A Hybrid Neural Network Controller for Stable Walking of a Humanoid Soccer Robot

S.H. Kasaei, S.M. Kasaei, S.A. Kasaei
Islamic Azad University Khorasgan Branch - Young Researchers Club

ABSTRACT: This paper describes the hardware and software design of the kid size humanoid robot systems of the PERSIA Team in 2009. This year, our developments for the humanoid robot is include: the design and construction of our new humanoid robots structure and implementation a new recurrent hybrid neural network for walking control. The control system consists of two neural network controllers, two standard PD controllers and robot walking planar. The proposed neural network controller has three layers, which are input, hidden and output layers.
The project is described in two main parts: Hardware and Software. The hardware section consists of the mechanical structure and the driver circuit board that Each robot is able to walk, fast walk, autonomously get up, kick and dribble when it catches the ball and The software is developed a robot application which consists of motion controller, autonomous motion robot, self localization based on vision system, AI, Trajectory planning and Network. The project is still in progress and some new interesting methods are described in the current report.
KEYWORDS: Humanoid Robot, Neural Network Controller, Intelligent Control, Biped Robot, Robot Locomotion

Introduction

The RoboCup scenario of soccer playing legged robots represents an extraordinary challenge for the design, control and stability of autonomous bipedal and quadrupedal robots. In a game, fast, goal oriented motions must be planned autonomously and implemented online while preserving the robot's postural stability and adapting them in real-time to the quickly changing environment.

In the humanoid league, many technology issues and scientific areas must be integrated to implement the humanoid robot, such as mechanics, electronics, control, computer science, and semiconductor. Besides, the research technologies of humanoid walking control, autonomous motion, vision and AI system, kicking and shooting ball will be applied [MMK00, MMM10, MMT10a, MMT10b, M+09a, M+09b, M+10a, M+10b, M+10c, PK99].

The Humanoid League team of the PERSIA was founded in 2008. In 2008 we ranked 1st place Humanoid Kid Size Soccer Robot League in 1st national Khwarizmi Robotic Competitions the Khwarizmi Robotic is one of the, Iranian major Robotic event. In 2009 we achieved the 5th in the 3rd International Iran-Open Competitions in the 3-3 games out of 15 teams. 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. In order to let the robot can autonomous play a soccer game, three basic skills are designed and implemented on it: image understanding for environment perception, move ability, and artificial intelligence. In order to let the robot have a high ability of environmental detection, a camera and array of sensor are equipped on the body of the implemented robot to obtain the information of the environment to decide an appropriate action. We used available personal digital assistant (PDA) for processing vision and higher level reasoning and brain of our robot. A control board with an ATMEGA128 microcontroller is mainly utilized to control the robot. Many functions are implemented on this system so that it can receive the vision signal obtained by the camera via a CF port and process the data obtained by gyroscopes and the digital compass. It also can process the high level artificial intelligence, such as the navigation. The humanoid robot is designed as a soccer player so that the implemented robot can walk, turn, and shoot the ball.

In this paper, we will at first describe the general hardware design of the PERSIA Humanoid Robocup Team, (section 1) and after that focus on our scientific approaches in software such as robot motion controller, sensor fusion, vision system and learning (section 2). Finally, section 3 concludes this paper. This document describes the current state of the project as well as the intended development for the RoboCup 2010 competition.

1 Mechanical Design

Figure 1 show our humanoid soccer robot, Persia robot includes motion mechanism, Shooting and dribbling mechanism. PERSIA kid-size humanoids are built with aluminum brackets. The kinematics chains are powered by high-torque servomotors. The actuators used in the PERSIA robots are the Hitec servo motor. The motion mechanism consists of 20 degrees of freedom distributed in 5 per leg, 3 per arm and other two degrees of freedom as a pan-tilt system holding the head and tow in trunk, we use an aluminum mechanism powered by two micro servomotors directly controlled by the camera, providing object tracking independently from leg or arm motions. To facilitate exchange of the players, all robots use mechanically the same structure.

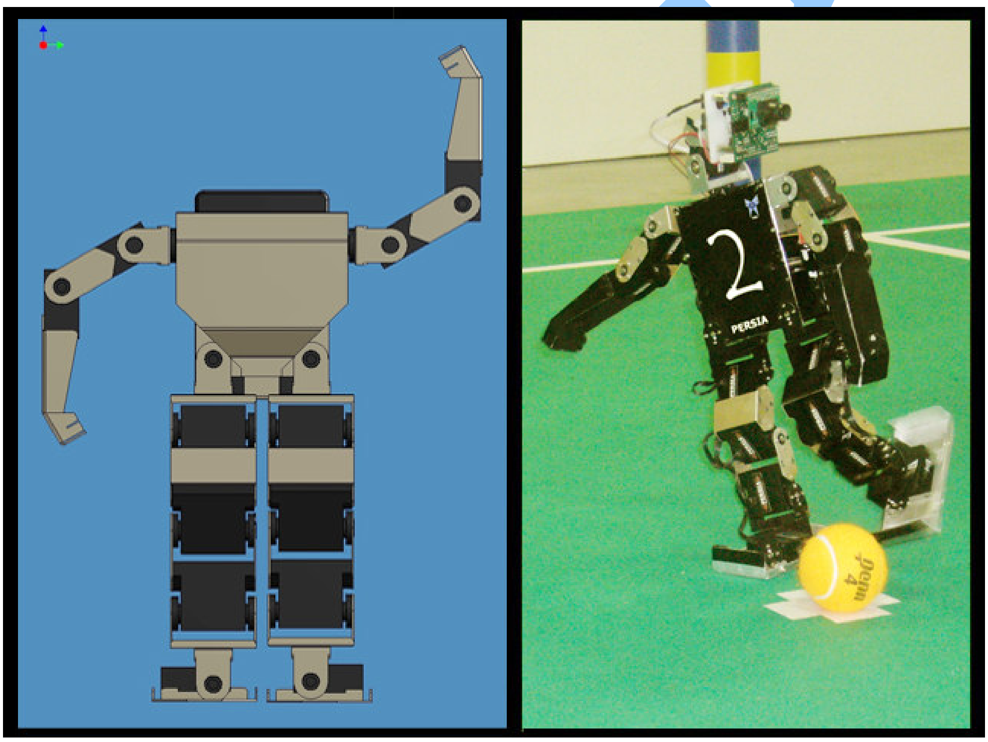

Figure 1. Mechanical construction of the PERSIA humanoid robots

The PERSIA Humanoid robot designed for has multipurpose capability. This robot equipped with main board for motion control, vision sensor, other balancing sensors, servo motors and etc. Our robots perform high mobility and stability. The maximum speed is approximately 0.25m/s.

It can also walk to any direction and targeting smoothly. For stable walking, we use an acceleration and gyro sensors. The acceleration sensor is also used to detect falls. In our robots we use two processors; one is used for motion control and receives data from gyro and acceleration sensors via A/D converter, and another is used for image recognition, behavior determination and so on.

2 Robot Software

In soccer game, for example, the robot searches a ball and two goals, and moves to its desired location with avoiding many obstacles, therefore we developed our humanoid robot software in Visual C++ and robot software is divided into four sections: vision, artificial intelligent, behavior engine, and motion control (actuation).

Figure 2 shows the block diagram of the software which runs in the robot's main processor. The program consists of 4 main blocks:

- **Hardware Interface**: Contains all low level routines to access hardware of the robot including sensors and actuators.
- **Vision**: Contains image processing algorithms such as recognition of landmarks and other object. Self localization is done using particle filtering. Particles are scored by comparing a simulated image from each particle with the current frame captured by camera. Using "Sampling-Importance Resampling" method, a new distribution of the particles is created after each step. Particles are also updated using a motion model. Final distribution of the particles converges to the real pose of the robot.
- **Planning**: Planning system of the robot is based on a multi layer, and multi thread structure. The layers are named Strategy, Role, Behavior and Motion. Each layer contains a Scenario which runs in parallel with the scenarios in the other layers. A scenario in a higher level can terminate and change the scenario running in the lower level; however it is usually done in synchronization with the lower level scenario to avoid conflicts and instabilities. (Such as stopping the walking motion while one of the feet is still in the air).
- **Network**: Mainly responsible for the wireless communication of the robot with the other robots or the referee box. This is done via WLAN.

- **Motion Control**: manages all the actuators of the robot, and controls locomotion or any other action of the robot according to the requests from Cognition.
- **Sensor Control**: manages other sensors, and interacts with the Sub-Controller.

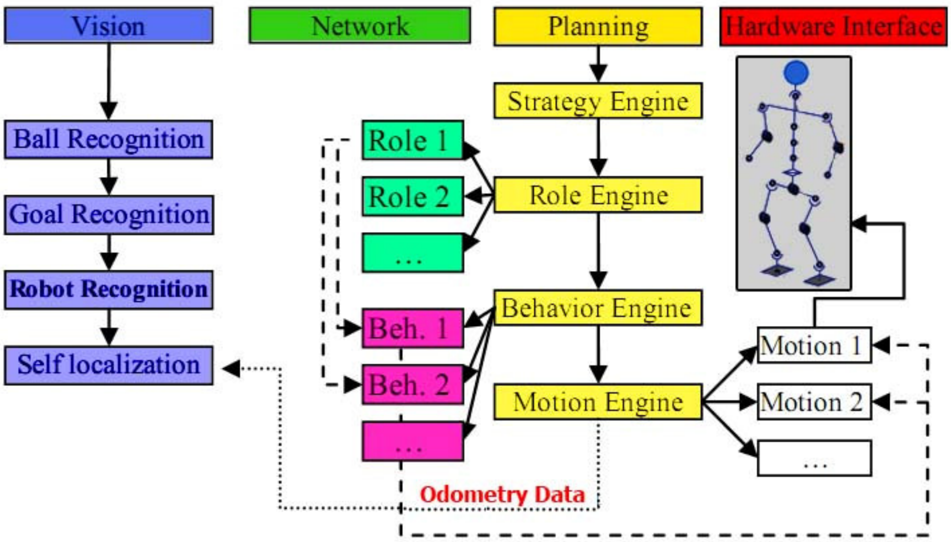

Figure 2. Block diagram of the software which runs in the robot's main processor

3 Motion Control via Artificial Neural Network

The neural network technique is a very effective tool for controlling complex non-linear systems when we have no complete model information, or even when considering a controlled plant as a black box. The use of a proposed recurrent hybrid neural network to control of walking robot is investigated in this section. The reason to use hybrid layer is that robot's dynamics consists of linear and non-linear parts. PERSIA Humanoid Robot has two kinds of locomotion pattern: Omni directional walking and special actions such as kicking and getup. Special actions are described using key frames, which can be edited very fast by our software tool ***HumanoidRobotControl***. Omni directional walking means the robot can walk in every direction with variable step length. The behavior module determines the target position and orientation according to the results of localization and the sensor measurements, and then constructs an action series which consists of the elementary gaits to realize Omni directional walking. A neural networks based control system is utilized to the control of walking robot. The control system consists of two proposed neural controllers, two standard PD controllers and two legged planar. The proposed neural network (NN) is employed as an inverse controller of the robot. The NN has three layers, which are input, hidden and output layers. In addition to feed forward connections from the input layer to the hidden layer and from the hidden layer to the output layer, there is also feedback connection from the output layer to the hidden layer and from the hidden layer to itself.

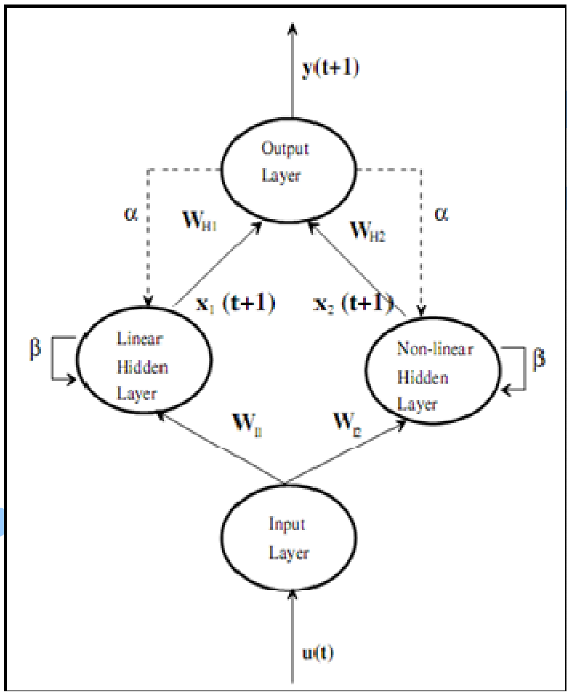

Figure 3. Schematic representation of the proposed three layered recurrent hybrid network

At a given discrete time t, let $U(t)$ be the input to a recurrent hybrid network, $Y(t)$, the output of the network, $X_1(t)$ the output of the linear part of the hidden layer and $X_2(t)$ the output of the non-linear part of the hidden layer. The equation of the feed forward hybrid network is as follows:

$$X_1(t+1) = W_I * U(t) + \beta X_1(t) + \alpha JY(t)$$
$$X_2(t+1) = F\{W_I * U(t) + \beta X_2(t) + \alpha JY(t)\}$$
$$Y(t+1) = W_{H1} * X_1(t+1) + W_{H2} X_2(t+1)$$

If only linear activation function is adopted for the hidden neurons, the above equations simplify to:

$$X(t+1) = X_1(t+1) = X_2(t+1)$$
$$Y(t+1) = W_H * X(t+1) \Rightarrow Y(t) = W_H * X(t)$$
$$X(t+1) = W_I * U(t) + S_1 * X(t) + S_2 Y(t)$$

Replacing $Y(t)$ by $W_H * X(t)$, gives:

$$X(t+1) = (S_1 + S_2 * W_H) X(t) + W_I * U(t)$$

where W_I is the matrix of weights of connections between the input layer and the hidden layer, W_{H1} is the matrix of weights of connections between the linear hidden layer and the output layer, W_{H2} is the matrix of weights of connections between the non-linear hidden layer and the output layer, $F\{\ \}$ is the activation function of neurons in the non-linear hidden layer. J is, $n_H * n_O$ matrices with all elements equal to 1, where n_H is the numbers of hidden neurons, and n_O, the number of output neurons. The weights in S_1 and S_2 are related to the feedback connections and are fixed. The weights of the connections from the output layer to the hidden layer have the same value α and those of the connections from the hidden layer to itself have the same value β. S_1 and S_2 are then given by $S_1 = \beta I$ and $S_2 = \alpha J$, Where I is the $n_H * n_H$ identity matrix.

$$\Rightarrow X(t+1) = (\beta I + \alpha J W_H) * X(t) + W_I * U(t)$$

General form of the equation can be written as follows:

$$X(t+1) = AX(t) + BU(t)$$

Where A is a $n_H * n_H$ matrix and B is a $n_H * n_O$ matrix.

The back propagation (BP) algorithm is used to update weights of the proposed neural network. The BP algorithm is a method of supervised neural network learning. During training, the network is presented with a large number of input patterns. The experimental outputs are then compared to the neural network output nodes. The error between the experimental and neural network response is used to update the weights of the network inter-connections. This update is performed after each pattern presentation. One run through the entire pattern set is termed an epoch. The training process continues for multiple epochs, until a satisfactorily small error is produced. The BP algorithm is the most commonly used to update the weights of the neural networks. Beginning with an initial (possibly random)

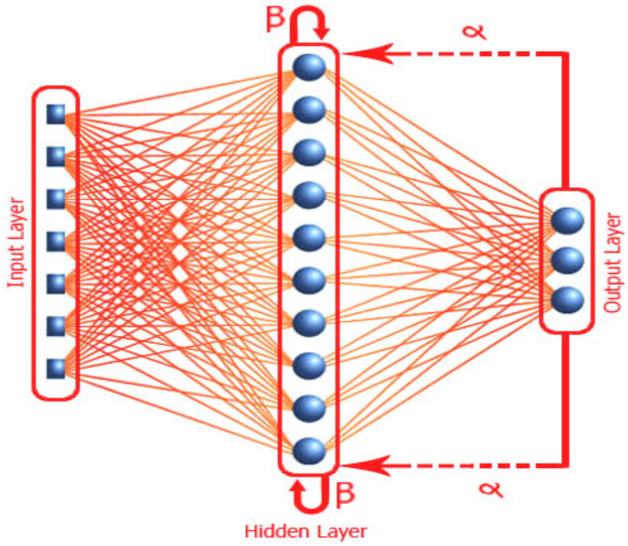

Figure 4. Three-layered neural network controller

weight assignment for three-layer feedforward network, proceed as flow:

Step1: Present i^p *, and form outputs,* o_i *, of all unit in network.*
Step2: Use the below equation to update Δw_{ji} *between the hidden layer and the output layer.*

$$\Delta w_{ji} = -\eta * \frac{\partial E_{pe}(t)}{\partial w_{ji}(t)} + \alpha \Delta w_{ji}(t-1)$$

Step3: Use the below equation to update weights between input layer and the hidden layer.

$$\Delta w_{ji} = -\eta * \frac{\partial E_{d}(t)}{\partial w_{ji}(t)} + \alpha \Delta w_{ji}(t-1)$$

Step4: Stop if updates are insignificant or the error is below a preselect threshold, otherwise return to Step1.

Where η is the learning rate, and α is the momentum term. $E_{pe}(t)$ is the propagation error between hidden layer and output layer. $E_d(t)$ is the error between desired and neural network output signals. Also as depicted in Figure 4 and 5, the first layer received reference inputs from the trajectory generator and also the control errors. Again, the second layer consisted of two parts, a linear part and a non-linear part. Finally, the third layer of the network produced outputs to drive the robot joints. The forward walking speed of PERSIA Humanoid Robot is 25cm/s. The image sequences of forward walking are shown in Figure 5.

Figure 5. Forward walking image sequence

Figure 6. Forward walking image sequence

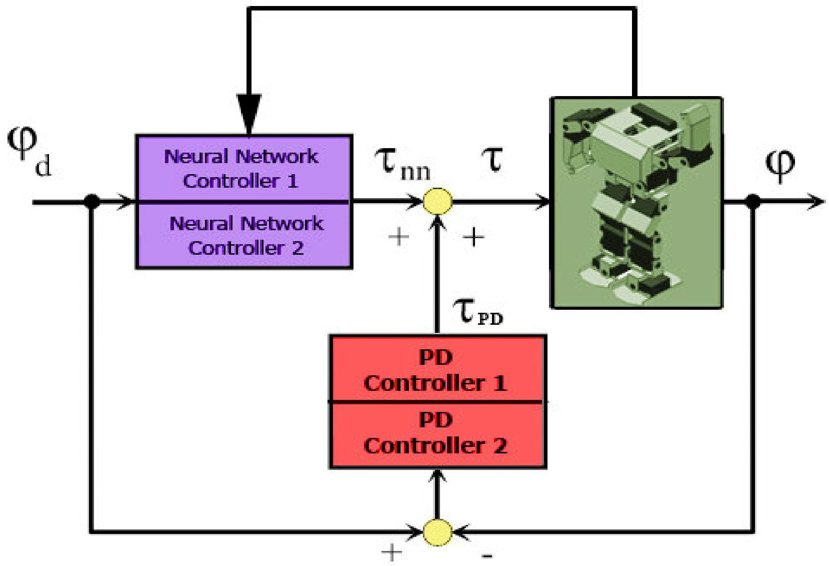

Figure 7. Proposed neural control system for walking robot

The image sequences of sideward walking and turning are shown in Figure 6 respectively.

4 Image Processing

One of the primary tasks for the vision system is to locate a particular goal and calculate the robot's position in relation to it. Utilizing a camera, each time the personal digital assistant (PDA) on each robot performs the processing of the current frame and calculates the position and direction of the robot. It also determines the position and position 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 (Color Adjustment) then recognizes the contiguous regions through either a BFS (Breadth-First Search) or a DFS (Depth-First Search) search algorithm and finally extracts the necessary information like: self-localization, ball, and goal and opponent position 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 processing the RGB color space is converted to HSL (Hue, Saturation and Luminance). To recognize a certain color, a combination of conditions on Hue, Saturation and RGB is used. This procedure makes the color recognition independent from the change of brightness and other 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 SD port connected to a camera with the speed of 15 to 30 frames per second. After identify the different objects on the field and estimated the real information, we are developing self-localization algorithms by classic triangulation methods. Basically, we infer robot position on field by the recognition of two landmarks and the goal pose with color. We will also implement localization algorithms based on field lines. Our new robot self localization method is based on detection of white lines in field. Because the white line points is one of the 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.

We employ a Monte Carlo Localization (MCL) method [MMT10b]. 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 $p(x_t \mid y_1 \ldots y_t)$ 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.

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

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.

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

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. We are also researching algorithms for color segmentation robust to variant light conditions and noise. The location of ball on the field is based on relative distance and orientation of ball with robot's position.

Figure 8. Vision Systems, Object detection (Goals, Flag Spots, Ball) and line detection for Self Localization algorithm

5 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 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!

6 Artificial Intelligent

In this section the AI part of the software is briefly introduced. There are three distinct layers: AI Core, Role Engine and Behavior Engine. AI Core receives the computed field data from World Map Modeling 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 behaviors. [PK99, M+09a]

7 Role Engine

Role engine module receives information from AI core; process them, and then selected a role. This module is the main section of robot software. The proposed rules for role engine have been turned by various experiences and they are independent of game field conditions. The output of this module is a set of high level commands that send to Behavior engine. Some of high level commands which are produced by role engine module are: Go to position, Go to ball, Targeting, Shooting ball, …

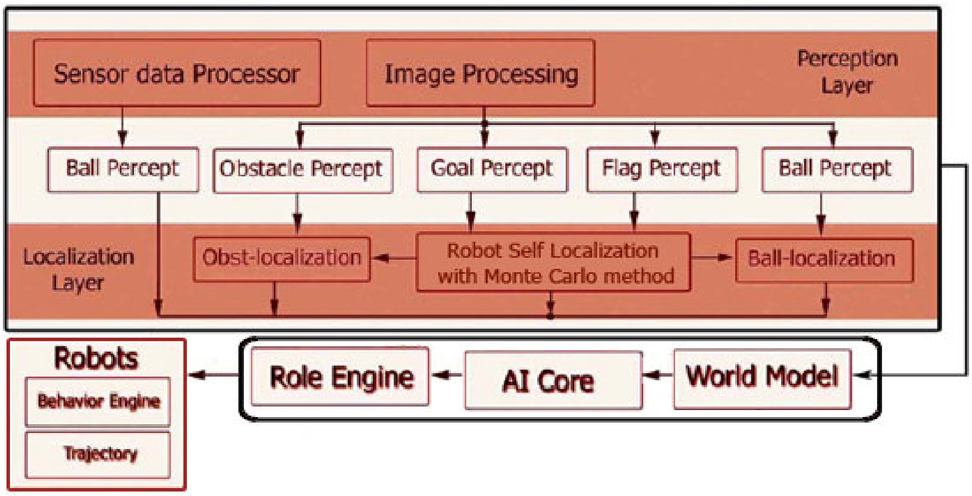

Figure 9. World model construction and artificial intelligent structure running on the PDA

8 Behaviors Engine

This module receives information from Artificial Intelligent unit. Total functions about Robot Behavior such as stability motors actions, robot path planning, turn camera, walking, shooting, dribbling; motion and etc are controlled in this section.

9 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) [MMK00, M+10d]. In this method different electrical charges are assigned to opponents, goal 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.

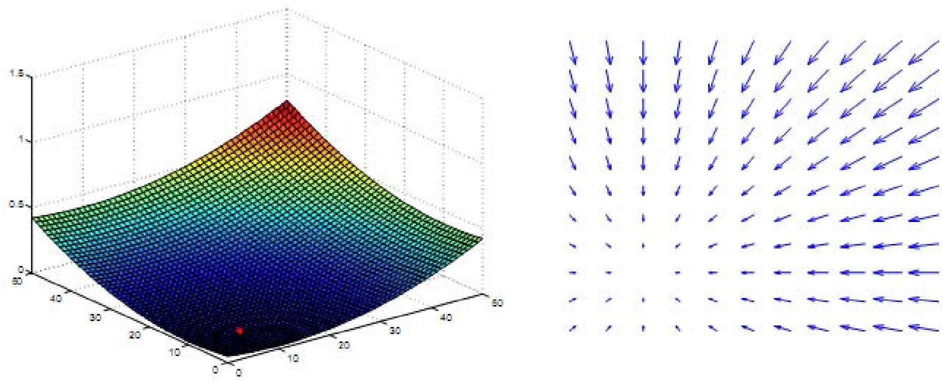

Figure 10. Goal force (Attractive potential to the goal)

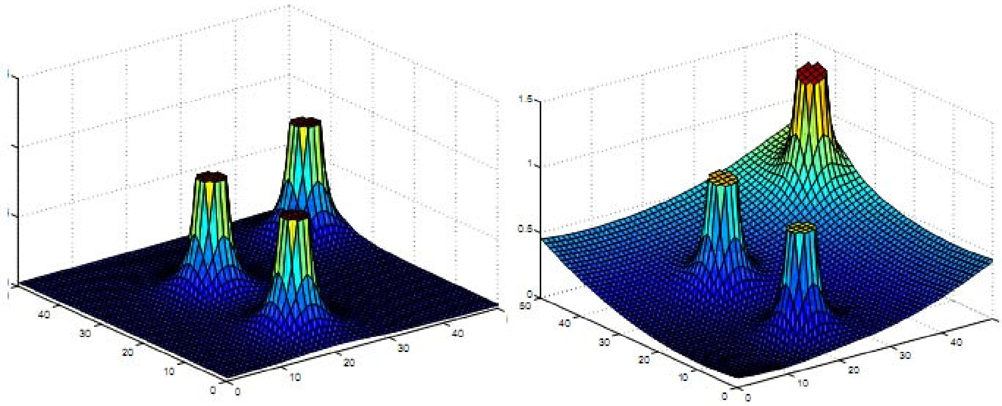

Figure11. Obstacle force and Goal Force+ Obstacle force

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.

10 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.

Conclusion and Future Work

In this paper, we present the details of design and implemented method of our humanoid soccer robot. The performance of our robot team in Iran-Open Robocup competitions 2010 (3st place) and 1st national Khwarizmi Robotic Competitions in 2009 showed that the combination of methods and techniques that described in this paper are led to a successful Humanoid Robot. In our robot, Omni directional walking, vision system and a novel motion controller based on artificial neural network and PD controller have been combined to create a comprehensive Humanoid robot. Our robots can get up from a fall, walk forward and backward, turn right and left, and kick the ball. A camera, two IR sensors and a digital compass are integrated so that the robot can obtain the environmental information to decide the action behavior. We try to making the robot more stable and reliable as the result of our researches. Future plans are to develop and implement autonomous team actions towards participation in the three-by-three games similar to other leagues in RoboCup. Further information's and video about our works are presented on our website www.IranAdro.com/HumanoidIndex.htm .

Acknowledgment

We would like to thank a lot, to Prof. Amir Hassan Monadjemi, who played important role in software design and fruitful discussions on Image Processing and Multi Agent Robot. Also, we are very grateful to all our sponsors.

References

[MMK00] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, S. A. Kasaei -** *Development an Omni directional Walking Based on Neural Network Controller for Biped Robot,* International Journal of Computer and Electrical Engineering Published by: International Association of Computer Science and Information Technology Press (IACSIT), Singapore, ISSN 1793-821X.

[MMM10] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, S. A. Mohades Kasaei -** *Development a Real Time Cooperative Behavior Approach for Autonomous Soccer Robots Applied in Robocup-MSL,* International Journal of Robotics and Automation (IJRA), Volume (1): Issue (2) pp: 26-41, published by: CSC Journals, Kuala Lumpur, Malaysia, ISSN: 2180-1312 (2010).

[MMT10a] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, M. Taheri -** *Design and implementation efficient and reliable mechatronics module for autonomous mobile soccer robot,* 13th international Conference on climbing and walking robot and the support technologies for mobile machines 31August-3 September 2010, Nagoya, Japan.

[MMT10b] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, M. Taheri -** *A practical approach to modeling, control and self-localization of a fully autonomous soccer robot besed on sensor and information fusion*, 13th International Conference on climbing and walking robot and the support technologies for mobile machines 31August-3 September 2010, Nagoya, Japan.

[M+09a] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, S. A. Mohades Kasaei, M. Taheri, M. Rahimi, H. Vahiddastgerdi, M.Saeidinezhad -** *Design and Implementation a Fully Autonomous Soccer Player Robot,* Proceedings of World Academy of Science, Engineering and Technology, Volume 39, ISSN: 2070-3740, Hong Kong, China, March 23-25, 2009..

[M+09b] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, M. Taheri, H. Vahiddastjerdi, M. Rahimi -** *Effective Mechatronic Models and Methods for Implementation an Autonomous Soccer Robot,* 17th IEEE Iranian Conference on Electrical Engineering , Tehran, Iran, May 12-14, 2009.

[M+09c] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, M. Taheri, H. Vahiddastjerdi, M. Rahimi -** *Design and Development of a Novel Industrial Humanoid Robot*, 9th Iranian Conference on Manufacturing Engineering (ICME), Birjand, Iran, March 3-5, 2009.

[M+10a] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, S. A. Mohades 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 Publishing Limited [ISSN 0143-991X] 37/3 (2010) 279–286, [DOI 10.1108/01439911011037686].

[M+10b] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, S. A. Mohades Kasaei, M. Taheri, S. A. Monadjemi -** *Dynamic role engine and formation control for cooperating agents with robust decision making algorithm*, Industrial Robot: an International Journal, Emerald Group Publishing Limited [ISSN 0143-991X].

[M+10c] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, S. A. Mohades Kasaei, M. Taheri** - *Design of an Action Selection Mechanism for Cooperative Soccer Robots Based on Fuzzy Decision Making Algorithm,* BRAIN. Broad Research in Artificial Intelligence and Neuroscience Volume 1, Issue 3, July 2010, pp: 5-18, Romania, ISSN 2067-3957.

[M+10d] **S. H. Mohades Kasaei, S. M. Mohades Kasaei, M. Taheri, M. Rahimi -** *PERSIA Team Description Paper2010,* T.D.P , RoboCup Symposium & Competition, Robot Soccer World Cup, LNCS, Springer,2010.

[PK99] **J. H. Pauk, H. Chung -** *ZMP Compensation by On-Line Trajectory Generation for Biped Robots,* IEEE International Conference on Systems, Man, and Cybernetics. 4 (1999) 960-965.