engineering at Sharif University of Technology, Tehran, Iran. Chiniforooshan's research interests include machine vision, computational geometry, and graph theory. Chiniforooshan's e-mail address is chinifor@sharif.ac.ir.

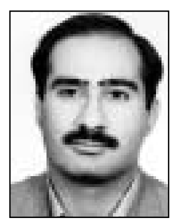


Reza Ghorbani is an aerospace undergraduate student in the Department of Aerospace Engineering at Sharif University of Technology, Tehran, Iran. Ghorbani's research interests include intelligent control, neural networks, mobile robots, and engineering design. Ghorbani's e-mail address is ghorbani@mech.sharif.ac.ir.

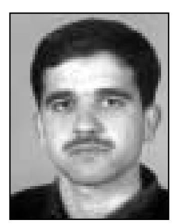


Moslem Kazemi is a hardware undergraduate student in the Department of Computer Engineering at Sharif University of Technology, Tehran, Iran. Kazemi's research interests include mobile robot control, teaching, and education. Kazemi's e-mail address is kazemi@linux.ce.sharif.ac.ir.

Hamidreza Chitsaz is a software undergraduate student in the Department of Computer Engineering and a pure math undergraduate student in the Department of Mathematics at the Sharif University of Technology, Tehran, Iran. Chitsaz's research interests include mobile robots and robotics, image processing and machine vision, speech processing, and number theory. Chitsaz's e-mail address is chitsaz@ce.sharif.ac.ir.



Farid Mobasser is a hardware graduate student in the Department of Computer Engineering at Sharif University of Technology, Tehran, Iran. Mobasser's research interests include multiagent robots and reconfigurable architectures. Mobasser's e-mail address is mobasser@ce.sharif.ac.ir.



Sayyed Sadjad is a software undergraduate student in the Department of Computer Engineering at Sharif University of Technology, Tehran, Iran. Sadjad's research interests include robot control algorithms and algorithmic graph theory. Sadjad's e-mail address is sajjad@linux.ce.sharif.ac.ir.

Edited by Jeffrey Bradshaw

The chapters in this book examine the state of today's agent technology and point the way toward the exciting developments of the next millennium. Contributors include Donald A. Norman, Nicholas Negroponte, Brenda Laurel, Thomas Erickson, Ben Shneiderman, Thomas W. Malone, Pattie Maes, David C. Smith, Gene Ball, Guy A. Boy, Doug Riecken, Yoav Shoham, Tim Finin, Michael R. Genesereth, Craig A. Knoblock, Philip R. Cohen, Hector J. Levesque, and James E. White, among others. 500 pp., ISBN 0-262-52234-9

Published by the AAAI Press / The MIT Press
To order, call 800-356-0343 (US and Canada) or (617) 625-8569. Distributed by The MIT Press, 55 Hayward, Cambridge, MA 02142