

Design and Implementation an Autonomous Mobile Soccer Robot Based on Omnidirectional Mobility and Modularity

S. Hamidreza Mohades Kasaei and S.Mohammadreza Mohades Kasaei

Abstract—The purpose of this paper is to design and implement a Middle size soccer robot to conform RoboCup MSL league. Due to their advantages, omni-directional mobile robots have found many applications especially in robotic soccer competitions. However, omni directional navigation system, omni-vision system and kicking mechanism in such mobile robots have not ever been combined. This situation brings the idea of a robot with no head direction into existence, a comprehensive omni directional mobile robot. Such a robot can respond more quickly and it would be capable for more sophisticated behaviours with multi-sensor data fusion algorithm for global localization. Despite recent advances, effective control and self-localization methods of omni-directional mobile robots remain as important and challenging issues. For this purpose, we utilize the sensor data fusion method in the control system parameters, self localization and world modelling. Omnidirectional vision sensor of the vision system deals with the image processing and positioning for obstacle avoidance and target tracking. The boundary-following algorithm (BFA) is applied to find the important features of the field. We utilize the sensor data fusion method in the control system parameters, self localization and world modeling. A vision-based self-localization and the conventional odometry systems are used for robust self-localization. The localization algorithm includes filtering, sharing and integration of the data for different types of objects recognized in the environment. In the control strategies, we present three state modes, which include the Attack Strategy, Defense Strategy and Intercept Strategy. The methods have been tested in the many Robocup competition fields of middle size robots. The methods have been tested in the many Robocup competition field middle size robots. Some new interesting methods are described in the current report.

Index Terms—Mobile robot, Machine vision, Omni directional movement, Autonomous Systems, Robot path planning, Object Localization.

I. INTRODUCTION

Robocup was established to promote AI and intelligence robotic research, and its dream is to defeat the FIFA World cup Champion team by a humanoid robot team. In RoboCup 2008, many teams realized team plays as well as developed advanced hardware composed of a powerful kicking device, an omni directional camera, and an omni directional platform. Our project provides six robots of the same type. Our aim is to have a cooperative team behavior while attaining a high individual robot performance in image

processing, decision making and motion control. Some studied subjects in this work are: "Artificial Intelligence includes Fuzzy Decision-Making and Behaviors Learning based on Strategy concept, Advanced Jacobean MIMO Controller, Image processing and high performance motor driver modules". Our robots are equipped with three-wheeled full omni directional navigation system and omni-vision system. We believe that a reliable mechanical structure design, robust pioneer algorithms and high performance sensor fusion are principles of a high team performance in dynamic environments [1].

The Middle Size League team of the ADRO was founded in 2006. In 2007 we ranked 2nd place Middle Size Soccer Robot League in 2nd International Iran-Open Robocup Competitions the Iran-Open is one of the, Asia's major RoboCup event. At the China-Open RoboCup 2007 as well as at the Iran-Open RoboCup 2007 we ranked 2nd place Middle Size Soccer Robot League, in 2008 we achieved the First Place in the 3rd International Iran-Open Competitions. The basis for our success was the robust and reliable hardware design, well-structured software architecture and efficient algorithms for sensor fusion and behavior generation. Our main research interest is both, the development of learning robots and the development of improved sensor fusion and sensor integration techniques. Among others, several different approaches have been investigated so far:

- ❖ Reinforcement learning to learn intercepting a ball.
- ❖ Reinforcement learning to learn dribbling and Path Planning.
- ❖ Development of a computational efficient algorithm for self-localization.
- ❖ Development of estimation procedures for robot and ball velocity
- ❖ Development of a vision system with an omni directional vision sensor and a front camera to recognize ball position and find necessary information in field.

In this paper, we will at first describe the general hard- and software design of the ADRO Robocup Team, (section 2) and after that focus on our scientific approaches in sensor fusion and learning (section 3). Finally, section 4 concludes this paper.

II. HARDWARE ARCHITECTURE

The ADRO Robot (Fig. 1) were designed and completely built in Robotic and AI center from Islamic Azad University-Khorasgan branch. The mechanical structure of

the players is layered and modular. Each layer can easily be replaced by an equivalent one. The components in the lower layer, namely motors, encoders, tow type wheels, batteries and an electromechanical kicker [4], reattached to an aluminum plate placed 6 cm above the floor. The robots' lower layer consists of a three-wheeled movement system. The wheeled movement system has three actuated wheels and each is driven by one DC motor. Having three independently controllable wheels, the robot can rotate or move in either direct or curved path. Using this system, robot can reach a maximum speed of about 2 meters per second. The second layer contains the control electronics. The third layer contains a laptop computer, an omni-directional vision system, a frontal camera and an electronic compass, all close to the maximum height of 80 cm. The players are capable of omni-directional motion, based on three omni-directional roller wheels. Every fully autonomous robot of "ADRO" is equipped with an omni-directional vision system, a normal camera as front vision, and an electromagnetic kicking device. The robot is controlled by a notebook PC is demonstrated in figure 1. The main board uses AVR microcontrollers as a digital processing unit that enables the robot to have some onboard processing. The program for the microcontrollers is written in AVR C language. The omni-directional vision system consists of a hyperbolic mirror, a Color Digital FireWire Camera (Basler Digital Camera) and on each robot we have a standard laptop with 2GHz CPU with Windows Operation System. On this laptop the autonomous behavior of the robot is computed, based on the data gathered from the omni-directional camera.

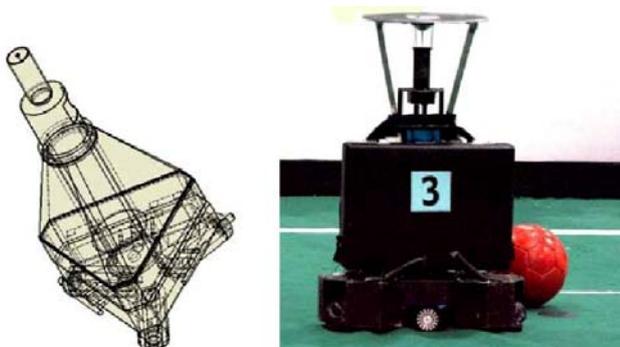


Fig. 1. Our robot is a 3-wheeled omni-directional mobile robot with an omni-directional vision system, a normal camera and a kicking device, and is controlled by a notebook PC

A. Omni directional wheels and robot chassis Figures

Omni directional robots usually use special wheels. These wheels are known as omni directional poly roller wheel. The omni-directional movement system consists of omni-directional wheels, DC motors, a drive shafting system and a controller. The three omni-directional wheels are the key component of the robot, each wheel radially equipped with dozens of small wheels. The three-wheel drive robot can move at any direction and at any moment. Although three such wheels are sufficient for the robot to move omni-directionally, a fourth wheel can provide redundancy in motion and control [2]. Our Robot structure includes three omni directional wheels for motion system and three small free wheels as feedback mechanism where shaft encoders are mounted on, as shown in figure 2.



Fig.2. Three omni directional wheels act as actuators while three free wheels are for feedback.

B. Omni directional vision system

Since the beginning of mobile robot, the map building was one of the most addressed problems by researchers. Several researchers used omni directional vision for robot navigation and map building [3]. Because of the wide field of view in omni directional sensors, the robot does not need to look around using moving parts (cameras or mirrors) or turning the moving parts [4]. The global view offered by omni directional vision is especially suitable for highly dynamic environments. The omni-directional vision system consists of a hyperbolic mirror, a fire wire color digital camera (Basler Digital Camera) and a regulation device. The hyperbolic mirror, design by ourselves. The mirror can make the resolution of the images of the objects near the robot on the field constant and make the distortion of the images of the objects far from the robot small in vertical direction. Searching through different articles and catalogues from various mirror-making companies; we found that they used the following hyperbolic curve for their omni directional vision mirror [5].

$$\frac{x^2}{233.3} - \frac{y^2}{1135.7} = -1$$

However, this equation is suitable for the mirror with large size and wide view. For our soccer player robot, we need an image with a diameter of 4m on the field, so to achieve a compact mirror with wide view, the above curve scaled down by a factor of 2.5 to yields:

$$\frac{x^2}{233.3} - \frac{y^2}{1135.7} = -6.43$$

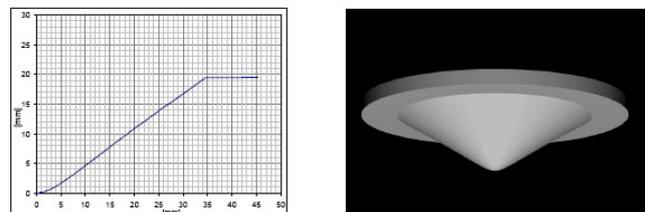


Fig.3. Profile of the overall mirror and the proposed mirror.

Next, a special three stages process was considered for the mirror manufacturing:

- a. Curve fabrication,
- b. Polishing,
- c. Coating.

In the first stage the curve was fabricated on steel 2045 with CNC machining. Then, the work-piece was polished by a special process and finally Ni-P electroless plating was employed.

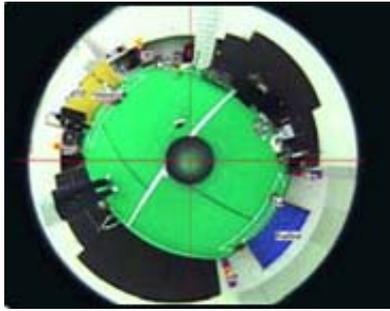


Fig. 4. The typical panoramic image captured.

The regulation device can adjust the height of the omni-directional vision system and the distance between the hyperbolic mirror and the camera.

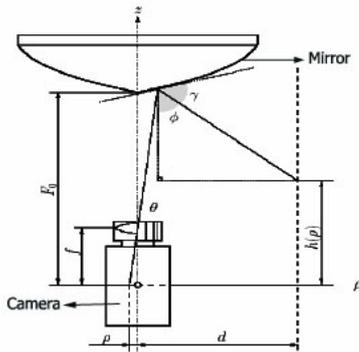


Fig. 5. The Schematic diagram of the omni directional vision system.

C. Ball handler and kicking system

One of the most essential parts of a soccer robot in Middle size and Small size classes in the kicking system, this system is in charge of kicking the ball upon the command issued by the processor of robot. Almost every team develops their own unique shooting device. In this section, we are going to describe designed and developed our multi power kicking system that enables loop and vary shooting power. The principle used for our kicking devices is self-inductance. By sending a current through a turn of wire a magnetic field can be built. As the number of turns or current increases, the magnetic field increases too. With magnetism ferromagnetic materials can be attracted or repulsed. This phenomenon is used in a solenoid. For kicking device the solenoid has to be really fast, because it travels 10cm in about 10ms when shooting at 10m/s. To design a good solenoid and to obtain maximum velocity of ball some parameters like: inductance, response time, resistance, force, dimensions and core-material should be balanced carefully. This design takes advantage of the property that a solenoid has a ferromagnetic core which is attracted into the coil centre. The piece of nylon which is attached to the iron bar is a non-ferro and shoots outwards and hits the ball.

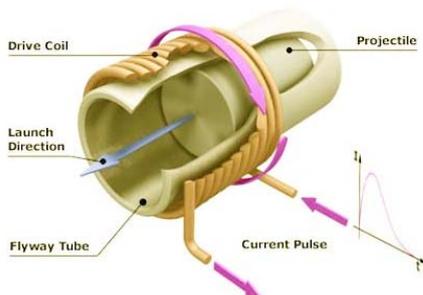


Fig. 6. Solenoid mechanism system.

It is able to shoot very fast, 10m/s when about 800 turns and a current of 60[A] are applied. It is rather small (length is about 20cm and about 5cm diameter) and lightweight (2[kg]). Only a transformer, a capacitor, some resistors and a switch is used so it is in theory very reliable. And most important shooting power can be varied by varying the time of the applied current. The disadvantage of the use of a solenoid is that it operates at a high voltage and current, so it can be quite dangerous. This can be solved by hiding dangerous parts in a black box. It also uses a lot of power for a really short time, so a capacitor is needed to supply high voltage and current. We used a DC-DC converter (Boost regulator) for getting different currents to have different power of shooting.



Fig. 7. Ball handler mechanism.

III. THE SOLENOID CONTROLLING CIRCUIT

Through this circuit, the time needed for first charging of the condenser to 450V, is 13 seconds. This voltage will be about 150v after a stroke with the maximum force. The time need for recharging the capacitor from 100 volt to 450 volt is about 9 seconds. Parallel to the capacitor is the solenoid, modeled with a coil and a resistance with the values of the solenoid. The solenoid is connected to a transistor which is controlled by a pulse source. The transistor can handle high currents (IRG4PC50FD). When the source-signal is high the transistor is closed and solenoid is activated, when low it is open and solenoid is idle. The transistor opens in 380[ns] and closes in 70[ns]. Parallel to the solenoid is a diode with a resistance to "catch" the back-current generated when turning of the solenoid. The solenoid designing for the normal conditions is not able to return to the first place after the stroke, because after imposing and cutting the voltage from Solenoid, its shaft will be stopped in the last possible place. For returning it to its first condition, we should use another mechanism. This returns the solenoid to its first place after the stroke. An easy and cheap solution for the above problem is using a spring at the end of the solenoid.

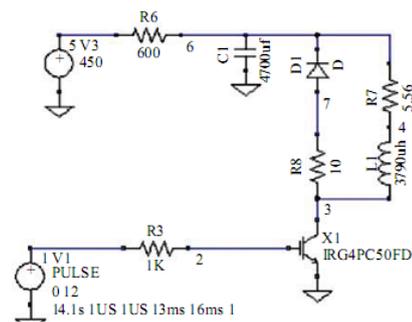


Fig. 8. The solenoid controlling circuit design.

IV. ROBOT SOFTWARE

We have developed a software system to fully utilize the hardware abilities. In this section introduces software parts contain: image processing algorithm, position controller architecture, world model construction, artificial intelligence, trajectory, and network and team strategy from a viewpoint of software system. Three actions are allotted to the robot: attack, support, and defense. The attack is realized through the following process: first, the robot acquires the ball. The robots continuously try to get the ball. Next, the robot face to ball and targeting to opponent goal then dribbles and shoots the ball into the opponent goal. During the dribble, the robot adjusts its direction toward the opponent goal and dribbles with the fastest possible speed. One of the advantages of our robot is a strong kicking device; therefore the robot can shoot a loop-ball into the opponent goal before the opponent defender comes close to the robot. The support robot takes a position behind and near to the robot with the ball. The support robot fetches the ball only when the ball is near to the support robot. The defence robot is located between the ball and own goal. The defence robot doesn't actively approach when the ball is far. In our team strategy three states are allotted to the team robots: attack, defense, and intercept. The robots autonomously choose to activate each of the roles.

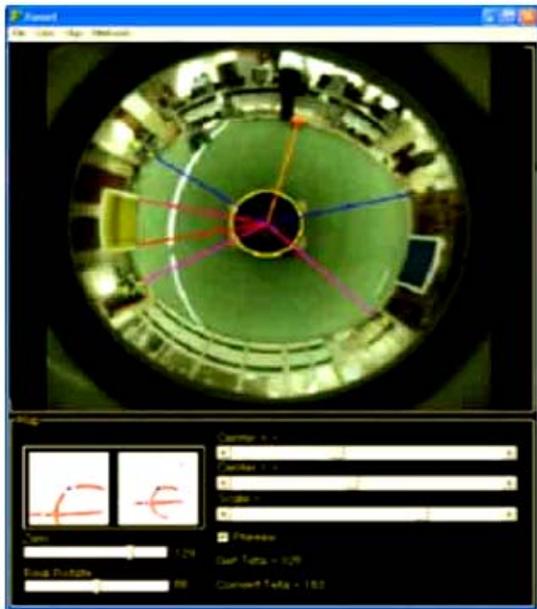


Fig. 9. Vision systems, object detection (goals, flag spots, ball) and line detection for self localization algorithm.

A. Image processing algorithm

Utilizing a digital camera, each time the computer on each robot performs the processing of the current frame and calculates the position, direction and velocity of the robot. It also determines the position and velocity of the opponent robots as well as the position and velocity of the ball. The image-processing algorithm first filters the image by using a table for labeling the colors then recognizes the contiguous regions through either a BFS or a DFS search algorithm and finally extracts the positions by looking in the Image to ground map table. 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 the pixels by swiping them with certain steps, when it finds a desired one and detects that object, saves its coordinates so the next time it can start back with the same point about. Sometimes for better image manipulation the RGB color space is converted to HLS (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 unpredicted conditions. We are trying to evaluate new methods to find some kinds of objects based on pattern recognition to reduce the effect of changing the colors on algorithm. The image processor receives its data through fire wire port connected to a Basler digital video camera with the speed of 20 to 30 frames per second.

B. World model construction

Although each agent tries to extract the real world map as accurate as possible, but “noisy data” and “non-global optimized” algorithms reduce the reliability of processed data. The world model module receives different data sets from every agent. Each data set contains different environmental information like self, ball and opponents' positions. Each data carries a ‘confidence’ factor; a larger confidence factor means a more reliable piece of information. The most recent data sets are then chosen for data fusion, in which the following rules and facts are applied:

- Closer object are of more accuracy.
- Objects further than a specific distance could be said to be totally inaccurate. (This distance is experimentally known)
- An object in the field cannot move faster than an extreme value.

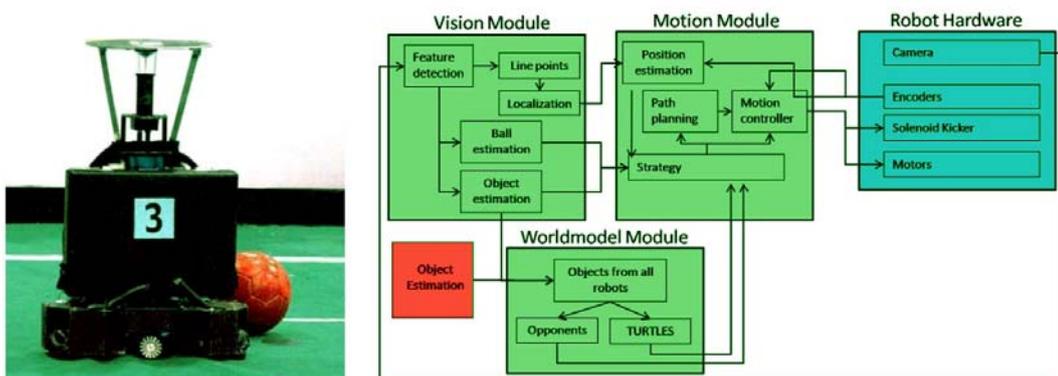


Fig 10. A schematic representation of the robot's hardware and software modules and their interconnections

With respect to the above fact, the module filters unwanted duplicates of objects, (many opponents close to each other seen by different agents), calculates the best approximation for ball and opponents' positions with first order Kalman filtering, gives every object a confidence factor, applies a low pass filter on data and finally constructs a complete world model. This new world model contains information about the objects which may not have been seen by each agent correctly and also enhances approximations of all environmental information. The constructed world model is then sent back to all agents so they will have a better view of the world around them [7].

The interaction between the modules on different machines is provided by a communication protocol which bundles commands and parameters generating command packets and interprets the incoming packets for other modules. In the following, each layer, its interface and parameters will be discussed in details.

C. Artificial intelligence

roles or behaviors. Then AI Core sends a set of roles to Role Engine to be assigned to the robots. Because there are instances in which the image-processing unit cannot see the ball, a memory is implemented in the AI Core for the position of ball that specifies which robot owns the ball. Since there is a relationship between new roles and old roles, roles are changed in a manner that robots never experiment sudden changes in roles (for example the role never changes from defense 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. Twin or triple roles are implemented so that the robots really cooperate with each other to do their roles. Behaviors are the building blocks of the robot's performance which includes simple actions like rotating, or getting the ball and etc. The Behavior Layer is the lowest layer in our architecture. This layer receives a sequence of behaviors along with some parameters from the upper layer (Role Engine) and executes the essential subroutines in order to accomplish a certain behavior. These subroutines use world model information and trajectory data in order to perform necessary movements.

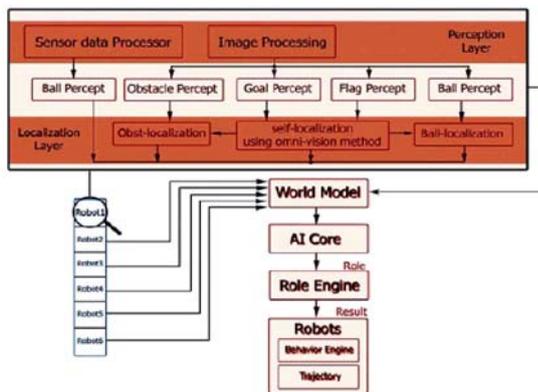


Fig. 11. World model construction and artificial intelligent structure.

D. Position controller architecture

Figure 12 shows the overall block diagram of the control system. This omnidirectional robot control loop contains a PID and a PD controller (with the transfer function PID H

and PD H respectively), a plant transfer function (PH which is obtained from the system dynamics) and a self-localization transfer function (as a feedback function that only senses the robot's position). A noise node, N, is also included that has an additive effect on the system position input. The input of the system is considered to be a step function and the output is the robot position and orientation.

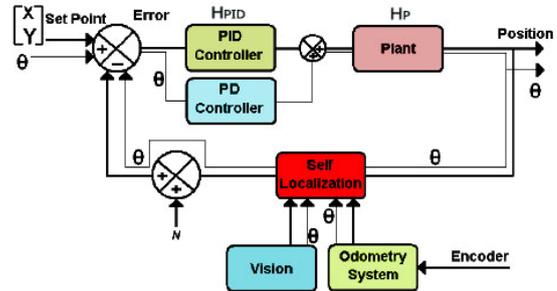


Fig. 12. Control diagram of the fully autonomous robot.

E. Robot Self-Localization

Our self localization method is based on detection of white lines in field. Because according to MSL rule no flag and no color goals exist in field since Robocup 2008, now the white line points are the only visual information that could be used as landmarks for robot's self localization. So our vision system tracks all white lines that exist only in the region field color (green) and robots use a digital compass (MTi-sensor) for the robot heading reference. After this section we try to convert the acquired white line point into the real world distance map.

We employ a Monte Carlo Localization (MCL) method [1]. For our algorithm, the field model is a Cartesian coordinate system with the origin at the center of the field. The robot's state is represented by a vector $X = [x, y, \theta]^T$ which consists of a position (x, y) and θ an orientation. We provided the algorithm, which detects only orientation, made the posture θ ingredient known in MCL, and planned the dimension reduction of the state vector. The orientation detection is explained further below. For localizing, we have to construct the posterior density $p(x_t | y_1 \dots y_t)$ from the state of a robot x_t and the sensor data y_t at the current time t . In the particle filter methods, a probability density is represented by a set of N random samples (Particle). The method proceeds in two phases. In the first phase we predict a current state of the robot. That is specified as a conditional density $p(x_t | x_{t-1}, u_{t-1})$ from the previous state x_{t-1} and a control input u_{t-1} . The predictive density is obtained by the following integral. For our algorithm, we set the control input u_{t-1} as odometry data and add it to each particle.

$$p(x_t | y_1 \dots y_t) = \int p(x_t | x_{t-1}, u_{t-1}) p(x_{t-1} | y_1 \dots y_{t-1}) dx_{t-1}$$

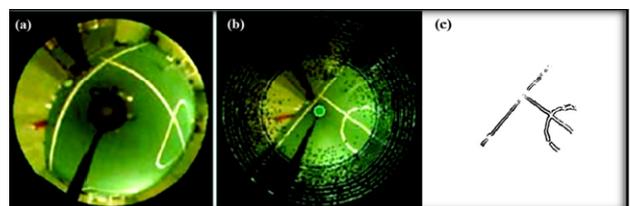


Fig. 13. The field lines detection and self localization.

In the second phase we update the density according to the sensor data y_t . The likelihood of y_t at state x_t is represented as $p(y_t | x_t)$. The posterior density is obtained using Bayes theorem.

$$p(x_t | y_1 \dots y_t) = \frac{p(y_t | x_t) p(x_t | y_1 \dots y_{t-1})}{p(y_t | y_1 \dots y_{t-1})}$$

Sensor data y_t is distance to the field line. The state is compared to y_t and the likelihood is updated of each particle. After that, weighted particles are normalized and re-sampled. Re-sampling is done according to the weight of each particle: new particles are generated around the particles that have high likelihood. In the RoboCup soccer field most constituents are straight lines or perpendicular segments of lines. The robot's orientation is detected by searching for inclination of the straight line ingredients in the circumference seen from the robot. This approach is simple, high-speed, and the derivation of the robot orientation becomes efficient. Our task is to use MCL in the RoboCup environment. It is possible to falsely detect orientation because the form of the field is symmetric against the center of the field. We solved the false detection by using compass sensor for this problem. Our self localization algorithm is a one of the very fast and effective algorithm to track robot's localization, and it only takes several milliseconds to finish the localization computation for one frame image. Experiments show that the position error of robot's self localization can be less than 50 centimeter. ADRO has developed and implemented three separate omni directional wheels coupled with shaft encoders placed 60° apart of the main driving wheels. Free shaft rotation and the flexible connection to the structure ensures minimum slippage and firm contact of these wheels to the ground, all these result in a great improvement in output precision. In order to avoid the remaining cumulative error, odometry system parameters can be initialized every time the vision could calculate the position reliably.

F. Robot Path Planning with Artificial Potential Field

Since the motion trajectory of each robot is divided into several median points that the robot should reach them one by one in a sequence the output obtained after the execution of AI will be a set of position and velocity vectors. 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 (also an alternative new method is in progress which models the robot motion through opponents same as the flowing of a bulk of water through obstacles)[7][8]. In this method different electrical 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 the robot. At a higher level, predictions can be used to anticipate the position of the opponents and make better decisions in order to reach the desired vector. In our path planning algorithm, an artificial potential field is set up in the space; that is, each point in the space is assigned a scalar value. The value at the goal point is set to be 0 and the value of the potential at all other points is positive. The potential at each point has two contributions: a goal force that causes the potential to increase with path distance from the

goal, and an obstacle force that increases in inverse proportion to the distance to the nearest obstacle boundary. In other words, the potential is lowest at the goal, large at points far from the goal, and large at points next to obstacles.

$$U(q) = U_{Goal}(q) + U_{Obstacle}(q)$$

If the potential is suitably defined, then if a robot starts at any point in the space and always moves in the direction of the steepest negative potential slope, then the robot will move towards the goal while avoiding obstacles. The numerical potential field path planner is guaranteed to produce a path even if the start or goal is placed in an obstacle.

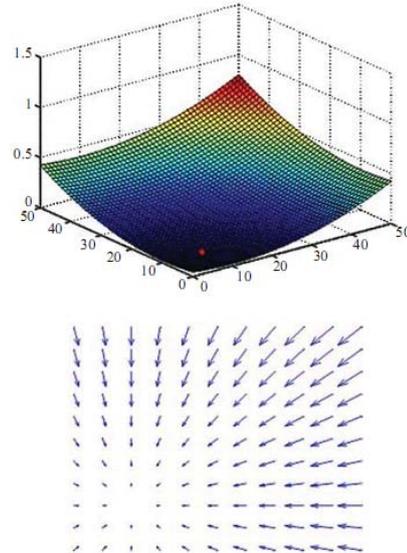


Fig. 14. Goal force (Attractive potential to the goal)

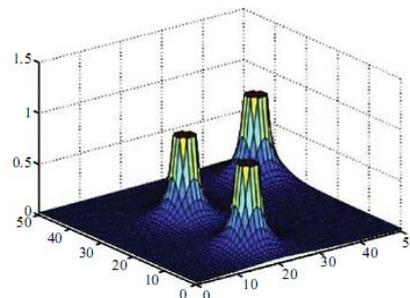


Fig. 15. Obstacle force (Repulsive potential)

If there is no possible way to get from the start to the goal without passing through an obstacle then the path planner will generate a path through the obstacle, although if there is any alternative then the path will do that instead. For this reason it is important to make sure that there is some possible path, although there are ways around this restriction such as returning an error if the potential at the start point is too high. The path is found by moving to the neighboring square with the lowest potential, starting at any point in the space and stopping when the goal is reached.

G. Network

The network physical layer uses the ring topology. The UDP (User Datagram 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, goals and the ball) is transmitted to the server from each client computer of

robots, the server combines them, constructs the complete global localization field then sends the appropriate data and commands (indicating which objects each robot should search for) back to the clients. When the data is completed it is passed to the AI unit for further processing and to decide the next behavior of the robots.

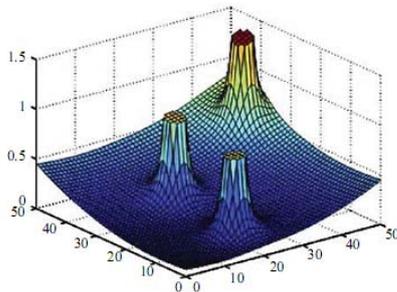


Fig. 16. Goal Force + Obstacle force

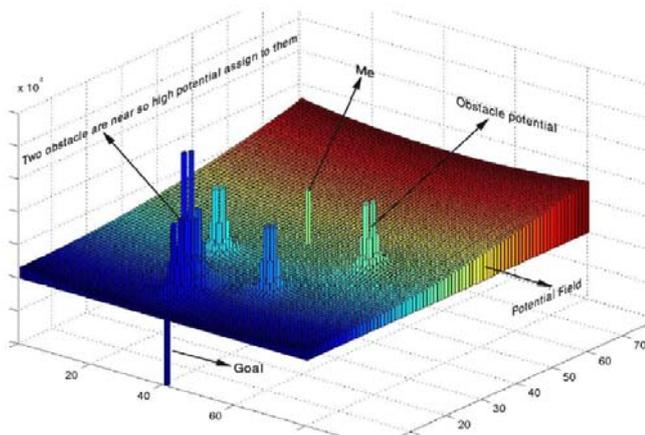


Fig. 17. Potential at every point; it is highest in the obstacles and lowest at the goal. Elsewhere it is generally higher farther from the goal and near obstacles.

V. CURRENT RESEARCH FOCUS

Our current main research focuses are listed as follows:

a. Robust robot vision: The final goal of RoboCup is that the soccer robot team defeats human champion, so robots will have to be able to play outdoors and get rid of the color-coded environment sooner or later. We will go on developing our robot vision system to make that the robot can work well in the environment with highly dynamic illumination and even in totally new field without any off-line calibration. We are also doing further research on the arbitrary FIFA ball recognition method by our omni-directional vision system based on the developments we have achieved [6]. We are also interested in recognize and distinguish the robots belonging to different teams and other generic objects by using the advanced pattern recognition techniques.

b. Multi-robot cooperation:

Multi-robot cooperation holds an important place in distributed AI and robotics field. We have designed a good multi-robot cooperation mechanism and also realized several two-robot cooperative behaviors. Now we have to do deeper research to develop our robot's cooperation ability by involving more robots and more complex cooperative behaviors in this mechanism.

c. Multi-robot cooperative perception:

Multi-robot cooperative perception is another important

research focus in multi-robot system. As the MSL environment be-coming more and more complex, and the field larger and larger, for the field of view of robot is limited, the occlusions of important objects such as ball is very common in dynamic competition, and the inconsistent world model of every robot brought by perception noise will also affect the team's strategy. So we are interested in building the accurate and consistent world model of multi-robot team by cooperative perception such as cooperative object localization and robot's cooperative localization.

VI. CONCLUSION

The performance of our robot team in Iran-Open Robocup competitions 2008 (1st place) showed that the combination of methods and techniques described in this paper are led to a successful soccer player team. In our robot, omni directional navigation system, omni-vision system and a novel kicking mechanism have been combined to create a comprehensive omni directional robot. The idea of separating odometry sensors from the driving wheels was successfully implemented. Three separate omni directional wheels coupled with shaft encoders placed apart of the main driving wheels. The result was reducing errors such as slippage in the time of acceleration. Combination of odometry and vision led to a more accurate and reliable self-localization algorithm. A new hyperbolic with a special coating technique was used to create a wide and compact omni vision mirror.



Fig.18. ADRO Middle size soccer robot team from Islamic Azad University Khorasgan Branch

REFERENCES

- [1] S.H.Kasaei, S.M.Kasaei, S.A.Kasaei, M.Taheri, S.A.Monadjemi " Modeling and Implementation a Fully Autonomous Soccer Robot Based on Omni-directional Vision System", Industrial Robot: an International Journal, Emerald Group[ISSN 0143-991X] 37/3 (2010) 279-286.
- [2] S.H.Kasaei,S.M.Kasaei, S.A.Kasaei,M.Taheri, "Design and Implementation a Fully Autonomous Soccer Player Robot" Proceedings of WASET, Volume 39, ISSN: 2070-3740, Hong Kong, China, March 23-25, 2009.
- [3] S.H.Kasaei, S.M.Kasaei, S.A.Kasaei, M.Taheri, "Effective Mechatronics Models and Methods for Implementation an Autonomous Soccer Robot" 17th IEEE Iranian Conference on Electrical Engineering, Tehran, Iran, May 12-14, 2009.
- [4] S.H. Kasaei, S.M.Kasaei, S.A.kasaei, M.Taheri "Design and Development of Novel Solenoid-Based Kicking System". 9th ICME, Birjand, Iran, March 3-5, 2009.
- [5] Hyun-Sik Shim, Yoon-Gyeong Sung, Seung-Ho Kim and Jong-Hwan Kim, "Design of Action Level in a Hybrid Control Structure for Vision Based Soccer Robot", Proceeding of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp1406-1411.
- [6] Lu, H., Zhang, H., et al.: "Arbitrary Ball Recognition Based on

Omni-directional Vision for Soccer Robots". RoboCup 2008: Robot Soccer World Cup XII, pp. 133-144 (2009).

- [7] S.H.Kasaei, S.M.Kasaei, S.A.Kasaei, M.Taheri, " Modeling and Implementation of Omni-directional Soccer Robot with Wide Vision Scope Applied in Robocup-MSL ", 2nd international Conference on Man-Machine System 2009, 11-13 October 2009, Penang, Malaysia.
- [8] S.H.Kasaei, S.M.Kasaei, S.A.Kasaei, M.Taheri, "path planning for mobile robots Navigation Applied to Dribbling via artificial Potential Field", 1st Iranian Conference on software development, Khorasgan Azad University (Isfahan). 20-23 may 2007, Isfahan, Iran.

Seyed Hamidreza Mohades Kasaei was born in Isfahan, Iran in 1986. He received the B.S. degree in Computer engineering from Khorasgan Islamic Azad University (Isfahan) in 2009, and he continuous educations in M.S. degree in computer engineering field of Artificial Intelligent in University of Isfahan. Currently he is manager of three Robocup teams in Robotic and Artificial Intelligence Center of Khorasgan Islamic Azad University, Isfahan, Iran. His teams work in Middle size soccer robot league, Humanoid soccer robot (Kid size) league and 2D soccer simulation league that they can obtain different ranks in robotic competition. He has published over 40 research papers including one Best Paper and one Innovation Awards from Emeral Group Publication (England). His main research interests are in Artificial Neural Network, Multi agent systems, Agent collaboration, Robotic engineering, Machine vision and Intelligent control systems.