IUT Flash (Persia) Team Description

Dr. AliReza Fadaie Tehrani [1]

Mohammad Ajoodanian [1], Peyman Amini [2], Pejman Foroughi [2], Omid Teheri [2], Behrouz touri [2],

Ahmad Movahedian [2], Alireza Haghshenas [2], Omid Goodarzi [1], Hamid Reza Moballegh [2],

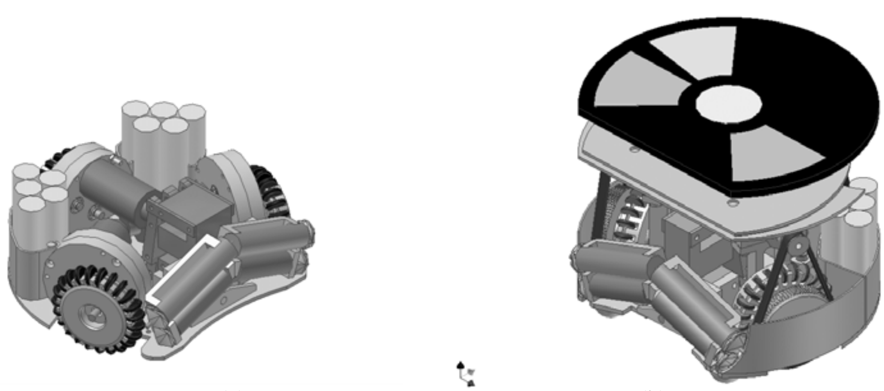
Mohammad Mehdi DaneshPanah [2], Amir Abdollahi [1], Mohsen Amiri [2], Omid Goodarzi [1], Masood Gheytuli [2], Iraj Hosseini [2], Hossein Ostadi [1], Hooman Samani [1]

[1] Isfahan University of Technology, Department of Mechanical Engineering, Isfahan 84154, Iran
[2] Isfahan University of Technology, Department of Electrical and Computer Engineering, Isfahan 84154, Iran

Abstract. Our project can be divided to two major sections: Hardware and Software. Each part was fully designed and produced by the members in charge. In turn, the software section, can be divided into the: "Image processing, Network, AI, trajectory, controller, coding and transmission" subsections, each being programmed on Delphi, as separate modules and then implemented in the master project. The processing is distributed on four computers, networked with coaxial cables. Three computers, connected to three cameras on top of the field, are in charge of image processing and a server finishes the remaining processes. The hardware, consisting of the robot itself and the receiver and driver circuit board were both designed and made by the students themselves.

1 Introduction

The IUT Flash team was formed in October 2001. Its main goal was to learn, since the project contains many different aspects that can be worked on. Some subjects studied during the work were: "Computer networks, System programming, Neural networks, Fuzzy decision-making, Fuzzy controllers, Image processing, control, and Data coding and transmission". A point about the project is the use of three cameras to get better results. Use of three cameras made us build a network mechanism for the data processing procedure. In this paper we will first introduce the hardware of robots (both mechanical and electrical) and then we will move to the software processing scheme and introduce the different units, in the order they are executed.



(a) (b)

Fig. 1. Our robot: (a) with gearbox. (b) Small size robot with Timing belt

2 Robot Hardware

The robots' mechanics consist of the wheel movement system and also the kicker and ball handling systems. The wheel movement system has three omni directional wheels (each has 24 little rollers on itself). Each wheel drives with a DC motor. Robots can reach a maximum speed of about 3 meters per second. Having three independently controlled wheels, the robot can rotate or move on every direct or curved path. The kicker's operation is based on a solenoid, which works with a Buck circuit (with a voltage about 60 volts). Two cylindrical plasticizer that inclined called spin-back which also has a damping characteristic is used for more control on Ball handling.

Each robot has five DC motors (three for driving wheels, and two for spin-back), which are driven by five drivers that are placed on the robot's main control board. The main board uses a 8951 microcontroller and a digital wireless receiver module that enables the robot to receive commands from a server computer. The program for the microcontroller is written in 8051 assembly language, which was first tested on Keil 8051 simulator and after debugging, implemented on the robots.

The microcontroller's program has three main parts:

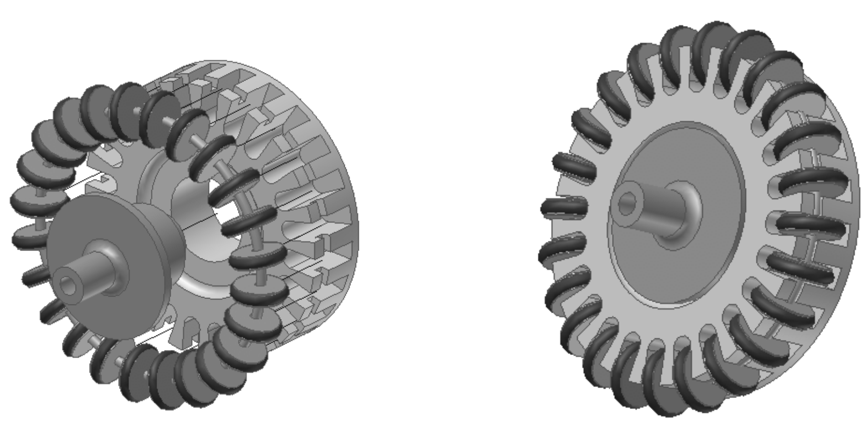
1. Receiving coded packets and decoding them to get the raw data needed by the robot.
2. Driving DC motors of the wheel rotation system and that of the kicker and spin-back, which is done by making proper PWM signals for switching the H-Bridge motor drivers.
3. Sending proper signal to a Watchdog-Timer.

All of the circuits and the electro-mechanical parts are supplied by a 24V, 1500 mAh Ni-MH, battery pack, constructed from 20 Ni-MH battery cells.

2-1 Mechanic

2-1-1 Optimizing the omni directional wheel

Decreasing the run-out of this wheel by increase the amount of little rollers from 12 numbers on per wheel in lost year to 22 numbers with the constant diameter so the run-out is nearly 0.08-0.1mm.It 's more better than lost year that the amount of run out was 1mm.



(a) (b)

Fig. 2. (a) The explosion view of new omni directional wheel. (b) Omni directional Wheel

2-1-2 Design and optimize the parameters of an effective ball handling mechanism

Using a cylindrical rollers which are parallel to the ground helps the ball controlling only at moving backward but if rollers are inclined (Angle α) about the Y axis(that is perpendicular to ground), this mechanism works like a center less grinding feeding mechanism so two rollers apply double torque on the ball not only in x (parallel to the ground) direction but also y, this will increase the linear acceleration of the ball (mass center) .By using two rollers the ball will remain in front of the robot when its rotates (CCW or CW).

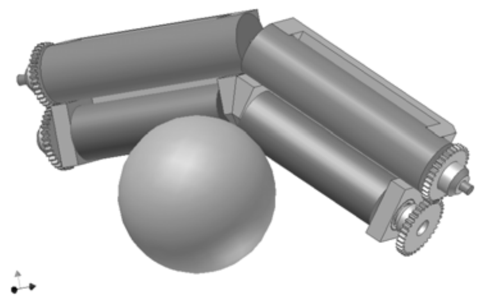


Fig. 3. Ball handling mechanism

3 Robot Software

3.1 Image processing algorithm

Two computers connected to three separate cameras, process the image and calculate the position, direction, velocity and the robots' ID numbers. The image-processing algorithm also determines the position and velocity of opposing robots as well as the position and velocity of the ball. For processing, the field of game is divided into two halves and an overlapped area seen by both of the cameras. The server gives the necessary commands to the image processing computers. The algorithm used to find objects is optimized to process the maximum number of frames. First it searches all of the pixels, when it finds an object, saves its coordination so the next time it can start back with the same point. The image grabbers' color system is 16bit RGB. For better image manipulation the RGB color space is converted to HSL (Hue, Saturation and Luminance). To recognize a certain color, a combination of conditions on Hue, Saturation and RGB is used. This procedure makes the color recognition independent from the change of brightness and other unwanted conditions.

The image processing procedure is delivers data 30 times per second. The image processor receives its raw data through fire wire port connected to a Sony digital video camera. Each frame is processed in less than 5 milliseconds.

3.2 Network

To finish the task of image processing, three computers (two clients which receive data from the cameras and one server which receive data from the other two) are networked to obtain a complete data from the field. The network physical layer uses the shunted ring topology. The UDP (User Defined Protocol) network protocol is used for the software communication layer. The data flow of the network is as follows: A half field data (the data representing the position and status of the robots, opponents and the ball) is transmitted to the server from each client computer, the server combines the data, constructs the complete image data and sends the appropriate commands (identifying which objects each client should search for) back to the clients. With the data completed it is passed to the AI (Artificial Intelligence) unit for further processing and to decide what each robot should do next.

3.3 Artificial Intelligence Algorithm

In this section the AI part of the software is briefly introduced. There are three distinct layers: AI Core, Role Engine and Behavior Engine.

AI Core receives the computed field data from image processing unit and determines the play state according to the ball position, opponents' positions and our robots' positions. Determination of the play state is done by fuzzy decision-making[2] for avoiding undesirable and sudden changes of behaviors or roles. Then it sends a set of roles assigned to the robots to Role Engine. Because there are many times in which the image-processing unit cannot see the ball, a memory is implemented in the AI Core for the position of ball which specifies who owns ball [7]. In AI Core roles are changed in a manner that there is a relationship between new roles and old roles, therefore robots never experiment sudden changes in roles (for example the role defense never changes to attack in next cycle).

Role Engine receives a set of roles from AI Core and provides the Behavior Engine with a set of behaviors for robots[6]. We have implemented twin or triple roles in which robots really cooperate with each other to do something.

Behavior Engine receives an array of behaviors from Role Engine and then according to the behavior each robot must does, with help of trajectory unit, it sets the control inputs for the controller.

3.4 Trajectory

Since each role has an execution method, which gives a point and an orientation, regarding the role of the robot, the output obtained after the execution of AI will be a set of vectors, each robot should reach. So the task of the trajectory will be to guide the robots through the opponents to reach the destination. The routine used for this purpose is the potential field method. In this method different electric charges are assigned to our robots, opponents and the ball. Then by calculating the potential field of this system of charges a path will be suggested for robot[8]. At a higher level, predictions can be used to foresee the position of the opponents and make better decisions to reach the desired vector.

3.5 Controller

So far we have reached a point where the trajectory has calculated a path in conjunction with the destination vector. If the robot moves on this path, it will reach the destination with no contact.

Due to the non-symmetric nature of the wheels movement system and different frictional forces in the wheels, we can't rely on exact operation of this open-loop system. Therefore, it's required to use a controller with a proper feedback to produce the correct control signals for the wheels of the robot in order to move accurately through the path.

The movement system has three parameters that are the speeds of the three wheels. Controlling the position of robot is done using a software implemented PID controller that obtains its feedback from the image-processing unit [4].

In Addition to the position controller developed in the software, each robot has an on-board velocity controller. The controller always corrects the direction and velocity of each robot to the value given by trajectory part.

3.6 Coding and transmission

The output of the control software is a pair of velocities for the wheels of each robot. These are then combined with the other settings such as the kicker and spin-back motors' data and the robot's ID. Then coding is done by adding parity bits, start and stop bytes. This is done to synchronize data transmission as well as to minimize any data error.

An 8951 microcontroller that is connected to the parallel port of the server computer does the coding procedure. The low-level routines send data from the PC to the microcontroller.

The microcontroller is programmed to redundantly send the settings to robots through its UART and by the radio transmitter module.

References

[1] Burdick, H.E. Digital Imaging theory and application. McGraw-Hill, NewYork, 1997.

[2] H.J.Zimmermann, H.J.Zimmerman. Fuzzy Set Theory and its Applications. Kluwer Academic Publishers, 4th Ed. 2001.

[3] Ripley, B.D. Pattern Recognition and Neural Networks. Cambridge University Press, 1996.

[4] Keigo Watnable, "Control of an Omnidirectional Mobile Robot", *Second International Conference Acknowledge-Based Intelligent Electronic Systems,21-23 April*

[5] Cantu, M. Mastering Delphi 6. Sybex. Inc, 2001.

[6] G.Weiss. Multiagent systems. A Modern Approach to Distributed Artificial Intelligence. MIT Press, 2000.

[7] S.J.Russell, P.Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, 2nd Ed 2002.

[8] R.R.Murphy. An Introduction to AI Robotics. MIT Press, 2000.