Design and Construction of a Soccer Player Robot ARVAND

M. Jamzad[1], A. Foroughnassiraei[2], E. Chiniforooshan[1], R. Ghorbani[2], M. Kazemi[1], H. Chitsaz[1], F. Mobasser[1], and S.B. Sadjad[1]

[1] Computer Engineering Deptartment
Sharif University of Technology
Tehran, Iran.
jamzad@sina.sharif.ac.ir, {chinif, chitsaz, sajjad}@linux.ce.sharif.ac.ir, {mobasser, kazemi}@ce.sharif.ac.ir
http://www.sharif.ac.ir/~ceinfo

[2] Mechanical Engineering Deptartment
Sharif University of Technology
Tehran, Iran.
{ghorbani, forough}@linux.ce.sharif.ac.ir
http://www.sharif.ac.ir/~mechinfo

Abstract. **Arvand** is a robot specially designed and constructed for playing soccer according to RoboCup rules and regulations for the medium size robots. This robot consists of three main parts: mechanics (motion mechanism and kicker), hardware (image acquisition, processing unit and control unit) and software (image processing, wireless communication, motion control and decision making). The motion mechanism is based on a drive unit, a steer unit and a castor wheel. We designed a special control board which uses two microcontrollers to carry out the software system decisions and transfers them to the robot mechanics. The software system written in C++ performs real time image processing and object recognition. Playing algorithms are based on deterministic methods. We have constructed 4 such robots and successfully tested them in a soccer field according to RoboCup regulations for middle size robots.

1 Introduction

In order to prepare a suitable ground for research in many aspects involved in Robocup, we designed and constructed all parts of the robots by our group members. These robots have a controllable speed of maximum 0.53 m/sec. In addition to the basic movements of robot, the special design of its mechanics, allows it to rotate around any point in the field. In practice, the distance between ball center and robot geometrical center is calculated and the robot can be commanded to rotate around the ball center until seeing the opponent team goal.

The machine vision system uses a widely available video camera and a frame grabber. Our fast image processing algorithm can process up to 16 frames per second and recognize objects in this speed. Objects are recognized according to their color and size. Communications between robots are done using wireless network under TCP protocols. The non deterministic software algorithms are designed based on object oriented methods and are written in C++ using DJGPP compiler in MS/DOS.

2 Mechanical Architecture

According to the motion complexity of a soccer player robot, proper design of its mechanics can play a unique role in simplifying the playing algorithms. In this regard, a specific mechanism was designed and implemented that together with the sensors and control feedbacks, to a good extent, verified our expectancy.

2.1 Motion Mechanism

Arvand consists of two motion units in front of the robot and one castor wheel in the rear. Each motion unit has a drive unit and a steer unit. The functionality of drive unit is moving the robot and that of steer unit is rotating the drive unit round the vertical axis of its wheel. The drive unit consists of a wheel which is moved by a DC motor and a gearbox of 1:15 ratio [1]. 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 the respective motor shafts and their resolutions are such that one pulse represents 0.14 degrees of drive unit rotation.

This mechanism has the following capabilities:

1. By rotating the drive unit round its vertical axis the rotation center of the robot changes accordingly and this allows 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 [2]:
$$v_1.r_2 = v_2.r_1 \tag{1}$$
where v_1 and v_2 are speeds of the left and right motors respectively, r_1 is the distance of the left drive unit from the rotation center and r_2 is the distance of the right drive unit from the rotation center. Therefore, the robot rotation center will not depend on the robot gravity center and on the position of drive units in the robot.
2. In our software system we can set the drive units to be parallel to each other and also have a specific angle related to robot front. This mechanism is useful for taking out the ball when stuck in a wall corner and also dribbling other robots.
3. The kicker consists of simple crowbar that connects the solenoid to a kicking arm. The kicking power is controlled by duration of 24 DC voltage applied to it.

3 Hardware Architecture

The goal of our hardware architecture is to provide a control unit independent from software system as much as possible and also reduce the robots mechanical errors.

Arvand hardware system consists of three main parts: Image acquisition unit, processing unit and control unit.

For all robots including the goal keeper we used a PixelView CL-GD544XP+ capture card which has an image resolution of 704x510 with the frame rate of 30 frames per second. The image acquisition system of goal keeper consists of a Topica PAL color CCD camera with 4.5 mm lens in front and two digital Connectix Color QuickCam2 for the rear view. For other robots we used a Handycam in front which could record the robot vision too.

The processing unit consists of an Intel Pentium 233 MMX together with a main board and 32MB RAM. Two serial ports onboard are used as communication means with the control unit. A floppy disk drive is installed on the robot from which the system boots and runs the programs.

The control unit senses the robot and informs the processing unit of its status. It also fulfills the processing unit commands. Communication between the control unit and the processing unit is done via two serial ports with RS-232 standard[3]. Two microcontrollers AT8952 and AT8951 [4] are used in control unit. They control the drive units, steer units, kicker and limit switches. Two limit switches are mounted on each steer unit. Microcontroller counts the number of pulses generated by the encoders mounted on a motor shaft to control the drive unit rotation. Each pulse represents 0.14 degrees of the drive unit rotation. The motors speed are controlled by PWM pulse frequency of about 70kHz

4 Software Architecture

Software architecture of **Arvand** consists of four main parts: Real time object recognition, Motion control, Communication and Decision making module. Due to the object oriented design of the software, we have defined 5 classes such as: Camera class (all related functions for working with frame grabber), Image class (machine vision functions), Motion class (motion functions which is the interface between software and hardware), Communication class (all TCP related wireless networking) and Motion class (all robot playing methods and algorithms).

4.1 Real time object recognition

Object recognition is based on detecting its color. We used HSI color model [5]. In this model a color can be detected by determining its domain in HSI space. To find all objects in a scene the image matrix is processed from top to bottom only once. In order to speed up this routine, instead of examining each single pixel in the image matrix, only one point from subwindows of size $m_w \times m_h$ (that can be the size of the smallest object) is tested. If this point has the desired color, then move one pixel upward until hitting a border point.

At this 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 to be a noise.

To find the next object the search is continued from the start point from which 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 objects marked as noise are deleted and for the remaining objects their size, distance from camera and angle are calculated.

4.2 Motion Control

This module is responsible for receiving the motion commands from the "Decision Making Module" and putting the hardware to work. As it is mentioned in the hardware architecture section, the communication between the processing unit and the control unit is via two onboard PC serial ports using RS-232. So, just some basic computations are done in this module and commands are sent via serial ports to the microcontroller where they are executed.

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

4.3 Communication

Communication between robots is done by wireless LAN under TCP protocol. The main kernel of communication class can be downloaded from [6]. Each robot has a wireless network card, and there is a server machine outside the field which coordinates messages between robots. The server also provides a useful user interface to command robots manually. Server's main responsibility is to receive the robots messages and inform them about each robot status. For example, if one robot knows that another robot is holding the ball it will not go for the ball.

4.4 Decision making

Principally, 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 deterministic approach in these routines. This module is a finite state machine (FSM) whose inputs come from changing state are machine vision results, motion control hardware feedbacks and server messages. Each robot playing algorithm kernel is finding the ball, catching it, finding the opponent goal and finally carrying the ball toward the goal and kicking. But there are a large number of parameters that affect this main kernel and cause interruppts in its sequence. For example, the main method for finding the ball is rotating. When our robot is moving inside the field it tries not to collide with other robots. This is done by calculating the distance and angle of other robots and change the speed of its motors such that object collision is avoided.

In addition, robot ability to measure the motors current feed back, allows it to determine the stuck situations and thus making appropriate move to come out of that state.

5 Conclusion

Arvand is the 2nd generation of robots constructed by our team. One advantage of **Arvand** is its mechanics capability to rotate around any point in the plane. This makes it possible for the robot to rotate around ball center while finding the 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 MS/DOS operating system, because it can be executed on a floppy disk which is a cheep and reliable device on mobile robot. Our robots showed a good performance in real games and we are going to improve our software algorithms based on individual techniques and also team play. The wireless LAN system used in our robots enabled the communication between robots which is the key to the success of team play algorithms and also many individual techniques.

References

1. Shigley, J.E., *Mechanical Engineering Design*, McGraw-Hill, 1986.
2. Meriam, J.L., *Dynamics*, John Wiley, 1993.
3. Mazidi, M.A., and Mazidi, J.G., *The 80x86 IBM PC and Compatible Computers, Volume II*, Prentice Hall, 1993.
4. MacKenzie, I.S., *The 8051 Microcontroller*, Prentice Hall, 1995.
5. Gonzalez, R.C., and Woods, R.E., *Digital Image Processing*, Addison-Wesley, 1993.
6. WATTCP http://www.geocities.com/SiliconValley/Vista/6552/l6.html