Cooperative Multi Agent Soccer Robot Team

Vahid Rostami, Saeed Ebrahimijam, P.khajehpoor, P.Mirzaei, and Mahdi Yousefiazar

Abstract**—**This paper introduces our first efforts of developing a new team for RoboCup Middle Size Competition. In our robots we have applied omni directional based mobile system with omni-directional vision system and fuzzy control algorithm to navigate robots. The control architecture of MRL middle-size robots is a three layered architecture, *Planning*, *Sequencing*, and *Executing*. It also uses *Blackboard* system to achieve coordination among agents. Moreover, the architecture should have minimum dependency on low level structure and have a uniform protocol to interact with real robot.

Keywords**—**Robocup, Soccer robots, Fuzzy controller, Multi agent.

I. INTRODUCTION

MRL is a research center at the Islamic Azad University of Qazvin which is mainly established based on the research activities of our RoboCup team members from 2002. MRL is also a good support for new established MSc Mechatronics Course in 2003. There are so many active research programs based on the RoboCup competition, final MSc thesis and industrial projects.

The MRL Middle size team has been started its work from Aug 2003 at this laboratory. Since last year, a few robot models are designed and built which are presented in this paper. Our team has been working for robotic competitions about two years and has gained so many valuable experiences in the field of robotics and Artificial Intelligent. Our goal is to make our robot well equipped by optimized algorithm of control, vision, path planning and playing strategies, according to the advanced mechatronics expertise.

II. MECHANICAL DESIGN

In a RoboCup competition, a robot must be prepared to accelerate in an appropriate direction, to recognize the ball and the other robots swiftly, to kick the ball with sufficient velocity, and with proper accuracy. It is evident that low weight, high acceleration, high velocity and appropriate safety factor are main parameters in designing of a RoboCup mobile robot. For achieving this purpose, optimization in selection of material is highly required for each part. [1],[2] We have designed two types of robot; 3-wheel triangle robot and 4-wheel Cartesian robot which are shown in Fig. 1.

Authors are with Iran Islamic Azad University, Qazvin, Iran (phone: +982813665275 fax:+982813665279, zip code : 34185-1416).

According to different maneuverability of these two robots, a team of both types could have more flexibility and powerful mobility.

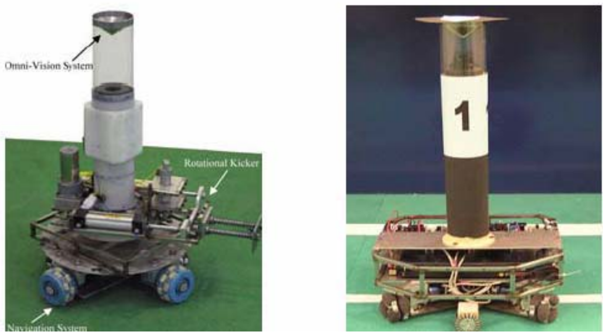

Fig. 1 Right: MRL's 3-Wheels robot; Left: MRL's 4-Wheels robot

The 3-wheel robot is energized by three 80 W DC Servomotors (Fig. 2a). Based on the physical characteristics of the robot, its Maximum speed and acceleration are 2 m/sec 3 m/sec2 accordingly. In order to reach the ball in a desired orientation, a robot with 3-wheel mechanism is quite faster and more flexible. For the kick mechanism, we have applied a spring capable of compressing and releasing to kick the ball. This mechanism might be charged and discharged in 3 seconds. A notable characteristics of this robot is its low weigh, which is totally 12 Kg including Batteries.

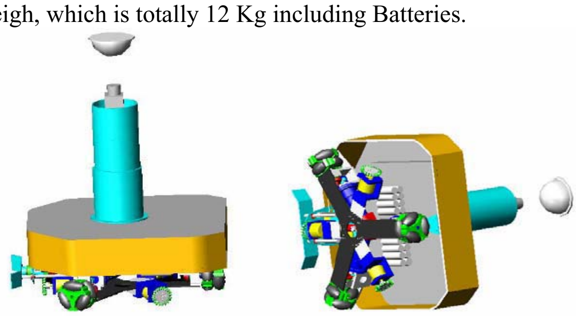

Fig. 2a MRL's 3-Wheels robot view and content of base

Second type presents a new kind of mobile robot which consists of four omni wheels and a rotational kicker part. As can be seen from Fig. 3, each direction(x or y) is supported with one DC servo motor which forces the robot to move. Two omni wheels obtain their acceleration from comparative generating motors, thus each pair of wheels force robot to move in one principle direction. Result is a relevant acceleration vector that causes displacement for mobile robot. Moreover, kicking system on top of the robot can turn 360 degrees which enables the robot to optimize the path reaching the ball with required orientation for proper kick. This saves the robot rotation time and also simplifies the control algorithm which is a significant advantage of this robot. Fig. 3. Left: Robot Navigation System; Each coupled omni wheels provide required acceleration. Relevant vector produces overall acceleration. Right: suspension system for reaching stability.

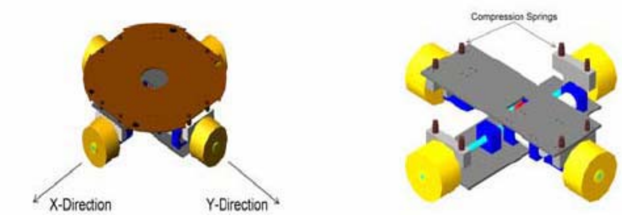

Fig. 2b Left: Robot Navigation System; Each coupled omni wheels provide required acceleration. Relevant vector produces overall acceleration. Right: suspension system for reaching stability

To design a robot with the mentioned abilities there are several problems which should be solved. As can be seen in Fig. 3, complicated design and instability are two main problems which are discussed in the following sections.

It is obvious that any rigid body standing on a surface with more than three contact points is unstable. To obtain stability, each wheel of the mobile robot must be supported with a suspension system which is designed and implemented for this purpose. In 4-Wheels robot the kick system is designed and implemented by Pneumatic cylinders which compress a spring mechanism. The energy stored in the spring could be released to kick the ball when is required. To provide estimated angle for shooting, all parts of the kicker part can revolve about center of the mobile robot. In Table I, some parameters for these two types of mobile robots are presented. [2],[3],[4]

TABLE I
3-WHEELS AND 4-WHEELS PARAMETERS

	4-Wheels Robot	3-Wheels Robot
Platform	Omni-Directional	Omni-Directional
Maximum speed	2 m/s	2 m/s
Maximum Accelera-tion	4 m/s^2	3 m/s^2
Kicker	Pneumatic, with rota-tion	DC Motor, Fixed on Robot
Weight	18 Kg	12 Kg
Wheels & kicker Motors	3 faulhaber DC 200 Watts	4 faulhaber DC 80 Watts

All items listed in Table II are used in both 3-Wheel and 4-wheel robots.

TABLE II
APPARATUS IN BOTH ROBOTS

Laptop	IBM T41 Processor 1.7GHz Centrino
Camera 1	Ieyes 320x640 Pixel - 30 FPS
Camera 2	Web Cam 240 * 320 (used in kicker part)
Image processing	Omni directional vision system
Other sensor	Dummy omni directional Encoder and infra-red sensors
Controller	Fuzzy Logic and PID controller

III. HARDWARE FEATURES

A multi-layered hardware system is implemented to control our soccer player robots which are shown in Fig. 3. This control system is composed of motor drivers which are connected to a master board via a bus mechanism. ATmega128 and ATmega8535 microcontrollers are used for master and drivers accordingly. A PID controller is designed and implemented to control the motor speeds. PID's parameters are predicted by an adaptive fuzzy algorithm.[5] Our vision system is based on two cameras; camera-1 is used as an omni-vision system and camera-2 is used to recognize the vicinity of the ball to the robot, whereas an IR sensor validates the presence of the ball in the vicinity of the kicker. The motor speeds are measured and controlled by using shaft encoders, and the robot position; and speed is sensed by a special shaft encoder mechanism attached to the robot base[6]. An IBM_T41 laptop is also used for higher layer processing system such as : image processing, artificial intelligence algorithms and playing strategies.

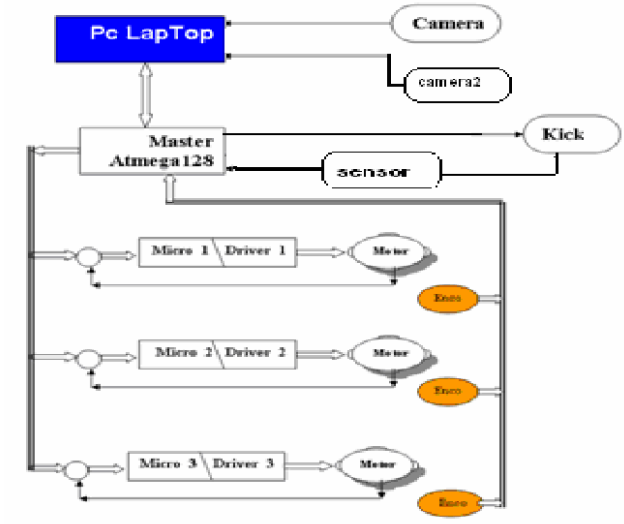

Fig. 3 Control system architecture

IV. FUZZY CONTROLLER

The fuzzy control algorithm (Fig. 4) calculates required velocities and positions of individual wheels to achieve the target by defining three parameters of the target (Xm , Ym, Ø m). The fuzzy controller implementation and solving inverse kinematics of the robot are allocated to the ATmega-128 microcontroller in order to reduce the load of the main PC. [7][8].

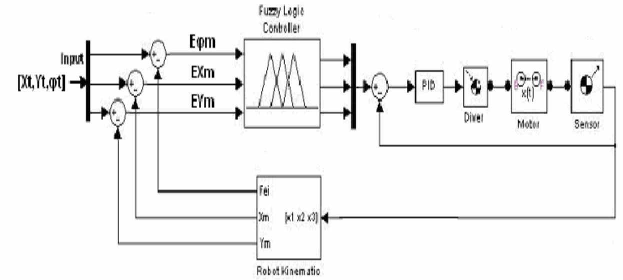

Fig. 4 Fuzzy controller block diagram

V. Communication

RS-232 protocol is used as an internal communication to connect Laptop to the Master board in order to pass the required data for robot movement. The communication between robots is based on wireless LAN with IEEE 802.11g protocol, as requested.

VI. Software Control Architecture

The control architecture of MRL middle-size robots is a three layered architecture where the top layer contains techniques for autonomously creating a plan of robot tasks, the middle layer is a sequencing capability to accomplish tasks and the bottom.

Functional layer provides standard robot skills that interface to the system hardware.

The *Planner* creates concrete plans according to some high-level, abstract plans, where each plan includes goals to be accomplished and chronological constraints between goals. *Interpreter* then decomposes the goals in to subtasks, and sends commands to *Skill Manager*. *Skill Manager* executes the commands, sending data back to the interpreter so that it can observe task execution. Status information is sent back to the planner; especially when strategy of the team is changed and planner have to create plans according to the new policy.

Debugging of this system is extremely difficult because of its distributed and real time nature. It needs to have a tool that can display sensor data, world-model data and action choices. The Monitoring unit logs the run-time data produced by agent's internal processes. Supervisor collects every agent's data, then tags and stores them for visualizing and analyzing. *Log Player* is a tool that reads the log-files and helps developers to explore them and find individual and teamwork problems of robots.

Log Player has a variety of display tools including *task-view* for viewing the hierarchical decomposition of tasks and *plan-view* for displaying and analyzing plan execution information.

Communication is necessary to create cooperative behaviors among teammates. MRL soccer agents use a memory called the *Blackboard* to share information with teammates [8],[9]. *Blackboard* is used to share World Model memory and also issue desirable information in order to carry out relational behaviors.

VII. Vision System

The main task of the vision system is to observe the environment around the player robot to determine direction and distance of all objects in the field (e.g. field, ball, goals ..).

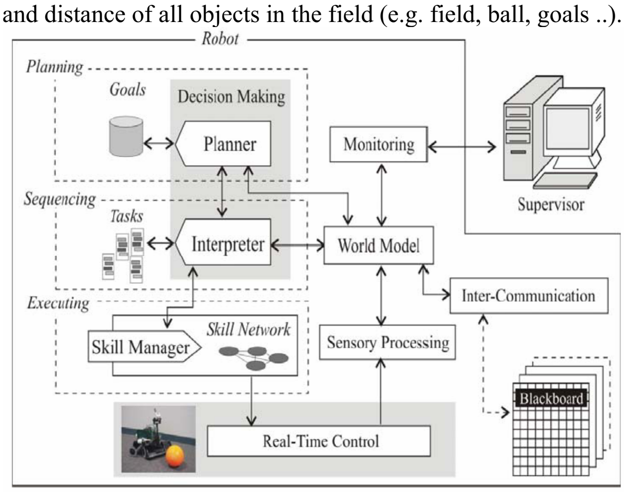

Fig. 5 Software Control Architecture for MRL Soccer Robots

Each robot has one omni-directional vision system with a uEye camera, and another simple web cam in front of the robot to detect the ball in near distances. The uEye camera has an interface of USB 2.0, and the laptop computer captures the image directly. This camera and its Omni-directional mirror are mounted on the robot as the camera optical axis is aligned with the vertical axis of the mirror. Many teams on RoboCup use omni-directional vision systems with different mirror geometries (e.g. Conic, Parabolic, Spherical and Hyperbolic). These are only simple general forms of omni-directional mirrors which are not used in our Vision System[2],[8]. At First a complex program was generated to design, analyze and simulate all kinds of omni-directional mirrors, then with this program we have fabricated a new kind of mirror called Isometric Mirror which has no distortion in its reflected image, and the distance scale remains constant in whole of image. Unlike other mirrors, the curve function of this mirror has no closed form and it is designed by numerical solution of some complex equations.

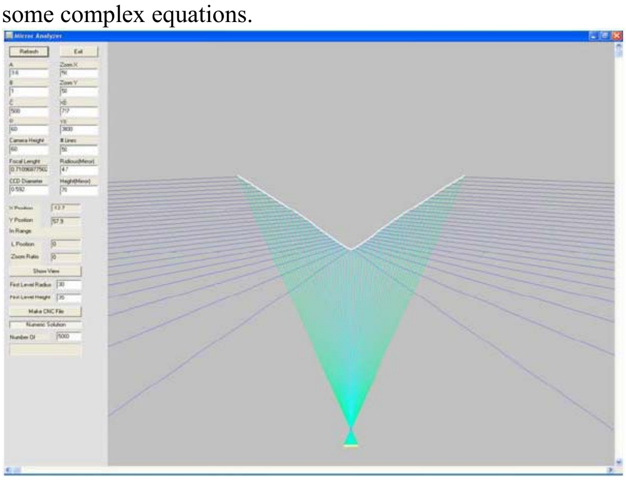

Fig. 6 Cutting section of Omni-Directional mirror with its reflected beams

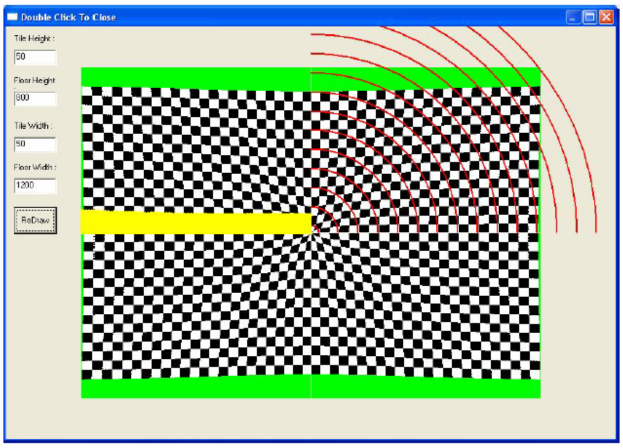

Fig. 7 Simulated result of designed isometric-mirror

The shape of robot will not be shown in the mirror, because the angle of the sharp cone of mirror is designed in that way the robot can see only its environment. To process the gathered images, at first a median filter is applied in order to reduce image noises, and then the eight standard color marks will be assigned to each pixel by the Color Lookup Table. The Color Lookup Table (CLT) is filled in another program, which classifies the YUV Color Space into eight standard colors. This program takes some supervised samples from user and then learned how to recognize the standard colors and save them in CLT.

Whenever the main Image processing program loads, CLT will be loaded from a file. After color mark assigning, image segmentation algorithm will be used to distinguish all of the objects in the field and find their position with the help of their special characteristics. For example this program searches image for red color marks, to find the ball. Then the bulk of red pixels will be found. The segmentation algorithm was learned to find the center of ball from its bulk shape in image. This is done by an artificial neural network. Hence the results will be obtained straight forward and in the minimum time [9].

Obstacle detection is quite simple in this vision system; every object which its color is not neither green nor white is an obstacle and should be avoided. In order to obtain a fast position estimation algorithm, a triangulation-based algorithm was designed for Self Localization of Robot. This algorithm needs only three angles to known points (for example center point of field or corner flags) as input. As the angles in an omni-directional vision system can be estimated with very high precision using only the angles and not the distances gives a good accuracy in this algorithm. Then this estimation results will be composed by the odometeric results from encoders and the exact position of robot will be found [10][11].

VIII. Conclusion

The modularity of the control architecture provides several benefits. The bottom-up development allows one to add functionality incrementally to the robot. Having achieved this framework we have begun to investigate of other AI disciplines. Machine learning techniques can be investigated from case-based reasoning in the planning layer to reinforcement learning in the skill layer [12]. We are going to develop a software tool by using SPADES framework [13] to simulate opponent's soccer robots.

So it will be possible to execute competitions between our real robots and the virtual opponent. Consequently it could help to improve the coordination system of the real robots.

References

[1] C.C De wit.: Trends in Mobile Robot and Vehicle Control. Control Problems in Robotics and Automation. (1998) 151-175.

[2] S.L. Dickerson and B.D. Lapin.: Control of an Omni-Directional Robotic Vehicle with Mecanum Wheels. National Telesystems Conference Proceeding. (1991) 1:323-328.

[3] M.-J. Jung, H.-S. Kim, S. Kim, and J.-H. Kim.: Omni-Directional Mobile Base OK-II. Proceeding of the IEEE International Conference on Robotics and Automation.(2000) 4:3449-345.

[4] Y. Mori, E. Nakano, T. Takahashi, and K. Takayama.: Mechanism and Running Modes of New Omni-Directional Vehicle ODV9. JSME International Journal. (1999) Series C, 42(1): 210-217.

[5] Jong-Suk Choi and Byung Kook Kim, “Near minimum-time direct voltage control algorithms for wheeled mobile robots with current and voltage constraints,” Robotica volume 19,pp. 29-39. 2001.

[6] D. Kim and S,Rhee , “Design of an Optimal Fuzzy Logic control using response surface methodology ” , IEEE Trans. Fuzzy system,vol 9 , pp 413, JUNE 2001

[7] C.S Tesng, B.-S. Chen, and H.-J Uang , “Fuzzy Traking Control fo Nonlinear System via T-S Fuzzy Model ” , IEEE Trans. Fuzzy sysrem,vol 9 , pp 381 , JUNE 2001

[8] Y. Yagi, S. Kawato, S. Tsuji, "Real-time omni-directional image sensor (COPIS) for visionguided navigation", IEEE Trans. on Robotics and Automation, Vol. 10, N. 1, pp. 11- 22, 1994

[9] P. Lima, A. Bonarini, C. Machado, F. M. Marchese, C. Marques, F. Ribeiro, D. G. Sorrenti, “Omni-Directional Catadioptric Vision for Soccer Robots”, Special Issue of the Robotics and Autonomous Systems Journal, Elsevier, Amesterdam, Vol 36, nº 2, 31st Augout 2001. 12. A. Bonarini, P. Aliverti, M. Lucioni, "An omnidirectional sensor for fast tracking for mobile robots", Proc. of the 1999 IEEE Instrumentation and Measurement Technology Conference (IMTC99), IEEE Computer Press, Piscataway, NJ, pp. 151-157

[10] Fabio M. Marchese, Domenico G. Sorrenti, “Omni-directional vision with a multi-part mirror”, Workshop on EuRoboCup’2000, Amsterdam, Netherlands, 28th May – 2nd June, proceedings on CD-Rom, Springer.

[11] MRL TDP ROBOCUP 2005 Osaka for middle size soccer robot .

[12] R. Peter Bonnaso and David Kortenkamp, "An Intelligent agent architecture in hich to pursue robot learning". In proceedings of the MLC_COLT' 94 Robot Learning Workshop, 1994

[13] Patrick Riley, SPADES system for Parallel Agent Discrete Event Simulation User's Guide and Reference Manual for Version 0.91, December 7, 2003. Accessible from http://spades-sim.sourceforge.net