International Journal of Industrial Engineering & Production Research (2010) pp. 163-171

September 2010, Volume 21, Number 3

International Journal of Industrial Engineering & Production Research

ISSN: 2008-4889

http://IJIEPR.iust.ac.ir/

"Technical Note"

Effective Mechatronic Models and Methods for Implementation an Autonomous Soccer Robot

S.M.R. Mohades Kasaei & S.H.R. Mohades Kasaei*

S.Mohammad.R. Mohades Kasaei, *Islamic Azad University Khorasgan Branch - Young Researchers Club*
S.Hamid.R. Mohades Kasaei, *Industrial Consultant, Islamic Azad University Khorasgan Branch - Young Researchers Club*

KEYWORDS

Mobile robot, Machine vision,
Omni directional movement,
Autonomous Systems,
Robot path planning,
Object Localization

ABSTRACT

Omni directional mobile robots have been popularly employed in several applications especially in soccer player robots considered in Robocup competitions. However, Omni directional navigation system, Omni-vision system and solenoid 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 behaviors with multi-sensor data fusion algorithm for global localization base on the data fusion. This paper has tried to focus on the research improvements in the mechanical, electrical and software design of the robots of team ADRO Iran. The main improvements are the world model, the new strategy framework, mechanical structure, Omni-vision sensor for object detection, robot path planning, active ball handling mechanism and the new kicker design, , and other subjects related to mobile robot.

1. Introduction

Robotic soccer is nowadays a popular research domain in the area of multi robot systems. RoboCup is an international joint project to promote artificial intelligence, robotics and related fields that includes several leagues, each one with a different approach, some only at software level, others at hardware, with single or multiple agents, cooperative or competitive [1].

In the context of RoboCup, the Middle Size League (MSL) is one of the most challenging. In this league, each team is composed of up to 5 robots with maximum size of 50x50cm base, 80cm height and a maximum weight of 40Kg, playing in a field of 18x12m. The rules of the game are similar to the official FIFA rules, with required changes to adapt for the playing robots [2]. Each robot is autonomous and has its own sensorial means. They can communicate among them, and with an external computer acting as a coach, through a wireless network. This coach computer cannot have any sensor; it only knows what is reported by the playing robots. The agents should be able to evaluate the state of the world and make decisions suitable to fulfil the cooperative team objective.

ADRO Robocup Team, is the Middle Size League Robotic Soccer team from Islamic Azad University, Khorasgan Branch (Isfahan) .The project started in 2006, coordinated by the Electrical and Computer Department and involves many student working on several areas for building the mechanical structure of

* **Corresponding author. S.Hamid.R. Mohades Kasaei**
***Email**: H.Kasaee@khuisf.ac.ir*
Paper first received March, 12, 2009, and in revised form August. 03. 2010.

the robot, its hardware architecture and controllers and the software development in areas such as image analysis and processing, sensor and information fusion, reasoning and control.
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 behaviour generation. Our main research interest is both, the development of learning robots and the development of improved sensor fusion and sensor integration techniques. So far several different approaches have been investigated:

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.

This paper is organized as follow. Sections 2 describe the hardware architecture. The perception technology of omni directional vision system, wheels and kicking mechanism is described in this section. Robot software is presented in section 3. The image processing algorithm, position control, artificial intelligence, World model construction, Trajectory and network in our robot, AI Core, Role Engine and Behavior Engine are explained in robot software section. Finally, section 4 concludes this paper.

2. Hardware Architecture

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 chassis of the robot is designed as a frame construction where there is the electronic circuit board, batteries, kicking device, motor controller and notebook PC. The omni-directional vision system and the normal camera are on the top of the framework.

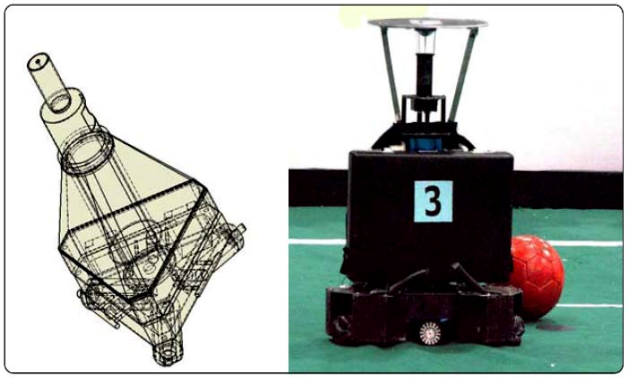

Fig. 1. Our robot is a 3-wheeled omni-directional mobile robot with an omni-directional vision system, a normal camera and a kicking device, and is controlled by a notebook PC.

2.1. Omni Directional Wheels and Kicking System

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 [7][12]. Our Robot structure.

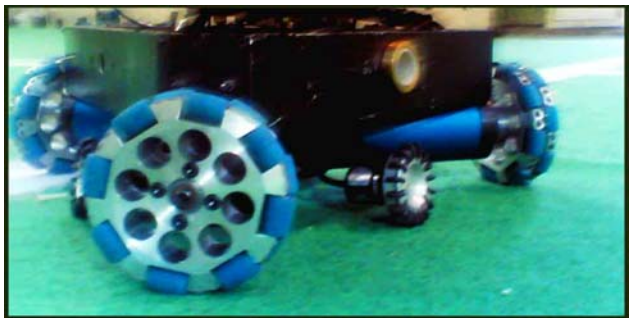

Fig.2. Three omni directional wheels act as actuators while three free wheels are for feedback.

include 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. In soccer player robot usually one direction is used for directing the ball to the goal or other destinations. Therefore, the ball handler and kicking mechanism is added in this direction which help the robot to get a suitable form for directing the ball. Although the conventional mechanism can work for any soccer robot, it has some limitations which cause additional movements especially in rotation of the robot. In other words, each robot has a special head for kicking the ball and must adjust it to the proper direction during the game. These adjustments in the

single head robot increase its rotation significantly and reduce its manoeuvrability. In order to optimize such rotations, we use two extra kicking mechanisms to form an Omni directional kicking system.

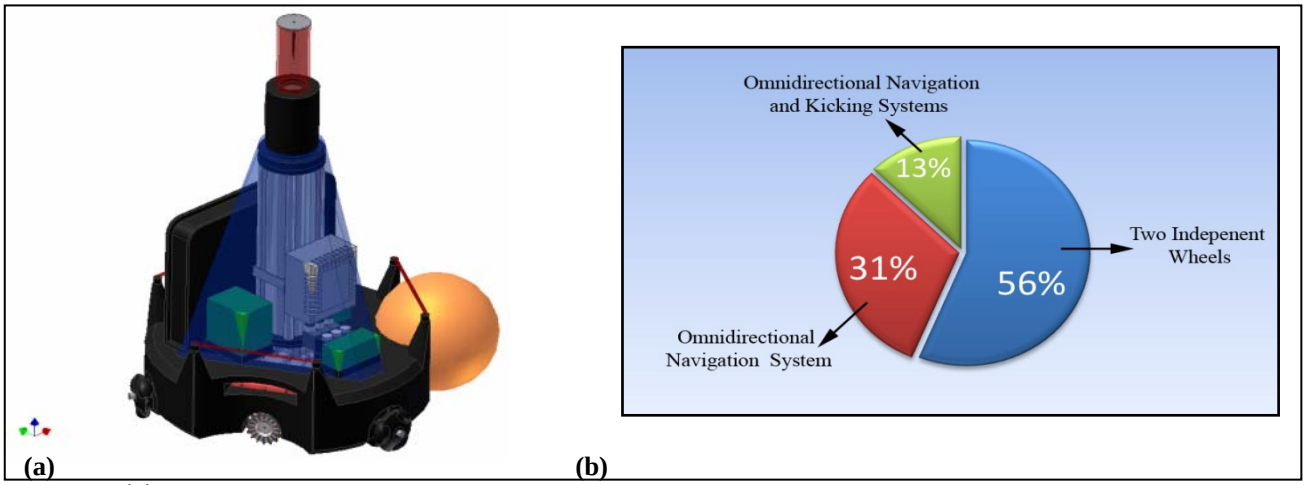

Fig. 3. (a) Omni directional kicking system of the robot, (b) Average rotation rate for three types of soccer player robots

The position of these kicking systems is shown in figure 3(a). As it can be seen from the figure, each kicker is assembled between two omni directional wheels and forms a system with three heads in 0, 120 and 240 angles. In an experiment carried out at our robotic centre, three types of soccer player robot in the form of 10 teams, which participated in Robocup, was examined in order to assess the rotation rate of each type. In this assessment, the number of complete rotations for each robot in one minute was measured and the total average of them was then calculated. Figure 3(b) shows these numbers for three types of robots, i.e.:

Two independent driving wheels mechanism
Omni directional navigation system (one head)
Omni directional navigation and kicking system (three heads)

There is a significant decrease in the rotation rate between the first type and the third type of these robots shown in Figure 3(b). The number of complete rotation per minute for Omni directional single head and three heads robot are 5 and 2 respectively. Therefore, using the Omni-kick system is very useful and can reduce the rotation rate in robots with the same navigation system about 60% percent. The low rotation rate in a robot also simplifies the algorithm needed for following a trajectory, cooperative behaviours and increases the speed and flexibility for directing the ball to the favourite destination.

2.2. 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 firewire colour 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, as demonstrated in figure 4. 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 [6].

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

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:

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

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

Curve fabrication,
Polishing,
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 electro less plating was employed. The regulation device can adjust the height of the Omni-directional vision system

and the distance between the hyperbolic mirror and the camera.

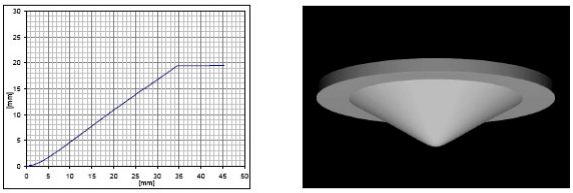

Fig. 4. Profile of the overall mirror and the proposed mirror and The hyperbolic mirror and the typical panoramic image captured.

2.3. Ball Handler and Kicking System

One of the most essential parts of a soccer robot in small to middle 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 trough a turn of wire a magnetic field can be build. 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. 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.

3. Robot Kinematics

Using omni directional wheels, the schematic view of robot kinematics can be shown as follows (figure 5) [5], where O is the robot center of mass, P_O is defined as the vector connecting O to the origin and D is the drive direction vector of each wheel. Using unitary rotation matrix, $R(\theta)$ defined as:

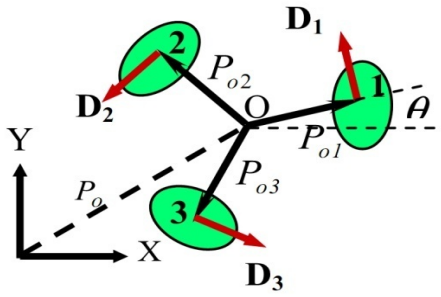

Fig. 5. Robot kinematics diagram

$$R(\theta)=\begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix} \tag{3}$$

The positions vectors $P_{O1},...,P_{O3}$ with respect to the local coordinates centered at the robot center of mass are given as:

$$P_{O1}=L\begin{bmatrix}1\\0\end{bmatrix} \tag{4}$$

$$P_{O2}=R(\frac{2\pi}{3})*P_{O1}=\frac{L}{2}\begin{bmatrix}-1\\\sqrt{3}\end{bmatrix} \tag{5}$$

$$P_{O3}=R(\frac{4\pi}{3})*P_{O1}=-\frac{L}{2}\begin{bmatrix}1\\\sqrt{3}\end{bmatrix} \tag{6}$$

The drive directions can be obtained by:

$$D_i=\frac{1}{L}R(\theta)*P_{Oi}, \tag{7}$$

$$D_1=\begin{bmatrix}0\\1\end{bmatrix},\quad D_2=\begin{bmatrix}1\\0\end{bmatrix},\quad D_3=\frac{1}{2}\begin{bmatrix}\sqrt{3}\\-1\end{bmatrix}, \tag{8}$$

Where, L is the distance of wheels from the robot center of mass (O). Using the above notations, the wheel position and velocity vectors can be expressed with the use of rotation matrix $R(\theta)$ as:

$$\begin{aligned}R_i&=P_O+R(\theta)*P_{Oi},\\ V_i&=\dot{P}_O+\dot{R}(\theta)*P_{Oi},\end{aligned} \tag{9}$$

The vector $P_O=\begin{bmatrix}X & Y^T\end{bmatrix}$ is the position of the center of mass with respect to Cartesian coordinates. The angular velocity of each wheel can be expressed as:

$$\phi_i=\frac{1}{r}V_i^T*R(\theta)*D_i \tag{10}$$

Where, r is the system wheel radius of odometry. Substituting for V_i from equation (9) yields:

$$\phi_i=\frac{1}{r}\left[P_O^T*R(\theta)*D_i+P_{Oi}*\dot{R}(\theta)^T*R(\theta)*D_i\right] \tag{11}$$

Note that the second term in the right hand side is the tangential velocity of the wheel. On the other hand, this tangential velocity is equal to:

$$L\dot{\theta} = P_{Oi}{}^{T} * \dot{R}(\theta) * R(\theta) * D_i \tag{12}$$

From the kinematics model of the robot, it is clear that the wheel velocity is a function of linear and angular velocities of robot center of mass, i.e.:

$$\begin{Bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{Bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3}-\theta) & -\cos(\frac{\pi}{3}-\theta) & L \\ \sin(\frac{\pi}{3}+\theta) & -\cos(\frac{\pi}{3}+\theta) & L \end{bmatrix} \begin{Bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{Bmatrix} \tag{13}$$

or in short:

$$\dot{\varphi} = \frac{1}{r} W * \dot{S} \tag{14}$$

where L is the distance of wheels from the robot center of gravity (O) and r is the main wheel radius.

4. Robot Dynamics

Linear and angular momentum balance for the robot may be written as:

$$\sum_{i=1}^{3} f_i R(\theta) * D_i = m \ddot{P}_O, \qquad L\sum_{i=1}^{3} f_i = J\ddot{\theta} \tag{15}$$

where $\ddot{P}_O$ is the acceleration vector, f_i if is the magnitude of the force produced by the *i*th motor, *m* is the mass of the robot and *J* is its moment of inertia about its center of gravity. Assuming no-slip condition, the force generated by a DC motor is described by:

$$f = \alpha U + \beta V \tag{16}$$

where, $V = \{V_i(t), i = 1,2,3\}$ is the velocity of each wheel. The constants α and β are motor characteristic coefficients and can be determined either from experiments or from motor catalogue.
Note that $U = \{U_i(t), i = 1,2,3\}$ is the voltage applied by supplier to the DC motors. Substituting equation (13) into equation (12) yields:

$$\sum_{i=1}^{3} (\alpha U_i - \beta V_i) R(\theta) D_i = m\ddot{P}_O, \qquad L\sum_{i=1}^{3} (\alpha U_i - \beta V_i) = J\ddot{\theta} \tag{17}$$

This system of differential equations may be written in the matrix form as:

$$\begin{bmatrix} m\ddot{X} \\ m\ddot{Y} \\ m\ddot{\theta} \end{bmatrix} = \alpha P(\theta) U - \frac{3\beta}{2} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 2L^2\dot{\theta} \end{bmatrix} \tag{18}$$

$$P(\theta) = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3}-\theta) & -\cos(\frac{\pi}{3}-\theta) & L \\ \sin(\frac{\pi}{3}+\theta) & -\cos(\frac{\pi}{3}+\theta) & L \end{bmatrix}^T \tag{19}$$

and

$$U = \begin{bmatrix} U_1(t) & U_2(t) & U_3(t)^T \end{bmatrix} \tag{20}$$

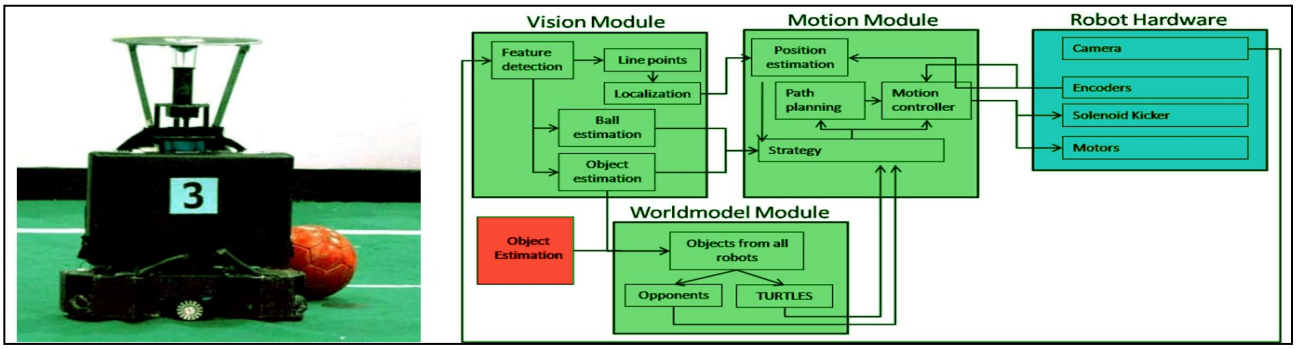

Fig. 6. Block diagram of the Robot Software

5. 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 defence. 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, defence, and intercept. The robots autonomously choose to activate each of the roles.

5.1. 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 labelling the colours then recognizes the contiguous regions through either a BFS (Breadth-First Search) or a DFS (Depth-First Search) 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 colour, a combination of conditions on Hue, Saturation and RGB is used. This procedure makes the colour 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 colours 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.

5.2. 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.

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 Line detection for Self Localization algorithm.

Fig. 7. Vision Systems, Object detection (Goals, Flag Spots, Ball)

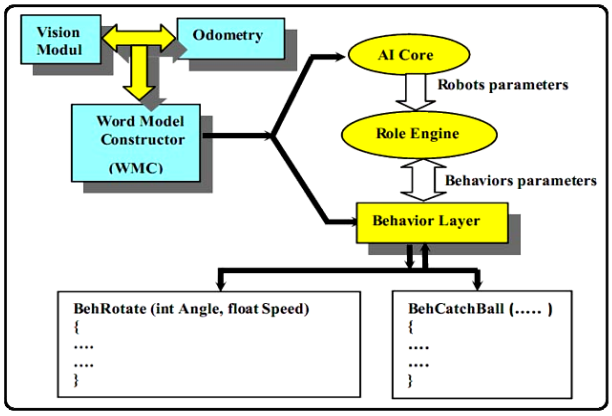

Fig. 8. World model construction and artificial intelligent structure.

and 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!

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.

5.3. Artificial Intelligence

In this section the AI part of the software is briefly introduced. There are three distinct layers: AI Core, Role Engine and Behaviour Engine. AI Core receives the computed field data from world modelling unit and determines the play state according to the ball, opponents and our robots positions. Considering the current game strategy, determination of the play state is done by fuzzy decision-making to avoid undesirable and sudden changes of roles or behaviours. 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 defence to attack in next cycle). Role Engine receives a set of roles from AI Core and provides the Behaviour Engine with a set of behaviours for robots. Twin or triple roles are implemented so that the robots really cooperate with each other to do their roles. Behaviours are the building blocks of the robot's performance which includes simple actions like rotating, or getting the ball and etc. The Behaviour Layer is the lowest layer in our architecture. This layer receives a sequence of behaviours along with some parameters from the upper layer (Role Engine) and executes the essential subroutines in order to accomplish certain behaviour. These subroutines use world model information and trajectory data in order to perform necessary movements.

5.4. 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 colour 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 colour (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.

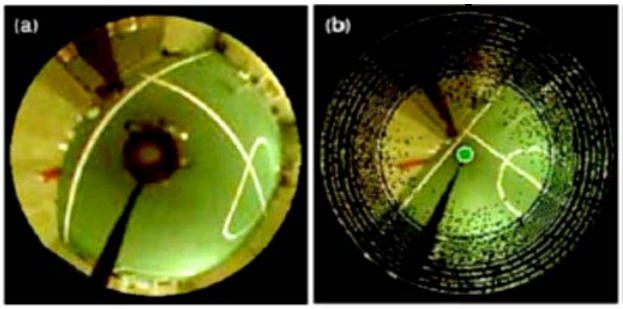

Fig. 9. The field lines detection and self localization

We employ a Monte Carlo Localization (MCL) method [11]. For our algorithm, the field model is a Cartesian coordinate system with the origin at the centre 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 $p(x_t \mid y_1 \ldots y_t)$ 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 $p(x_t \mid x_{t-1} \ldots u_{t-1})$ 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.

$$p(x_t \mid y_1 \ldots y_t) = \int p(x_t \mid x_{t-1}, u_{t-1}) p(x_{t-1} \mid y_1 \ldots y_{t-1}) dx_{t-1} \quad (21)$$

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 $p(y_t \mid x_t)$. The posterior density is obtained using Bayes theorem.

$$p(x_t \mid y_1 \ldots y_t) = \frac{p(y_t \mid x_t) p(x_t \mid y_1 \ldots y_{t-1})}{p(y_t \mid y_1 \ldots y_{t-1})} \quad (22)$$

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 Robocop environment. It is possible to falsely detect orientation because the form of the field is symmetric against the centre 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 centimetre. 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 [9].

5.5. Trajectory

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) [10] [13]. 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.

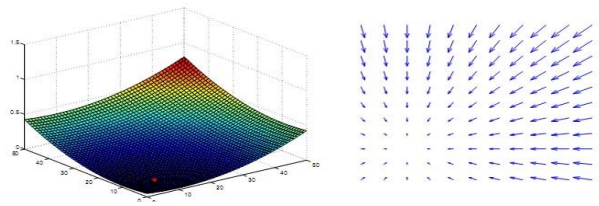

Fig. 10. Goal force (Attractive potential to the goal)

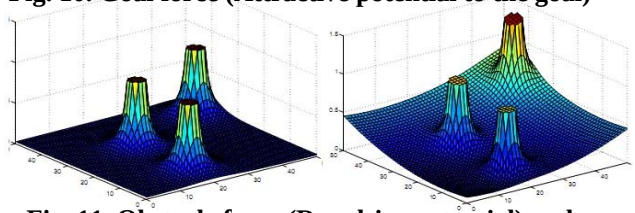

Fig. 11. Obstacle force (Repulsive potential) and Goal Force+ Obstacle force

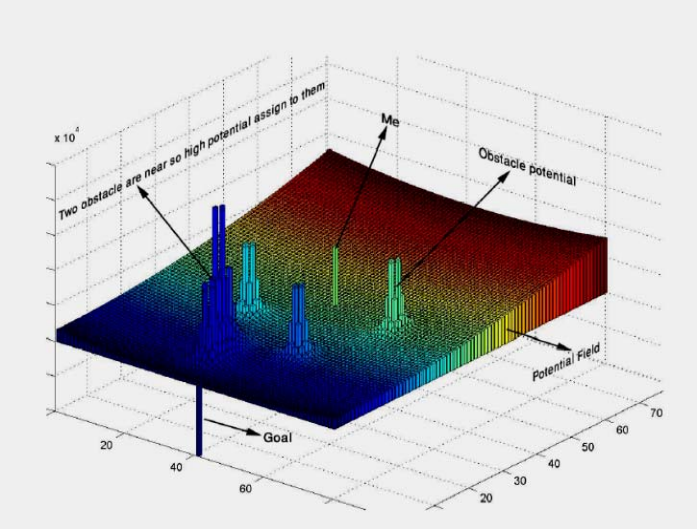

Fig. 12. Potential at every point; it is highest in the obstacles and lowest at the goal. Elsewhere it is generally higher farther from the goal and near obstacles.

In other words, the potential is lowest at the goal, large at points far from the goal, and large at points next to obstacles. 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. 13. ADRO middle size soccer robot team from Islamic Azad University Khorasgan Branch

$$U(q) = U_{Goal}(q) + U_{Obstacle}(q) \qquad (23)$$

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 neigh boring square with the

lowest potential, starting at any point in the space and stopping when the goal is reached.

5.6. 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 behaviour of the robots.

6. 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.

References

[1] Mohades Kasaei, S.H., Mohades Kasaei, S.M., Mohades Kasaei, S.A., Taheri, M., Monadjemi, S.A., *"Modeling And Implementation A Fully Autonomous Soccer Robot Based On Omni-Directional Vision System,"* Industrial Robot: an International Journal, Emerald Group Publishing Limited, 2010, pp 279–286.

[2] Mohades Kasaei, S.H., Mohades Kasaei, S.M., Mohades Kasaei, S.A., Taheri, M., Rahimi, M., Vahiddastgerdi, H., Saeidinezhad, M., "*Design and Implementation a Fully Autonomous Soccer Player Robot,*" Proceedings of World Academy of Science, Engineering and Technology, Vol. 39, ISSN: 2070-3740, Hong Kong, China, March 23-25, 2009.

[3] Mohades Kasaei, S.H., Mohades Kasaei, S.M., Mohades Kasaei, S.A., Taheri, M., Vahiddastgerdi, H., Rahimi, M., "*Effective Mechatronic Models and Methods for Implementation an Autonomous Soccer Robot*," 17th IEEE Iranian Conf. on Electrical Engineering , Tehran, Iran, May 12-14, 2009.

[4] Mohades Kasaei, S.H., Mohades Kasaei, S.M., Mohades Kasaei, S.A., Taheri, M., Vahiddastgerdi, H., "*Autonomous Navigation Systems for Mobile Robot Based on Omni Directional Vision System*," Proceedings of 12th Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines (Clawar), Istanbul, Turkey, 9-11 Sept.2009

[5] Kalmar-Nagy, T.; Ganguly, P., D.Andea, R., *"Real-Time Trajectory Generation for Omni Directional Vehicles*," Proceeding of the Americal control conference, Anchorage, AK, USA, May 2002, pp. 285-291.

[6] Taheri, M., Mohades Kasaei, S.H., Mohades Kasaei, S.M., Vahiddastjerdi, H., Rahimi, M., Monajemi, S.A., *"Design and Development of an Autonomous Robot for Environmental Dynamic Monitoring*," 9th Iranian Conf. on Manufacturing Engineering (ICME), Birjand, Iran, March 3-5, 2009.

[7] Alexander Gloye, Ra´ul Rojas. *"Robot Heal Thyself: Precise and Fault-Tolerant Control of Imprecise or Malfunctioning Robots"*. RoboCup 2005 International Symposium, Osaka, japan, 2005.

[8] Mohades Kasaei, S.H., Mohades Kasaei, S.M., Mohades Kasaei, S.A., *"Development a Real Time Cooperative Behavior Approach for Autonomous Soccer Robots Applied in Robocup- MSL,"* International Journal of Robotics and Automation (IJRA), Malaysia, 2010, pp. 26-41.

[9] Yagi, *"Omni directional Sensing And its Applications*," IEICE TRANS, INF. & SYST. Vol. E82-D, No. 3, 2001, pp. 568-579.

[10] Taheri, M., Mohades Kasaei, S.H., Mohades Kasaei, S.M., Vahiddastjerdi, H., Rahimi M., Monajemi, S.A., *"Design and Development of an Autonomous Robot for Environmental Dynamic Monitoring*," 9th Iranian Conference on Manufacturing Engineering (ICME), Birjand, Iran, 2009.

[11] Tesng, C.S., Chen, B.S., Uang, H.J., "*Fuzzy Traking Control for Nonlinear System via T-S Fuzzy Model*," IEEE Trans. Fuzzy system, Vol. 9 , 2001, pp. 381.

[12] Yagi, Y., Kawato, S., Tsuji, S., *"Real-Time Omni-Directional Image Sensor (COPIS) for Vision Guided Navigation,"* IEEE Trans. on Robotics and Automation, 1994, pp. 11-22.

[13] Mohades Kasaei, S.H., Mohades Kasaei, S.M., Taheri, M., Vahiddastjerdi, H., *"path planning for mobile robots Navigation Applied to Dribbling via artificial Potential Field,"* Iranian Conference on software development, Khorasgan Azad University (Isfahan). 2008.