Design of an Action Selection Mechanism for Cooperative Soccer Robots Based on Fuzzy Decision Making Algorithm

S. Hamidreza Mohades Kasaei
Islamic Azad University – Khorasgan Branch, Young Researchers Club, Isfahan, Iran
H.Kasaee@Khuisf.ac.ir

S. Mohammadreza Mohades Kasaei
Islamic Azad University – Khorasgan Branch, Young Researchers Club, Isfahan, Iran
M.Kasaee@Khuisf.ac.ir

S. Alireza Mohades Kasaei
Islamic Azad University – Khorasgan Branch, Young Researchers Club, Isfahan, Iran
Alireza.Kasaee@Gamil.com

Mohsen Taheri
Islamic Azad University – Khorasgan Branch, Young Researchers Club, Isfahan, Iran
Css_Mt@Yahoo.com

Abstract

Robocup is an international competition for multi agent research and related subject like: Artificial intelligence, Image processing, machine learning, robot path planning, control, and obstacle avoidance. In a soccer robot game, the environment is highly competitive and dynamic. In order to work in the dynamically changing environment, the decision-making system of a soccer robot system should have the features of flexibility and real-time adaptation. In this paper we will focus on the Middle Size Soccer Robot league (MSL) and new hierarchical hybrid fuzzy methods for decision making and action selection of a robot in Middle Size Soccer Robot league (MSL) are presented. First, the behaviors of an agent are introduced, implemented and classified in two layers, the Low_Level_Behaviors and the High_Level_Behaviors. In the second layer, a two phase mechanism for decision making is introduced. In phase one, some useful methods are implemented which check the robot's situation for performing required behaviors. In the next phase, the team strategy, team formation, robot's role and the robot's positioning system are introduced. A fuzzy logical approach is employed to recognize the team strategy and further more to tell the player the best position to move. We believe that a Dynamic role engine is necessary for a successful team. Dynamic role engine and formation control during offensive or defensive play, help us to prevent collision avoidance among own players when attacking the ball and obstacle avoidance of the opponents. At last, we comprised our implemented algorithm in the Robocup 2007 and 2008 and results showed the efficiency of the introduced methodology. The results are satisfactory which has already been successfully implemented in ADRO RoboCup team. This project is still in progress and some new interesting methods are described in the current report.

Keywords: Cooperative Multi-agent system, Decision Making, Decision Tree, Role Assignment, ID3 Algorithm.

1. Introduction

Robot soccer games had been popular with educational institutions around the world since the inauguration of the Robocup competition in 1997. This initiative provide a good platform for Machine Learning (ML) techniques and Multi Agent research, dealing with issues such as cooperation by distributed control, neural networks, genetic algorithms, decision tree, decision making, fuzzy logic and also many hybrid approaches as a combination of some aforementioned ones ML techniques. In order to work in the dynamically changing environment, the decision making system of a soccer robot system should have the features of flexibility and real time adaptation. The robots in middle-size league should only use local sensors and local vision. Each team can have a maximum number of six robots. They can communicate with each other through a central computer via a Wireless Network. The rules in the competition are the same as the international soccer rules as far as they are practical for robots. The image information of the entire soccer field is captured by robot CCD camera then analyzes the image information to determine the situation of the soccer field and sent to the corresponding host computer. The host computer then analyzes the information to determine the situation of the ball and robots position in the field [1][2]. According to the determined situation, the host computer decides a strategy and plans the motion modes and the corresponding velocity commands for every soccer robot of the same team. Each soccer robot of ADRO then receives a role command from the host computer. A decision tree has several nodes arranged in a hierarchical structure. It implements decisions in a simple, apparent, multistage manner. Furthermore, since each node of a decision tree uses only a simple splitting role and a small subset of all features, the entire decision process is fast and efficient.

In this paper a new methodology for decision making in RoboCup soccer with all abovementioned problems consideration is introduced. .We present our work in Dynamic Role allocation for a soccer robot team to modify robots role and strategies and decide which strategy should be applied to the current situation. Dynamic Role allocation allows a team to divide its main objective in a couple of sub-objectives more specialized and adapted to the location of each of the teammates and the strategy of the team. Strategic game play involves role switching for teams with same robots and formation control during offensive or defensive play, collision avoidance among own players when attacking the ball and obstacle avoidance of the opponents [4].

Figure 1. ADRO middle size Soccer team from Islamic Azad University- Khorasgan banch, Isfahan, Iran

2. Robot Software

We have developed a software system to fully utilize the hardware abilities. In this paper introduces software parts contain: Robot Strategy, Role Specification, World Model Construction, Artificial Intelligence, Trajectory and Network from a viewpoint of software system [3][5]. 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.

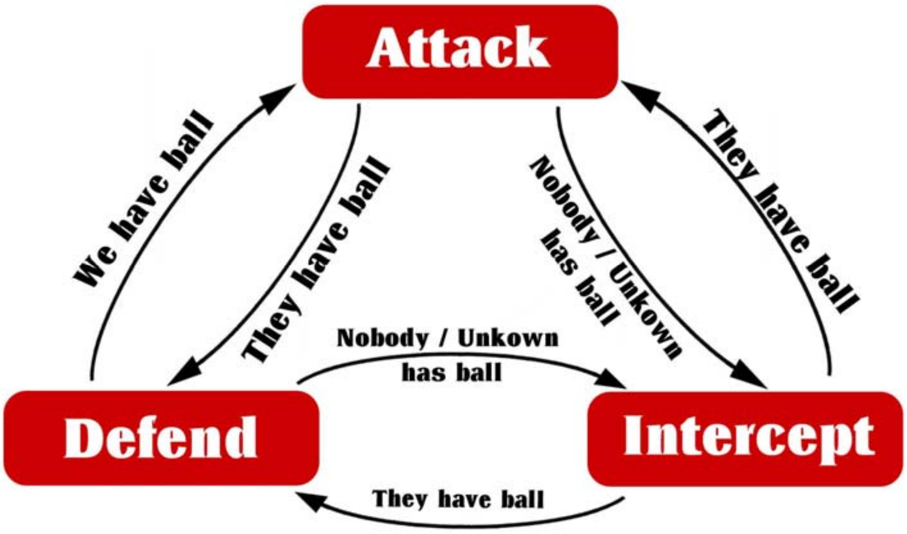

Figure 2. Finite state machine to determine the team strategy.

3. Robot Behavior

In the proposed methodology some applicable skills are introduced and the decision making policy is developed by considering the features and limitations of this environment. The skills are classified in two layers, in the first layer there are the simple actions which are already implemented (Low_Level Behaviors) and in the second layer the actions are more complicated and sometimes a combination of basic skills are used (High_Level Behaviors). Besides, the decision making process has two steps; First step considers the robot's abilities of performing a high level skill according to his circumstances and the second step considers the robot positioning and choosing the best action regarding to the first step results. The Low Level Behaviors are defined as the actions, which are already implemented by ADRO RoboCup team; they can be employed by sending the proper commands to robot actuator, these commands are: Move(X,Y), Capture_Image(), Read_Ecoder() and etc. High Level Behaviors are those in which the world model information and the Low Level Behaviors are being applied. These skills consist of the actions with ball like: pass, shoot, dribble, etc. and actions without ball like: mark a robot or position, find objects in noisy frames. To actions with ball these actions, basically we need some information about the ball treats, when a force with a particular angle is applied, considering the environment parameters. A number of complicated mechanical formulas could help to predict the ball movements. Some of the most applicable skills with ball are as Shoot, Ball, Dribble (Move with ball), Clear Ball, and Actions without ball skills make robots to be arranged in positions so that they would have the most chance to create opportunities for team or to get the opponents opportunities.

4. Strategy

In robot soccer systems, images of objects on the field are processed by a vision system. Analysis of this primary data will yield information such as identification of objects including ball, line, player, and opponents. Other information such as object identity (identity of player), opponent, position, orientation and velocity can also be computed. Based on this information, each of the players carries out assigned roles including attacker, defender, supporter and goalkeeper. The simplest role selection strategy is to have a fixed role that does not change throughout the game. However, permanent role causes undesirable behavior such as a defensive player not going for the ball even though the ball is near but outside its defense zone or a forward player giving up its possession of ball when it incidentally enters a defense zone. The main objective in competitive environment is to score goals; and if a player is in a better position to secure a scoring chance, it must be given the opportunity. A cost function evaluates some parameters like the ball distance, localization, player's orientation towards the ball, etc. and obtains a value for each role. This parameters will be calculated periodically and roles will be assigned to robots according to the values obtained [7][12].

Figure 3. ADRO team formation control against MRL attacker robot

5. Decision Making Phase I

In this phase, the robot ability of performing an action according to the environment situation (e.g. opponents' positions/speed, ball position/speed and their predicted states) is tested by Decision Makers (DM) to confirm if that action can be done by the robot in that condition. Some of these decision makers are explained below:

a) DM for shooting to a position:

To determine if a robot can shoot to a position or not, first the robot calculates the minimum degree, for shooting, if the degree could be found less than Max Kick Degree then tries to find an angle between min degree and Max Kick Degree and a force less than Max Kick Force. If the angle and a force found with these properties the DM returns true, otherwise it returns false.

b) DM for Shoot to goal

First, the agent quantizes the goal, to n discrete positions. For each position first checks the Shoot to position conditions, if the result is true then shoot the ball toward the goal otherwise check and select other behaviors according to environment situation.

c) DM for Dribble

To determine if it is safe enough for player to dribble, First, the player check the path is free with a potential field algorithm, Then he checks If there is no opponent with distance less than or equal with SecureDribbleDistanceinside; If the result is true then select Dribble behavior, otherwise check and select other behaviors according to environment situation.

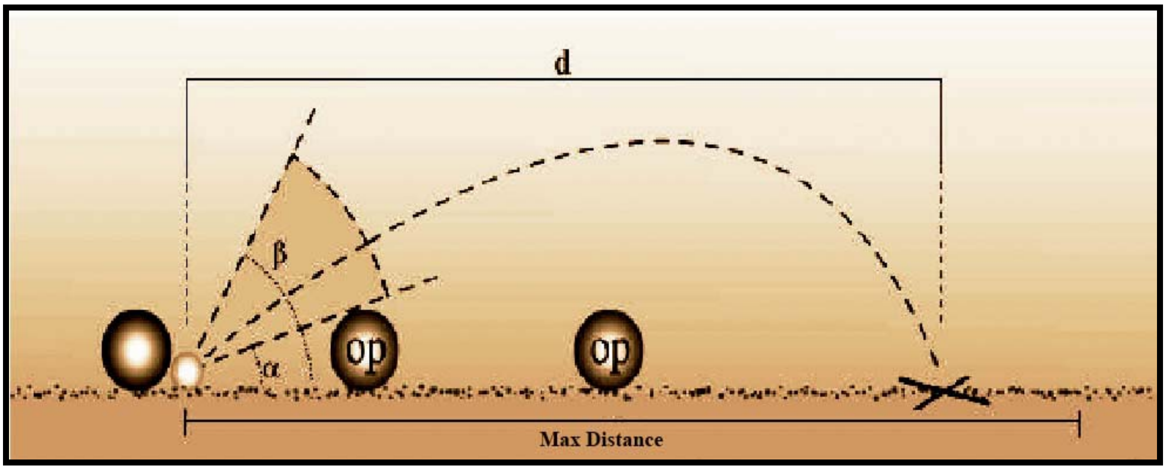

Figure 4 Shooting to a position requires minimum angle α to be less than (*Max Kick Angle*) and minimum distance *d* less than (*Max Kick Distance*)

6. Decision Making Phase II

In The major issues we have addressed in this phase are the dynamic assignment of roles and team strategy. We adopted a formation/role system. Formation contains:

• Formation Name: Like real soccer team formations.
• Strategic area: The area in which the player is mostly supposed to be.
• Center of strategic area: also known as the home position
• Player role: we introduced ٤ applicable roles for agents: goal keeper, defender, half backer and attacker.

In this methodology the player role and team strategy are dynamically assigned to the layers according to with different factors. The most important factor is the ball position. According to this changes in strategy the strategic area of the player, changes. To select the appropriate strategy we developed a fuzzy algorithm.

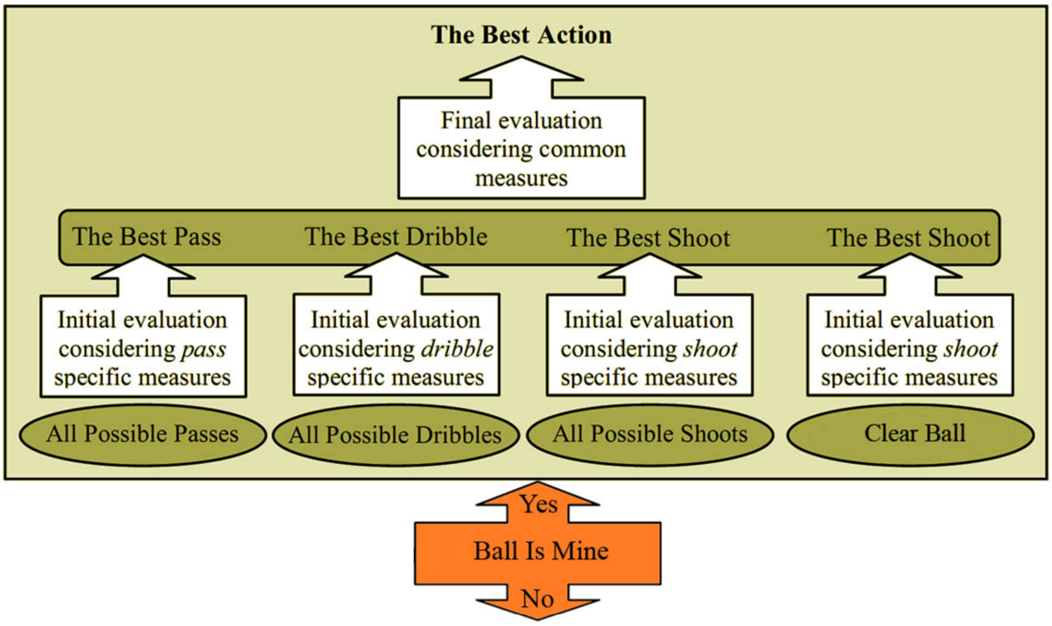

Figure 5. The diagram for action selection when players own the ball

In addition this algorithm helps a robot to find out the proper distance with ball according to his strategic area. Depending on the robot's team or the opponent's team owns the ball, the output variable strategy of fuzzy function may change. The last remaining condition is when the robot owns the ball, in this state agent uses the phase 1 decision making results to perform an appropriate operation.

7. Roles Specification

The roles can be assigned to each robot in a statically or in a dynamical way. Dynamic role allocation benefits for example, from opportunistic situations like fast ball changes along the field, or failures in some robot. We have considered six main roles in the Robocup domain: Goal-keeper, Attacker, Defender (Right Defense, Center Defense, Left Defense), and Supporter. Goal-keeper is the only role assigned statically. The main reason to have a single player to the Goal-keeper role is that the rules do not allow that players enter in its own goal area (like in the hand ball). The rest of roles can be exchanged among robots according to game conditions. Next, we are going to describe the objectives of each role and the advantages we have obtained using dynamic role allocation:

a) Goal-keeper: Its goal is to protect its own goal from shots by the other team players. Also, it should rest in its own area.

b) Attacker: It tries to get the ball and to carry it, or to kick it, towards the goal. When the other team has got the ball, it tries to recover it actively (going after the ball). None of the other roles are devoted to get the ball. This approach has one implicit advantage: It avoids collision among players of our team, which is explicitly penalized in the rules.

c) Defender: Its goal is to intercept the ball if an opponent kicks it to its goal. Furthermore, it should stand in the way of the opponent and should try to hide the goal preventing the opponent to kick the ball. Another implicit consequence of the Defender role is that one robot of the team always remains in a position near its own net. This fact is very useful taking into account that the ball quickly moves from side to side. We have always one robot covering its defending half of the field.

d) Supporter: The function of this role is to assist the Attacker in its path, and to cover the maximum amount of field in case the ball will be kicked in the wrong way. The main contributions of this role are to recover the ball if the kicks made by the striker do not go in the good direction, and also to maintain a good position for future passing kicks.

8. World Model

The World Model is responsible to build a world model using sensorial data. From the sensory inputs and the static information about the game, the Word Model builds the game model, which consists of basic information, like ball position and player postures (Live or Dead), and advanced information such as cooperation decisions. The variables used to define the world model are stored in a Blackboard. The Blackboard is a data pool accessible by several components, used to share data and exchange messages among them. Traditional blackboards are implemented by shared memories and daemons that awake in response to events such as the update of some particular data slot, so as to inform the components requiring that data updated. In our implementation, the Blackboard consists, within each individual robot, of a shared memory among the different components, organized in data slots corresponding to relevant information (e.g. ball position, goal position), accessible through data-keys. Some variables of the blackboard are local, meaning that the associated information is only relevant for that robot, but others are global, so their updates must be broadcasted to the other teammates (e.g., the ball position) (Lima 2002). The cooperation is divided in a high-level cooperation and a low-level cooperation. The former one is stored in the Black-board, and consists of Group-Level and Team-Level Tactics, that can be viewed as analogues of the coach's directives in real soccer. The Group Level Tactics defines tactical parameters for the different player groups: defense, mid-field and attack. For instance, a good defensive tactic is to form a defensive line with the goalkeeper to block all paths to our goal. The Team-Level Tactics set general tactical conditions of the whole team. Parameters as basic formation, e.g. 2 defenders 1 attacker, if we are in a defensive play, or 1 defender 2 attackers, if we are in an offensive play. The low-level cooperation is outside the blackboard because it is a commitment between the robots that are involved in a cooperative action, e.g. when a robot tells another teammate to move to a certain position, in order to be able to receive a pass. It's necessary to have a communication method between all the robots, so they can exchange messages among them. We pretend to use an Agent Communication Language (ACL) (FIPA 2002), allowing us to use a standard and highly flexible message format. [6][9]

Figure 6. Defense strategy with two robots

9. 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 [11]. AI Core receives the computed field data from world 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.

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.

10. Multi-Cost Function for Role Assignment

Role assignment used by many teams is usually computed in real time. Role assignment is necessary to avoid collision of players going for the ball or no player being assign such a role to attack the ball. At the Strategy Decider block has as inputs, amongst others, the score and the remaining time. Based on this information this block selects a fitting strategy that is to be applied,

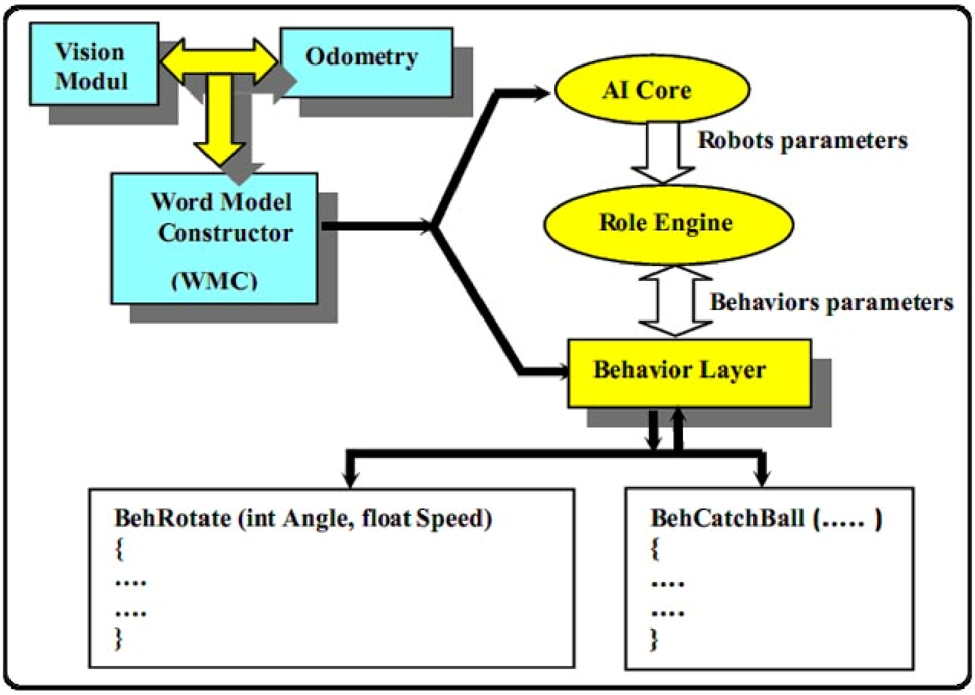

Figure 6. World model construction and artificial intelligent structure

i.e. a combination of a formation (e.g. 1 keeper, 2 defenders, 3 attackers) and a tactic (e.g. center attack, tight defense). The Strategy Function has the following functionality: (1) load formations and tactics before the game starts; (2) dynamically choose new formations and tactics during the game, based on the environment information; (3) send a list of ٦ unique role identifiers (6 Role IDs) to the Role Engine, based on the chosen formation and tactic. The Role Engine is the component in charge of determining the applicability of the roles with respect to the robots and assigning the most suitable role to each robot. In other words, the role engine determines robot a set of suitable roles for each robot and a rank for each role as a measure of its applicability. On the basis of these ranks, the Role Assigner assigns to each robot the most suitable role while avoiding conflicts with the assignment of this role to another robot. The ranking is based on information about the mechanical state of the robot, the positions of the robot and its team members, and the position of the ball. A general definition for parameters that use in fuzzy arbiter is "value to estimate the cost of executing an action". The parameters will be calculated periodically and roles will be assigned to robots automatically according to the values obtained. In our role assignment the parameters will be individually computed by each robot as inputs to the fuzzy arbiter factors are Distances to ball, goal, path obstacle and etc. These fuzzy variables are defined below:

a) Distance to ball: is the distance of the robot to the ball.
b) Distance to our goal: is the distance of the robot to our goal.
c) Distance to opponent goal: is the distance of the robot to the opponent goal.
d) Orientation: is the orientation of the robot with respect to the straight line path to the ball.
e) Path obstacle: is the angle bounded between the vector of the robot to the ball and the vector of the robot to the obstacle. To fuzzily the distance variable, the ratio of the minimum

distance To Ball to the distance To Ball value is used, see equation 1. That is, the nearest robot to the ball will have a membership of value 1.0 for this variable.

$$U_{Dist.ancetoball} = \frac{Dist.ancetoball\_Min}{Dist.ancetoball} \qquad (1)$$

Equation2 describes the membership function for the Orientation variable. A single cosine function is used. The robot that is directly facing the ball will have an orientation angle of 0 degrees, and a membership value of 1.0 for Orientation.

$$\begin{aligned} &U_{Orientation} = Cos(Orientation) && For \quad -90 \le Orientation \le 90 \qquad (2)\\ &U_{Orientation} = 0.0 && Otherwise \end{aligned}$$

The 'or' operation used is the algebraic sum operation. All the fuzzy memberships are added together and the resultant is the membership value of the role Assigned. Next are the equations to select the appropriate robot for each role. In this paper we suppose that the order in roles assignment is Striker, Defender and Supporter.

$$\begin{aligned} &U_{Strike.r} = Min(U_{i,Strik.r}) && \forall i \in (1...n) \\ &U_{Defender} = Min(U_{i,Defender}) && \forall i \in (1...n) \quad \wedge \quad i \ne Robot_{striker} \qquad (3)\\ &U_{Supporter} = Min(U_{i,Supporter}) && \forall i \in (1...n) \quad \wedge \quad i \ne Robot_{striker}, Robot_{Defender} \end{aligned}$$

The robot with the highest membership value is assigned the highest priority order among the robots to the role of "attack the ball". Every robot updates its utilities periodically, and broadcasts this information to its teammates. We will refer to this information as coordination information. Finally, the Behavior Executer block executes the behavior that belongs to the assigned role (Role ID). The behavior has to make decisions that are based on the environment at that time. For this purpose, it requires information from the World Model. The assigned role is constructed based on a decision tree (Behavior tree), which consists of the following three types of nodes: Selector, Sequence, and Action, see Fig. 7, 8, 9. The reason that we choose for a behavior tree over for instance a finite state machine is that behavior trees support a clear visual interpretation. The design uses weights for selecting a strategy and assigning the roles. These weights are a priori defined and during the match these weights cannot be changed. In the future, learning will be added to adjust the weights during a match such that a robot will make different decisions as the match develops. The information sent is its own location, and an estimation of its distance to the ball. Coordination information is sent at 5Hz. When one robot receives coordination information it updates data associated to the corresponding robot in its global model. This global model stores position of the teammates and, combined with the position of the ball is used to calculate utilities functions.

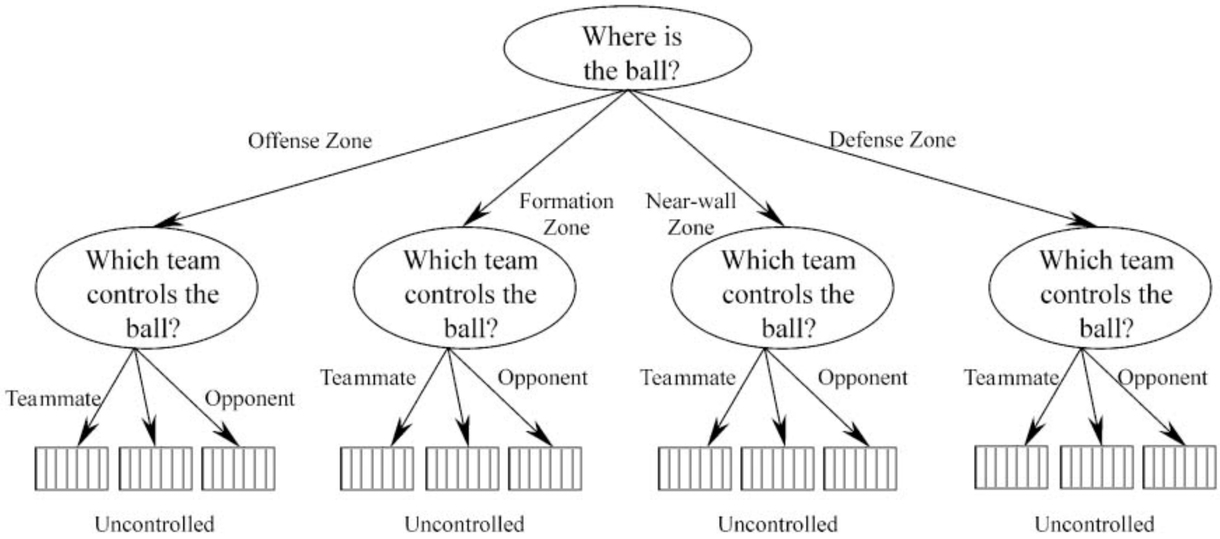

Figure 7. The decision tree which implements the strategy-based decision making system.

11. 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 (Figure 11). 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

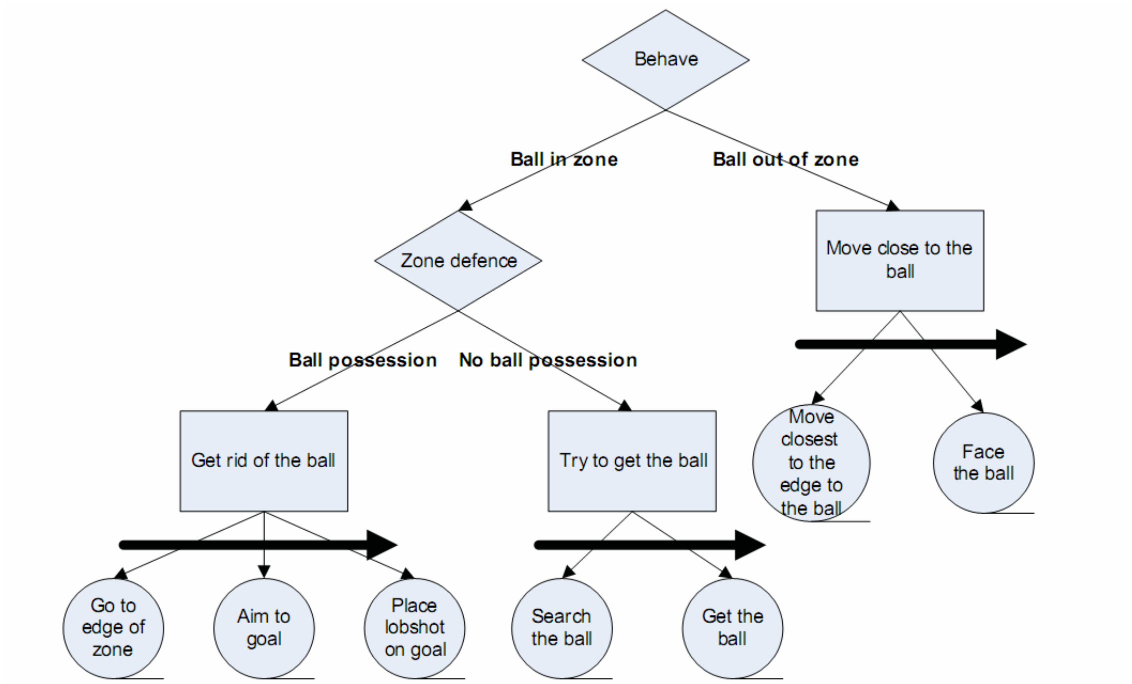

Figure 8. Example of a behavior tree: selector (diamond), sequence (rectangle), action (circle).

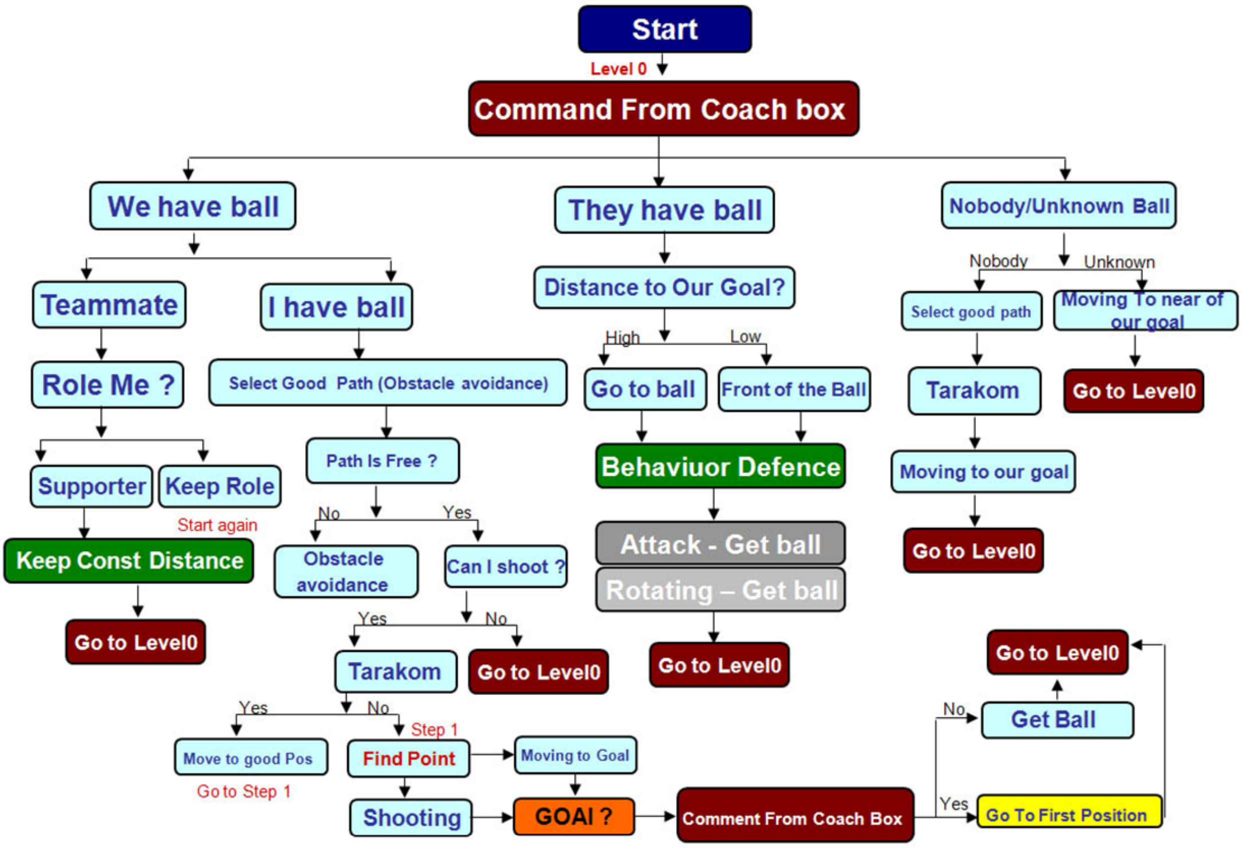

Figure 9. The decision tree which implements for soccer robot AI

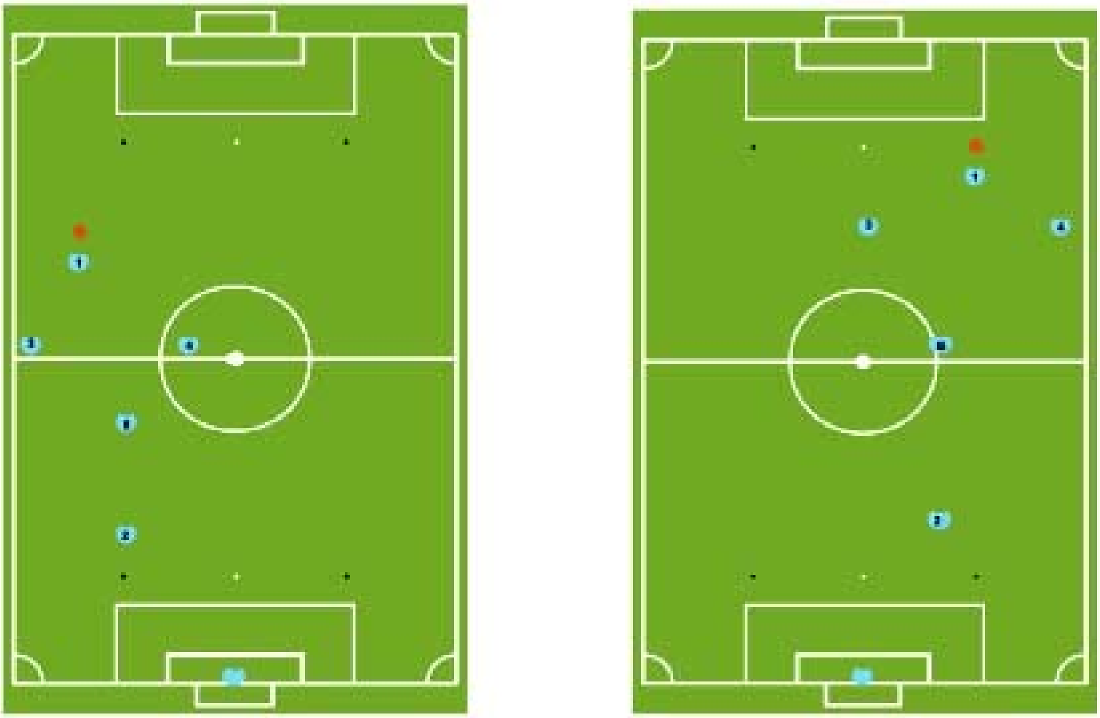

Figure 10. ADRO Robots in some different game situations.

opponents same as the flowing of a bulk of water through obstacles)[7][8][9]. 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. 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. 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.

12. 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.

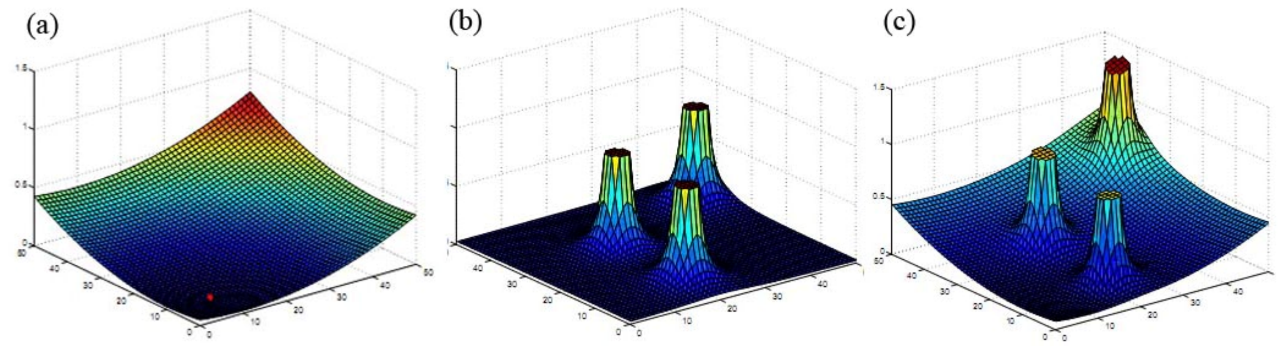

Figure 11. Potential at every point; it is highest in the obstacles and lowest at the goal. Elsewhere it is generally higher farther from the goal and near obstacles. [Image a describe the Goal force (Attractive potential to the goal) and image a describe the Obstacle force (Repulsive potential) and finally image c is Goal Force plus Obstacle force]

13. Result

We implemented this methodology on ADRO RoboCup Team. We compare ADRO team performance2008 with its previous version 2007 The results showed the success of this methodology; The team performance in coordination and collaboration highly improved; in fact the players switched their strategic area smoothly as the team strategy changed in a reasonable manner, the robots carried out the high level behaviors much more efficiently and at last the final results enhanced significantly. 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.

14. Conclusion and Future Work

In this paper, a new method for decision making in RoboCup Middle size soccer robot (MSL) has been proposed. First identification used to the RoboCup middle size soccer robot which led to mechanical formulas, then soccer skills were introduced and classified in different layers then a two phase mechanism for decision making is presented, We have developed a basic role engine and formation control mechanism among members of a multi robot team. Robot localization and local ball estimation are the elements shared. Combining periodically the information received with local information, each robot updates a global model of the environment. Using coordination we have also got a very good way to identify the rest of the team members. In the development of robot soccer where players are homogenous, role engine and formation control becomes a necessity to formulate an efficient strategy to achieve the goal of a successful game. Using a rule based approach allows the strategy for role selection to be naturally developed using domain expertise rather than the alternative of trying to find a suitable cost function that would provide the same performance. in this mechanism both fuzzy and non fuzzy algorithms are applied and finally the proposed methodology implemented on a ADRO RoboCup soccer team and the results. Further information's and video about our works are presented on our website *http://www.IranAdro.com*

15. Acknowledgment

We would like to thank a lot, to Dr. 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.

16. References

[1] 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].

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

[3] 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.

[4] S.H.Mohades Kasaei, S.M.Mohades Kasaei, S.A.kasaei, M.Taheri "*Design and Development of Novel Solenoid-Based Kicking System".* 9th Iranian Conference on Manufacturing Engineering (ICME), Birjand, Iran, March 3-5, 2009

[5] Heung-Soo Kim, Hyun-Sik Shim, Myung-Jin Jung, and Jong-Hwan Kim., "Ac-tion Selection Mechanism for Soccer Robot", Proceedings of IEEE international Symposium on Computational Intelligence in Robotics and Automation, 1997, pp 390-395.

[6] Hyun-Sik Shim, Yoon-Gyeoung Sung, Seung-Ho Kim and Jong-Hwan Kim, "Design of Action Level in a Hybrid Control Structure for Vision Based Soccer Robo", Proceeding of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp1406-1411.

[7] S.H.Mohades Kasaei, S.M.Mohades Kasaei, S.A.Mohades Kasaei, M.Taheri " *Autonomous navigation systems for Mobile Robots base on Omnidirectional Vision System*", 12th international

Conference on climbing and walking robot and the support technologies for mobile machines 9-11 September 2009, Istanbul, Turkey.

[8] S.H.Mohades Kasaei, S.M.Mohades Kasaei, S.A.kasaei, M.Taheri " *Improvement performance for Humanoid Soccer robot via Robust Structure Design and Fuzzy Rule-Based Motion Controller*", the International Conference on Mechanical Engineering, Tokyo, Japan, May 27-29, 2009. (ISSN 2070-3740).

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

[10] Roth, M., Vail, D., Veloso, M. "A world model for multi-robot teams with communication". In: IROS2003, October 27 - 31, 2003, Las Vegas, USA.

[11] S.H. Mohades Kasaei, S.M.Mohades Kasaei, S.A.Mohades 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).

[12] S.H.Mohades Kasaei, S.M.Mohades Kasaei, S.A.Mohades 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.