6

Distributed Architecture for Dynamic Role Behaviour in Humanoid Soccer Robots

Carlos Antonio Acosta Calderon, Rajesh Elara Mohan and Changjiu Zhou
Advanced Robotics and Intelligent Control Centre, Singapore Polytechnic
500 Dover Road, Singapore 139651

1. Introduction

In recent years, robotics competitions have flourished all over the world. These competitions have been accepted among the scientific community because of their roles in the in the advancement of science. Roboticists have understood that competitions do not only nurture innovative ideas, but they also serve as a common testbed, where approaches, algorithms, and hardware devices could be compared by evaluating them in the same environment and under identical conditions. Competitions also motivate students to be involved in robotics to acquire new technological and problem solving skills. Robot soccer has proved to be a challenging and inspiring benchmark problem for artificial intelligence and robotics research. In a soccer game, one team of multiple players must cooperate in a dynamic environment and sensory signals must be interpreted in real time to take appropriate actions. The soccer competitions test two multi-robot systems competing with each other. The presence of opponent teams, which continuously improve their systems, makes the problem harder every year. The number of goals scored is an objective performance metric that allows a comparison of the systems.
Two of the most successful competitions for robot soccer are FIRA (Federation of International Robot-soccer Association) and RoboCup. Both FIRA and RoboCup have their international conferences co-located with the games to promote scientific dissemination of the novel ideas and solutions proposed by the teams. Currently, there is a number of different soccer leagues in RoboCup and FIRA focusing on different aspects of the soccer challenge. Both competitions have a humanoid league, where autonomous robots with a human-like body and human-like senses play soccer against each other. RoboCup has set the final target of the humanoid robot soccer competitions for being able to develop a team of humanoid soccer robots capable of defeating the human world champion team by 2050 (Kitano & Asada, 2000). Although humanoid soccer robots are far from human performance, their progress is particularly visible. Nowadays, the robots manage basic soccer skills like walking, kicking, getting-up, dribbling, and passing.
The early stages of these competitions consisted only of basic robotic soccer skills, such as walking, getting-up, penalty kick, and obstacle avoidance. In 2005 RoboCup introduced the 2 vs. 2 soccer games for the Humanoid League, which became 3 vs. 3 soccer games in 2008. It is planned to increase the number of robot players in the games until eventually it reaches

11. Raising the number of players poses new challenges for the roboticists and further expands the possibilities of team play. The increased complexity of soccer games with more players will make structured behaviour a key factor for a good humanoid soccer team. Cooperative behaviour of the humanoid robots would give advantage to the team to achieve its ultimate goal, to win the game.
In soccer, cooperative behaviour is displayed as coordinated passing, role playing, and game strategy. Wheeled robots with global vision systems and centralized control have achieved such behaviours; example of this is the RoboCup Small-Size League. In the RoboCup Small- Size League, teams are able to display formations according to the strategy of the team. In addition, the robots show a well-defined behaviour according to their assigned roles during the game. The robot's role would determine the type of contribution of a robot to the strategy of the team. Passing is therefore a consequence of the behaviour for the roles and the support for the strategy. In RoboCup Small-Size, a central computer is responsible for deciding the roles, positions and behaviour of the five robots of the team. The central computer receives a complete image of the soccer field from an overhead camera; this image is then processed and used to calculate new positions and states for each robot. Finally, the position and states are sent to the robots. In the humanoid soccer games, there is no central computer; each robot is meant to be autonomous and is equipped with a camera as a main source of information about its environment. The partial information about the environment that the humanoid robot can collect with the camera, along with received information from the team-mates, is the information used to determine its behaviour. As it could be guessed, the partial information about the environment and other robots makes the problem of behaviour control quite challenging. Most of the teams in the RoboCup Humanoid League have identified the need to have different roles for the robots in the team. This role assignation is then useful to specify particular behaviours that must be unique to the role of the robot, e.g. the goalie is the only robot that is allowed to dive to block the ball when it is drawing near the goal. Despite the obvious advantages to the static role assignation, some drawbacks are still observed. The roles can only be changed before or after the game, i.e. during a normal game, the roles of the robots are not allowed to change. If a robot is damaged, and cannot continue the game, the other robots could not adjust their roles to compensate for the team's disadvantage.
This Chapter describes the approach developed at the Advanced Robotics and Intelligent Control Centre (ARICC) of the Singapore Polytechnic with the team of humanoid robots named Robo-Erectus. Robo-Erectus team has taken part in the RoboCup Humanoid League since 2002 (Zhou & Yue, 2004). The work described here deals with role assignation for the robots, team formation and the relation of strategy and behaviour for the humanoid robots.
The rest of the Chapter is organized as follows. First, a review of related works are presented; part of this work has been done in different robotic platforms. After reviewing these approaches, the proposed method is then presented. Next is an introduction of the Robo-Erectus humanoid robot, the hardware, the control, and the software architecture used for the experiments. Section 5 presents the experiments and results obtained with the proposed approach. These experiments were conducted on a simulator as well as the actual robots. Finally, Section 6 provides concluding remarks about this approach and future work.

2. Literature Review

Cooperative behaviour is an intrinsic feature of the soccer robot competitions. It has been addressed from different points of view; these approaches are based on the capabilities of the robots to perceive the world. The perception of the system would bring advantages and disadvantages and not all these approaches can be shared among the different soccer leagues. For example, the Small Size League of the RoboCup and FIRA use a global vision system that obtains the positions of the robots and the ball from the images. This information is used by a server computer to calculate and send the next positions of all the robots. In this scenario, the cooperative behaviour of the system is conceived by one central orchestrator, who has a full picture of the game. In this work, despite the main focus are humanoid robots, relevant approaches of other leagues are also discussed.

The RoboCup Four-Legged League successfully addressed many aspects of the soccer problem. In this league, four autonomous Sony AIBO robots play in each team. Each robot perceives the environment from a camera and tries to estimate its position as well as the positions of its teammates, foes, and the ball. Cooperative behaviour in this league is an emergent behaviour achieved by all the independent robots in the team. The challenge is to have the group of robots working together towards the same goal, without interfering among themselves, but also supporting their roles.

Previous work in the RoboCup Four-Legged League had addressed the cooperative behaviour problem. Phillips and Veloso presented an approach to coordinate two robots for supporting the team attack. The robots achieved this behaviour by assigning roles to the robots, i.e. attacker and supporter. The supporter robot will position itself to not interfere with the attacker, yet able to receive a pass or recover a lost ball (Phillips & Veloso, 2009). Other researchers have focused on the autonomous positioning of the robots at the beginning of a game or after a goal. Most of the leagues consider it a foul to manually place the robots to resume a game, and each league would assign some kind of penalty to the team that incurs in such situation. In addition, after a goal the robots would have a certain amount of time to self-position themselves before the game is resumed. It is quite common to have robots that played as defenders located in a different region than a defender should be located. If that is the case it might be more meaningful to place the robot behaving as another role just for that specific period of time, instead of having the robots moving back to its defending area, which would require more time than the allowed. Work et. al. proposed a method for player positioning based on potential fields. The method relies on roles that are assigned by a given strategy for the game. The potential fields calculate the shortest paths for the robot to self-position after a goal (Work et al., 2009). Zickler and Veloso proposed random behaviour tactics for the robots in the team. The proposed method can be used to generate a shorter plan in contrast with plans which are too far in the future. The main advantage of the method is the ability of re-planning short plans (Zickler & Veloso, 2009).

Teams in the RoboCup Four-Legged League, Standard Platform League, and Humanoid League have studied the problem of cooperative behaviour from the point of view of player role. A robot with a specific role in the game would contribute to the final objective of the team in a different way. Some teams have addressed the role assignation as a static problem (Acosta et al., 2007) others have addressed the problem as a dynamic assignation. An example of role assignment concept in the planning layer is used in NimbRo humanoid robots (Behnke & Stueckler, 2008). It implements default role negotiation and role switching.

A player can be assigned as a striker, a defender, or a goalkeeper. If only one player is on the field, it plays offensive. When the team consists of more than one field player, the players negotiate roles by claiming ball control. As long as no player is in control of the ball, all players attack. If one of the players takes control, the other player switches to the defensive role. Another application of the role concept is goal clearance by the goalkeeper. The goalkeeper switches its role to field player when the ball gets closer than a certain distance. In this case, it starts negotiating roles with other field players like a standard field player. Thus, the goalie might walk toward the ball in order to kick it across the field.

Coelho et. al. approached the coordination and the behaviour of the robots in a team from the point of view of genetic algorithm (Coelho et al., 2001). They use the genetic algorithms to optimize the coordination of the team. The fitness of the objective function is associated with the solution of the soccer problem as a team, not just as player of the team. The algorithm is flexible in the sense that we can produce different configurations of the team. One drawback of the method is that this process must be done offline and it does not permit online adjustments.

Communication between the Darmstadt Dribblers humanoid robots is used for modelling and behaviour planning (Friedmann et al., 2006). The change of positions of opponents or team members can be realized. A dynamical behaviour assignment is implemented on the robots in such a way that several field players can change their player roles between striker and supporter in a two on two humanoid robot soccer game. This change is based on their absolute field position and relative ball pose.

Risler and von Strik presented an approach based on hierarchical state machines that can be used to define behaviour of the robots and specify how and when the robots would coordinate (Risler & von Strik, 2009). The roles of the robots are assigned dynamically according to the definition given inside the state machines. For example, if a robot is approaching towards the ball with a good chance to score, the robot's role would become striker, and if the state machine defines that only one striker should be in the team, the previous striker would negotiate for another role.

Previous works on cooperation of the behaviour are mainly based on the role of the player. As presented, some works focus on static roles of the players that cannot change during the execution of the game. Others use dynamic assignation of the roles during a game, the criteria are based on the position of the players. The work presented here uses a dynamic role assignation based on the strategy that the team has for the game. Other factors like remaining time and goal difference are used to determine the new formation of the team.

3. Dynamic Role Assignation

On a three versus three humanoid robots game, each robot has assigned a role, e.g. goalie, defender, striker. The fixed or static roles of the robots do not change throughout the whole duration of a robot soccer game, regardless of the game time and goal difference. This method does not cater for scenarios when the team needs to win the game to qualify for the next round, or a premeditated draw game as part of a league game strategy. In some cases the roles of robots are not bounded or limited to an area of the soccer field, where the robots are free to roam in the field. This will cause unnecessary and inefficient movement of the robots.

In a robot soccer 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 flexibility and online adaptation. Thus, fixed roles are not the best approach, even though it is possible to display some level of cooperative behaviour, the system lacks the flexibility to adapt to unforeseen situations. A solution to the fixed roles of the robots is to allow a flexible change of strategies and roles of the robots in a soccer game, according to the game time and goal difference (Acosta et al., 2008). The robot's area of coverage may be limited according to their current roles. This is to allow better deployment and efficiency of the robots movement.

The proposed approach conceived the team as a self-organizing strategy-based decision-making system, in which the robots are able to perform a dynamic switching of roles in the soccer game.

3.1 Cooperative behaviour

In order to achieve a designated goal, agents must work together as a team. However, one thing to remember is that each agent has a different role, and that performing the role is crucial to achieve the desire goal. The changing of roles will be filtered based on three criteria; Strategy, Game Time and Goal Difference respectively.

- Strategy, the strategy to be used for the game will be selected before kick-off and half time of the game. The strategy defines the final objective of the team, and specifies if team should be more offensive or defensive in their play.
- Game Time, the time of the game, 10 minutes for each half of the normal game, and 5 minutes for each half of extra time when required.
- Goal Difference, defined as the difference of own team goal score and opponent team goal score.

The above criteria would then determine a particular formation for the team. Formation here is the number of players for particular roles, not directly the position of the robots on the field. For example, a formation for a team of three robots could be one goalie, one defender, and one striker; another formation could be one goalie and two strikers. In this regard, the formation would specify the number of players with different roles that are required at that moment.

Each role would specify the behaviour of the robot and the region where the robot should be located. However, the robots are free to move around the field, but the specified regions are used as reference for the robots and also for some other behaviour like self-positioning.

The change of formations based on the three criteria has been implemented as a finite state machine on all the robots. The three criteria are updated as follows:

- Strategy, the strategy can only be selected before kick-off and half time of game. The strategy would remain the same for as long as the game last.
- Game Time, the time of the game is kept by each robot, and it is updated when a signal is received from the computer that provides messages like kickoff, stop, etc.
- Goal Difference, it is updated when receive signals from the computer that provides the messages.

The computer that sends the signals for kickoff, stop, resume, etc is known as a referee box. Many leagues in the RoboCup have implemented the use of the referee box with two purposes. First, to standardize the signals that are sent to the teams, the referee box broadcasts the signals to the wireless networks of both teams; and second, to reduce the human interference in the game and to increase the autonomy of the system. The referee box sends signal only when the referee wants to let the robots know about particular situation e.g. a free kick. Most of the referee boxes include some other kind of information that robots are not able to perceive just yet, information such as time and goal difference. In our proposal, we have included both into the referee box messages.

3.2 Strategies

The strategy of a team will define the main objective of the team for a particular game. The strategy is fixed for our approach, which means that it can only be changed if the game stops i.e. half time or of full time (when playing extra time). However, from our experiments we have discovered that it is more meaningful not to change the strategy during a game, unless it is really necessary. While other works have defined some strategies for the games, we have defined four strategies that embrace, in our opinion, all the possibilities of the soccer games.

3.2.1 Normal Game Strategy

The Normal Game strategy is used when the team does not have a specific agenda, which may be used in a friendly game so as not to reveal unnecessary strategies. This strategy uses a Normal Formation throughout the whole game including extra time, regardless of game time and goal difference. Figure 3.1 below illustrates the Normal Game Strategy.

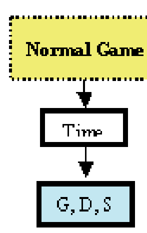


Fig. 3.1 The finite state machine for the Normal Game Strategy.

3.2.2 Must Win Strategy

The Must Win strategy is used when the team has to win the game. The nature of this strategy is more aggressive, with Offensive Formation implemented during the second half of normal game time, and All Out Formation implemented during the second half of extra time, if the team is still losing or draw. When the team is winning, the formations will change to defensive mode to maintain the lead. Figure 3.2 below illustrates the Must Win Strategy.

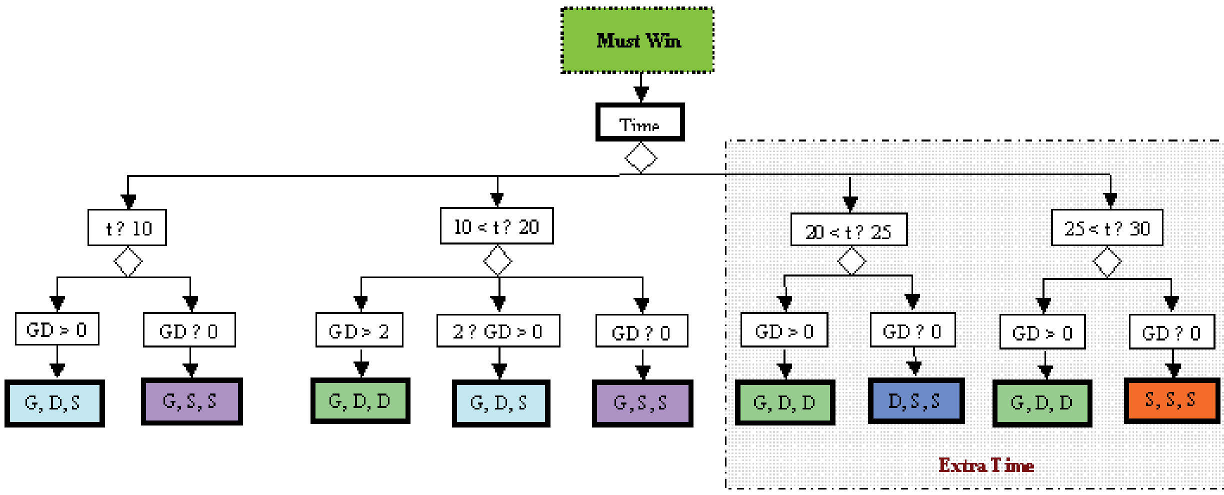


Fig. 3.2 The finite state machine for the Must Win Strategy.

3.2.3 At Least Draw Strategy

The At Least Draw strategy is used when the game strategy is just to aim for a draw or a marginal win. This strategy can be used as a part of first round game, when the team does not want to unnecessarily reveal the full potential of the robots to rival teams. This strategy will implement a normal formation when draw, and a defensive formation when the team is leading. Figure 3.3 below illustrates the At Least Draw Strategy.

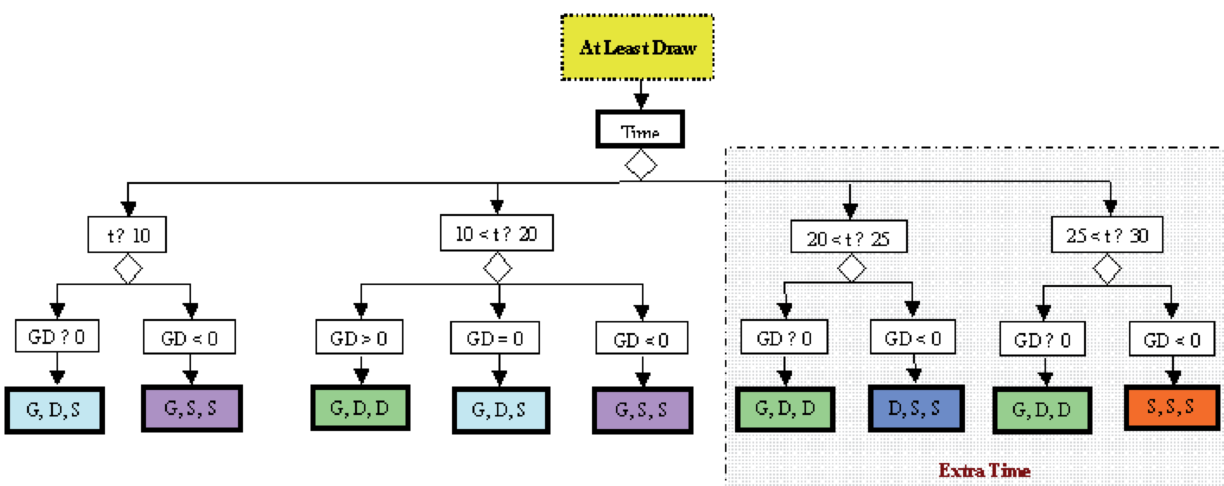


Fig. 3.3 The finite state machine for the At Least Draw Strategy.

3.2.4 Close-Up Strategy

The Close-Up strategy is used to narrow the goal difference when the team is losing to the opponent team. For example, when opponent team scores 10 goals and own team scores 3 goals, this strategy will try to narrow the goal difference to 10 goals versus 6 goals. Figure 3.4 below illustrates the Close-Up Goals Strategy.

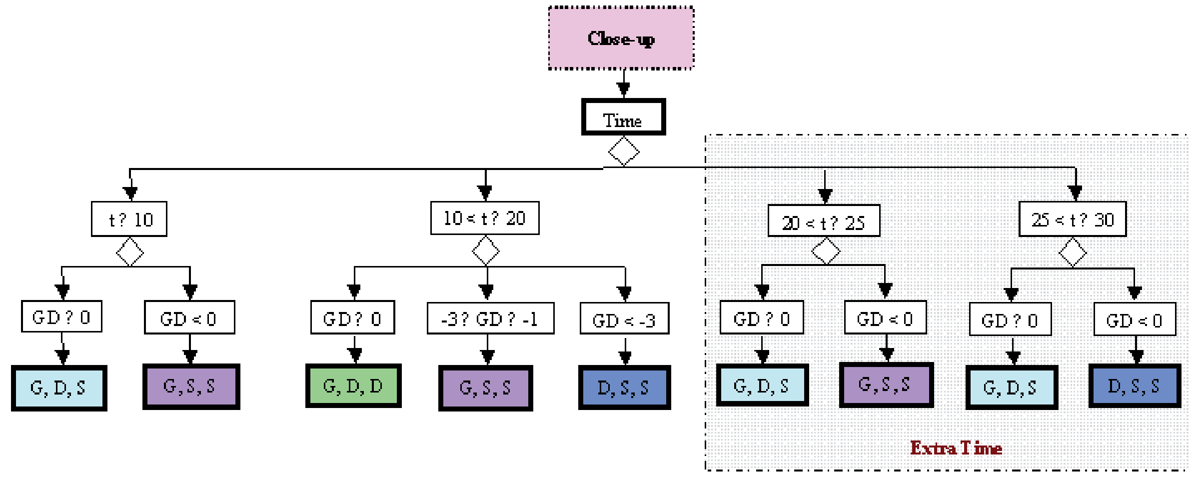


Fig. 3.4 The finite state machine for the Close-up Strategy.

3.3 Negotiation of roles

At several points during the robot soccer game, the robots need to communicate with one another. This may involve informing agents of events or responses, asking for help or information, and negotiating to iron out inconsistencies in information or to agree on a course of action. Negotiation will be necessary when changing of roles is required. This process is complicated since each robot must have its own decisions and there is no central coordinator to assign the roles to each robot.

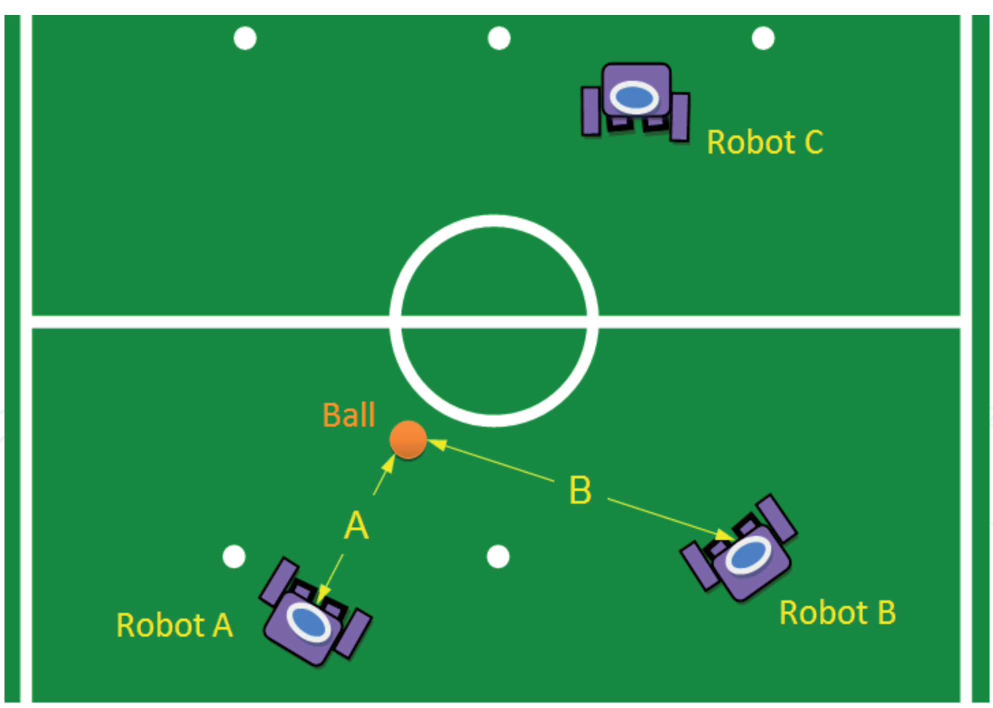


Fig. 3.5 Robots approach through ball approximation

In review of the dynamic roles design, the ball approach method should be taken into consideration. This is for better robot deployment and efficiency during dynamic role change. Team formation change may be based on the positions of the three robots. This could be useful when two robots required negotiating for a particular role. The proximity

and approach to the ball could be used to determine which robot would get the role. This means that if we have a situation as the one presented in Figure 3.5, where robot A and robot B are disputing for the role of striker and defender, how should they solve this situation? There are two criteria employed to solve this problem. First, the robots should evaluate if they are attacking or defending. This is evaluated by determining if the opponents are close to the ball, in that case it is considered that the team is defending; otherwise, it is considered that the team is attacking. Second, the robots will evaluate the distance to the ball. If the robot believes that its distance is shorter than that of the other robot, it will approach to the ball and win the role, i.e. defender when the team is defending or striker when attacking.
The robots in the field can approach the ball based on proximity, the robot nearer to the ball will move forward. As illustrated in Figure 3.5, Robot A will approach the ball rather than Robot B due to its proximity, as distance A is shorter compared to distance B. This is taken into consideration that both Robot A and B heading is facing the ball. As each robot has the ability of knowing its own heading orientation, the ball approach method considers the heading of the robot. Illustrated in Figure 3.6 below, Robot A is heading away from the ball, while Robot B is heading towards the ball. Although Robot A distance is nearer compared to Robot B, however it will take time for Robot A to turn its heading position towards the ball. Hence Robot B will approach the ball instead, while Robot A will proceed with heading adjustment. This method is limited to a difference in distances, defined by a threshold, in comparison to the other nearby robots. If the ball distance for Robot A is less than the threshold difference compared to Robot B, then Robot A will still adjust its heading and approach the ball.

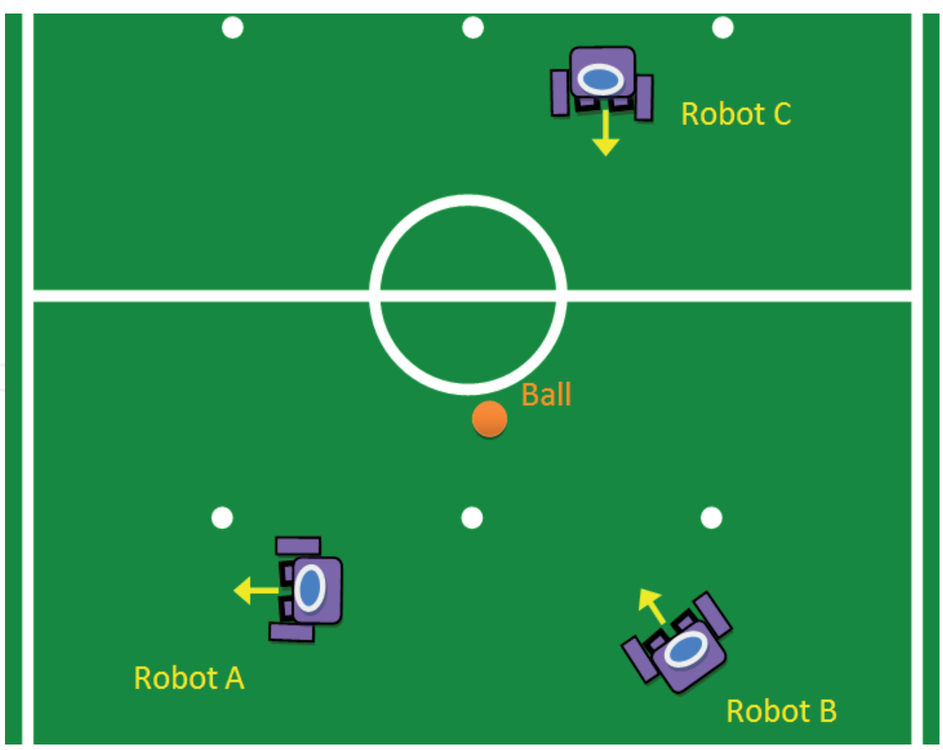


Fig. 3.6 Robots approach through ball approximation and heading.

Figure 3.7 below illustrates the close-up view of the robots approach through ball approximation and heading. With the robot heading perpendicular to the ball as the 90°

heading reference, robot heading 181° to 359° will not initiate the approach. As illustrated below, Robot A heading falls out of the 0° to 180° heading range. While Robot B distance is less or equal to the threshold, therefore Robot B will proceed to approach the ball, and will change its role accordingly e.g. from defender to striker.

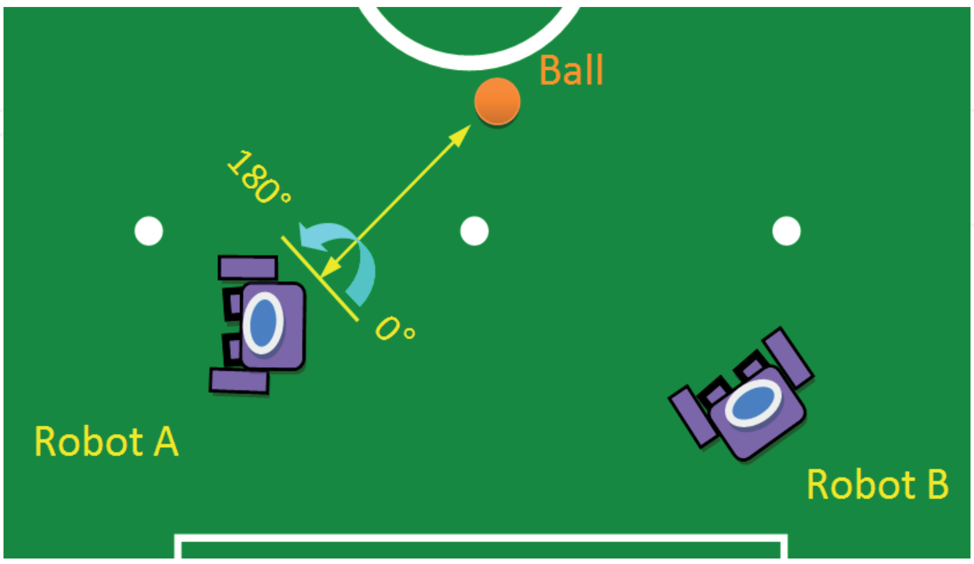


Fig. 3.7 Robots approach through ball approximation and heading.

3.4 Formations, roles, and players

During a game there could be situations where a team must play with substitute robots, and occasionally the team must play with fewer players. To deal with these situations, the proposed formations have roles with priorities. These priorities indicate which roles must be filled first, and which roles must always remain filled. Each robot keeps a list of the other robots broadcasting in its network; with this information each robot is aware of the number of team members of the team. However, when a robot that is currently playing fails and needs to be replaced by another robot. This new robot will be added to the list, the previous robot will identify that there are four robots in the list when this happen, the robots monitor the messages to discover which robot went dead. If a robot is not able to discover the missing robot, it will broadcast a request, so that the playing robots will reply. This request is also broadcasted if one robot does not broadcast any message for a period of time. This is done to try to identify any dead robot. When a robot is faulty and its role is a priority one after the replacement is in the field the robot will renegotiate their roles. In the scenario that there is no replacement, a robot with a non-priority role will switch to the priority role. The Table 3.1 below presents the different formations and the priority of the roles.

Formation	Highest	High	Low
Defensive	Goalie	Defender	Defender
Normal	Goalie	Defender	Striker
Offensive	Goalie	Striker	Striker
Super Offensive	Defender	Striker	Striker
All Out	Striker	Striker	Striker

Table 3.1 Formations and priorities of the roles per formation.

3.5 Area of Coverage

The proposed method also defines the area of coverage or region where the robots should be limited according to their roles. This is to facilitate effectiveness and prevent unnecessary movement during the role changing. This roles area of coverage proposal is to enhance the ball approach criteria described in the previous Section 3.4. Figure 3.8 shows the areas of coverage for the roles of goalie, defender and striker. These areas are also used for the self-positioning behaviour, the robots defined points on the field as starting positions, but they have a higher priority to be inside the area rather than to reach the point. Only during the self-positioning behaviour the area of coverage of a striker becomes the same as that of the defender, but the attraction points are different; i.e. for the striker is closer to the half line.

- Goalie - Movement limited to the Goal Area. This is to cater for cases when the ball stops just at the goal line, and the goalie has to move behind the goal line to kick out the ball, or else the goalie would not know how to react in a situation when ball stops just at the goal line.
- Defender - Movement limited to the own half of the field, as illustrated in Figure 3.8(b) below. The defender's area also includes the goal area.
- Striker - Movement limited from the middle of lower half of field, to the opponent's goal, as illustrated in Figure 3.8(c). However, when the formation has more than one striker, the striker's area becomes the whole field.

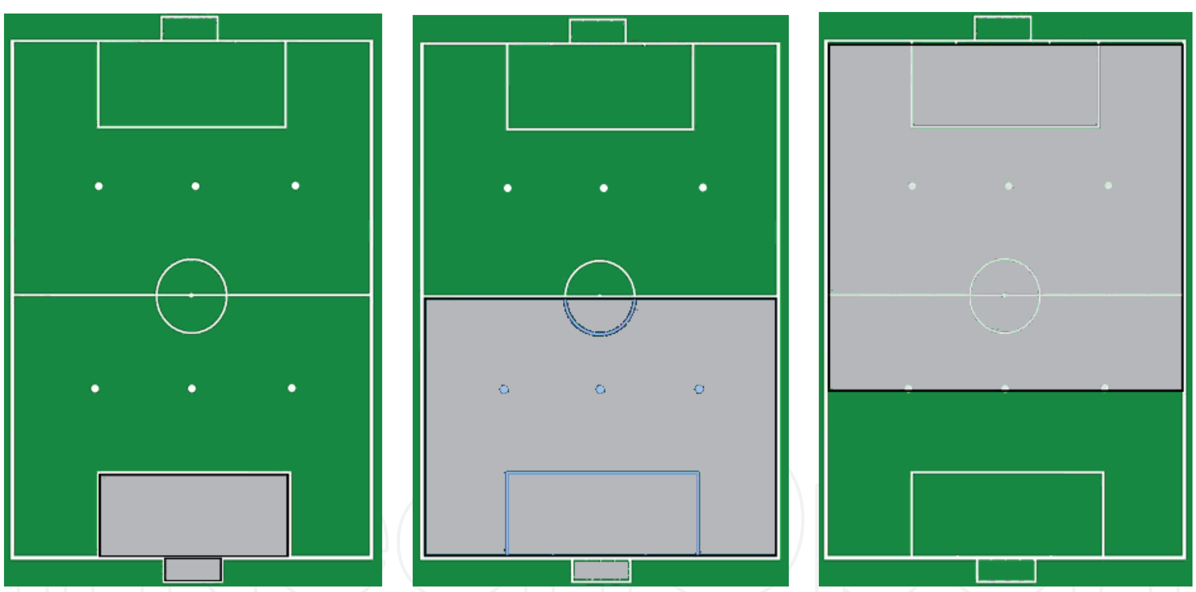

(a) (b) (c)

Fig. 3.8 Area of coverage for roles of (a) goalie, (b) defender, and (c) striker.

4. Robo-Erectus, The Humanoid Soccer Robot

The Robo-Erectus project (www.robo-erectus.org) has been developed in the Advanced Robotics and Intelligent Control Centre (ARICC) of Singapore Polytechnic. The humanoid robot Robo-Erectus is one of the pioneering soccer-playing humanoid robots in the RoboCup Humanoid League (Acosta et al., 2007). Robo-Erectus has collected several awards since its first participation in the Humanoid League of RoboCup in 2002. Robo-Erectus won the 2nd place in the Humanoid Walk competition at the RoboCup 2002 and got 1st place in the Humanoid Free Performance competition at the RoboCup 2003. In 2004, Robo-Erectus won the 2nd place in Humanoid Walk, Penalty Kick, and Free Performance. In 2007, it finished 6th in the 2 vs. 2 games, and 3rd in the technical challenge. In the RoboCup 2009, it qualified to the second round of round robin of 3 vs. 3 games.

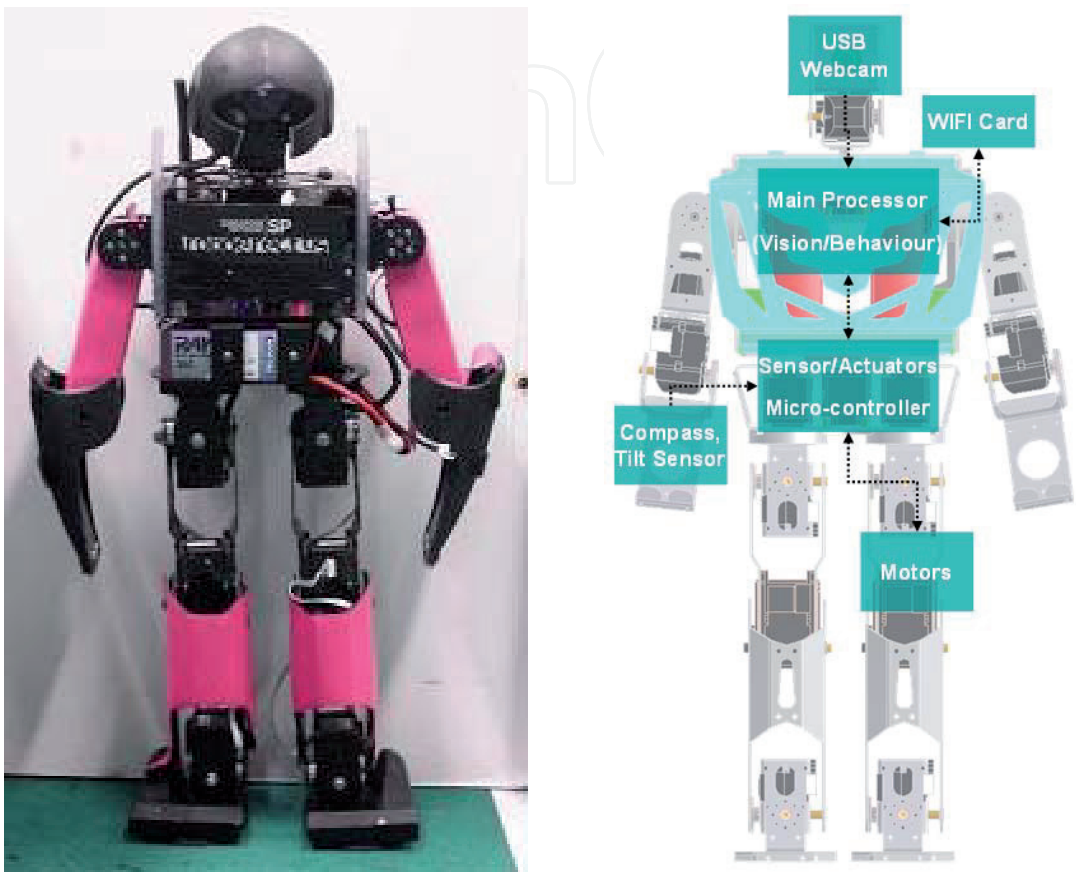


Fig. 4.1 REJr-X1, the latest generation of the family Robo-Erectus.

The Robo-Erectus project aims to create a humanoid robot that can be used for teaching, research and competition. After the introduction of the TeenSize to the RoboCup Humanoid League, the team has developed two parallel types of Robo-Erectus. Robo-Erectus Junior is the name for the robots that take part in the RoboCup Humanoid KidSize, as a group of at least three robots, where the restriction is that the robots must be less than 60cm in height. The second type of Robo-Erectus is called Robo-Erectus Senior and this robot competes in the RoboCup Humanoid TeenSize, in which only one robot takes part, whereby the restriction of the league is that the robots should be more than 1m tall.

The proposed method in this Chapter applies to a team of humanoid kidsize robots. Therefore, the rest of this Section will focus on the Robo-Erectus Junior. Each robot in the team has the same hardware and software features as described here.

4.1 Robo-Erectus Junior

The latest version of Robo-Erectus named Robo-Erectus Junior-X1 (REJr-X1), as shown in Figure 3, has been designed to be fully autonomous and to deal with the challenges of the 3 vs. 3 games.

Figure 4.1 shows the design of the humanoid robot REJr-X1. The skeleton of the robot is constructed with aluminium braces, the head and arms of the robot are made of plastic. Despite its simplicity, the mechanical design of the robot is robust and lighter than their predecessors. Its human-like body has a height of 52cm and weight of just 3.3kg, including batteries.
Robo-Erectus Junior has a total of 21 degrees of freedom. Table 4.1 shows the body parts and their associated degrees of freedom. Each degree of freedom uses as actuator a Dynamixel DX-117 Digital Servomotor. These servomotors have a typical torque of 28.89kg·cm and a speed of 0.172sec/60°. Each knee joint uses a Dynamixel RX-64 Digital Servomotor that provides a higher torque than that of DX-117. Each smart actuator has a micro-controller in charge of receiving commands and monitoring the performance of the actual motor. An RS485 serial network connects all the servomotors to a host processor, which sends positions and receives the current data (angular positions, speed, voltage, and temperature) of each actuator.

Body Part	Roll	Pitch	Yaw
Head		✓	✓
Body			✓
Shoulder	✓	✓	
Elbow		✓	
Hip	✓	✓	✓
Knee		✓	
Ankle	✓	✓	

Table 4.1 List of Degrees of Freedom for the humanoid robot Robo-Erectus Jr.

The control system of the Robo-Erectus Jr-X1 consists of a network with two micro-processors and several devices. Instead of using a single processor to deal with sensory data, processing and actuators control, Robo-Erectus uses dedicated processors for particular tasks. This design was implemented firstly in the Robo-Erectus Junior-AX and it has proven to improve the performance of the system (Acosta et al., 2008). Below are listed the tasks of each processor:

1. The main processor is responsible to coordinate all behaviours of the robot. It receives the processed sensorial information from the sensor-motor processor, which is used to take decisions and finally send the motors commands back to the sensor-motor processor. This processor has a wireless network interface that allows communication with the other robots and a computer that sends game signals. This processor also processes the images from an USB camera.
2. The sensor-motor processor is a dual DSP microcontroller, which receives the motor commands from the main processor. These commands are parameters for the gait generator. The gaits of the humanoid robot are generated online and finally sent to the servo-motors by a RS485 bus. The motor feedback is collected, processed, and sent back to the main processor by a RS232 bus. The data of servo-motors is updated every 16.6 ms. Along with the motor information, this processor also collects data from the accelerometers and gyroscopes before passing these data to a Kaman filter to produce more relevant information, about the state of the robot.

4.2 Virtual RE, The Robo-Erectus Junior Simulator

Robo-Erectus is equipped with sensors and actuators that allow it to navigate autonomously and display an intelligent behaviour. The robot uses a camera as main sensor, and it is able to perceive different colours and to track them. It also contains a dedicated processor to control the behaviour of the robot, wireless communication with the control PC and the teammates, and a sub-system to control sensors and actuators. A simulator was developed to simulate all the sensor of the robot. In addition the simulator is able to simulate two teams with a number variable of robots to reproduce a game (see Figure 4.2). The simulator uses the server-client framework, where each robot is considered a server and a single client could connect to it and control it.

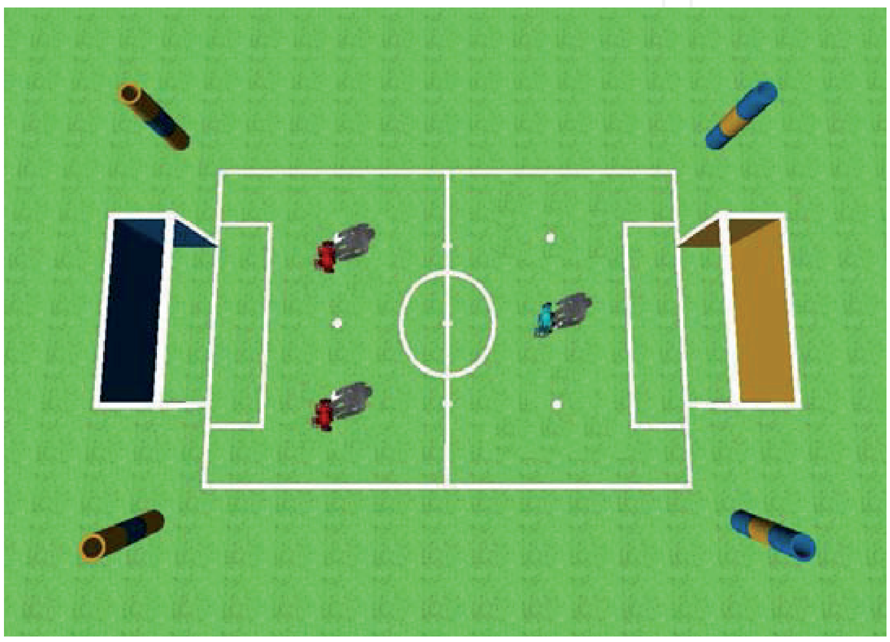


Fig. 4.2 Virtual RE the simulator for the Robo-Erectus Jr.

The simulator of Robo-Erectus Junior-AX, called Virtual-RE, was also used during these experiments. Virtual-RE provides several possibilities of visualization and interaction with the simulated worlds (Acosta et al., 2007). To simulate rigid body dynamics, Virtual-RE uses the Open Dynamics Engine (ODE), which has a wide variety of features and has been used successfully in many other projects. The visualization, as well as the computation of imaging sensor data is based on OpenGL libraries, because this standard offers the best performance with modern hardware on different platforms.

5. Experimental results

In order to test the effectiveness of our algorithms, we performed two sets of experiments: one with the simulator Virtual-RE and the other with the actual Robo-Erectus Jr robots. All the sets involve the four strategies as well as winning and losing scores for both the regular time and the extra time. In addition, each set contains games with faulty players, some with substitute robot and others without.

The behaviours of each role have been implemented in a finite state machine, with simple actions for each robot and self-positioning. The opponents for all the sets played with static formation: goalie, defender, striker.

Table 5.1 shows the resulting analysis of the data. The data presented is the percentage of wrong formation during the possible changes in the game. The possible changes happen every five minutes or after a goal. The 7% of wrong role selection in the simulation is due to situation where two robots decided to have the same role. In all these cases, the role was successfully corrected after one or two seconds to conclude their negotiation. In the case of the Robo-Erectus Jr robots the percentage was higher, because of some unexpected situations with a few robots and they have to be taken out, serviced and their computer rebooted. This situation caused a wrong role assignment, however the robots corrected their roles after negotiation. In a similar way, the wrong formation is due to the faulty robot.

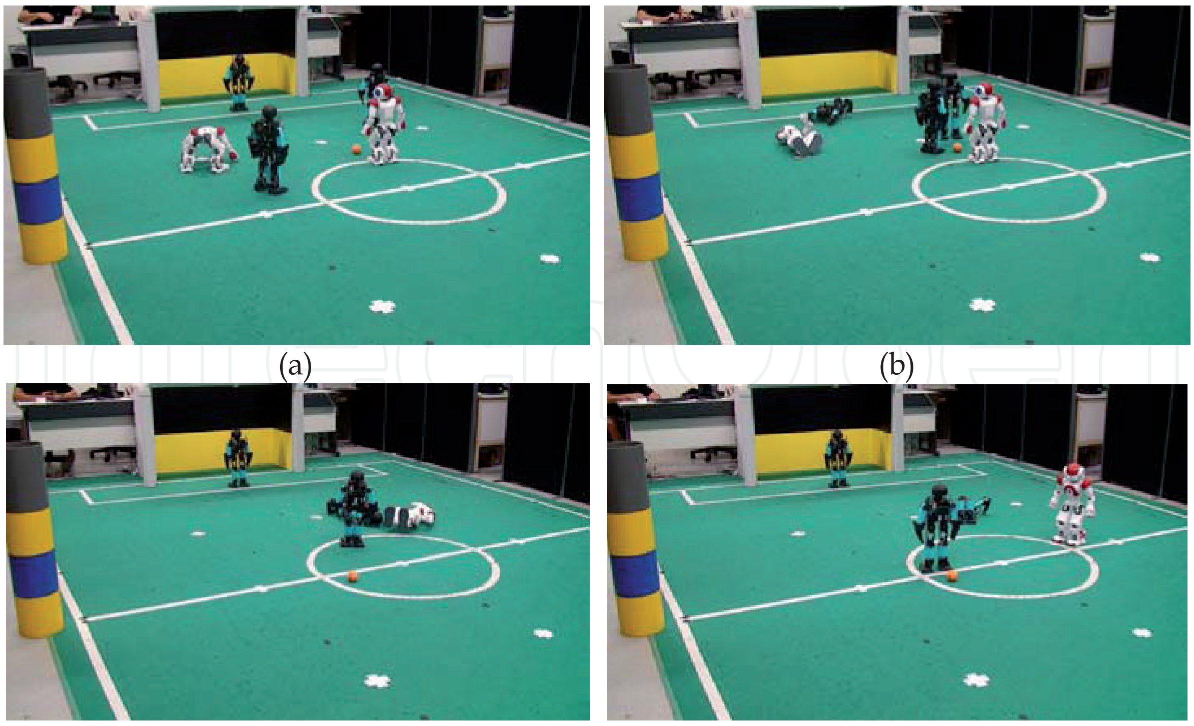
Set	Wrong Formation	Wrong Role
Simulation	0%	7%
Robo-Erectus Jr	5%	18%

Table 5.1 Percentage of wrong formation and wrong role selection for possible changes without faulty robots.

Table 5.2 shows the wrong formation and wrong role selection with faulty robot with and without replacement. For the replacement sets, the problem was similar to the one observed in the Robo-Erectus Jr in the previous experiments due to the faulty robot. In both simulation and real robot experiments, the robots managed to correct their wrong roles after few seconds. For the no replacement set, all the robots managed to switch to a higher priority role, except for one trial where the actual robot maintained its role, but this was because the robot's program entered into a deadlock and has to be reset.

Set	Wrong Formation	Wrong Role
Simulation (Replacement)	7%	7%
Robo-Erectus Jr (Replacement)	15%	23%
Simulation (No Replacement)	0%	0%
Robo-Erectus Jr (No Replacement)	0%	7%

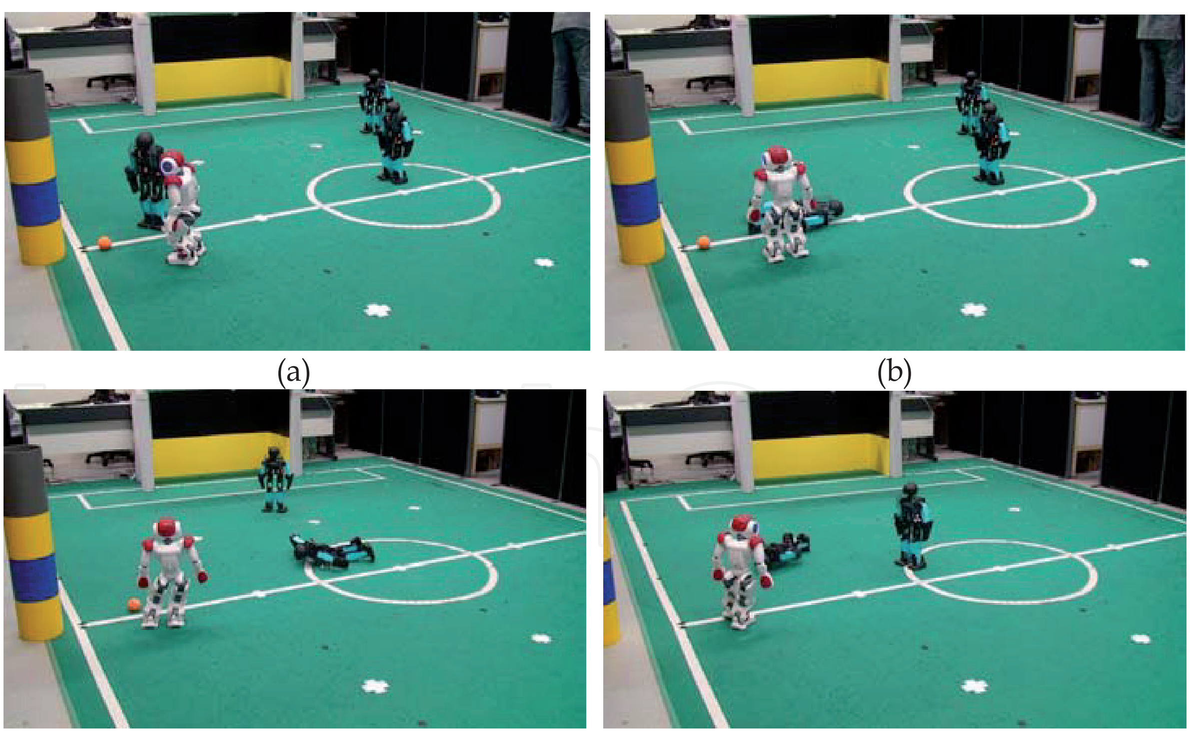
Table 5.2 Percentage of wrong formation and wrong role selection for possible changes with faulty robots.



(a) (b)

(c) (d)

Fig. 5.1 Sequence of the game with formation At Least Draw with formation one Goalie and two defenders.



(a) (b)

(c) (d)

Fig. 5.2 Sequence of a game with a strategy Must Win with a formation three strikers. Experiment with one faulty robot.

In this set of experiments, the Robo-Erectus Jr played against a team of Nao robots with a fixed formation of goalie, defender and striker. Figure 5.1 shows a sequence of the game with Nao robots with the strategy of At Least Draw, with a formation goalie, defender, defender. In Figure 5.2 the strategy employed is Must Win with an All Out formation. In this game one robot was taken out to simulate a faulty robot. Robot in Figure 5.2(b) is taken out as can be seen in Figure 5.2(c). In Figure 5.2(c) both robots try to approach to the ball.

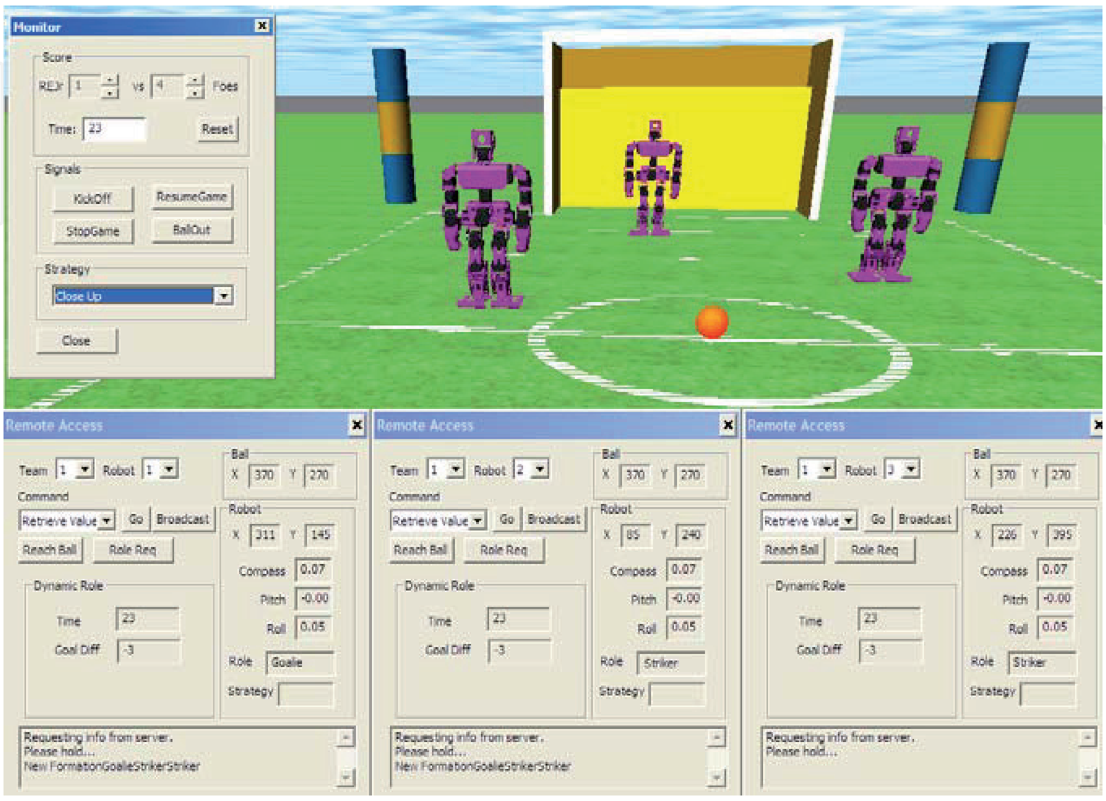


Fig. 5.3 Screenshot from the Virtual-RE with a Close-up Strategy and a formation Goalie and two strikers. This is during the first half of extra time.

The set of experiments with the simulator Virtual-RE were conducted in a similar way as the real Robo-Erectus Jr, i.e. three versus three games with different strategies (see Figure 5.3). Due to the flexibility that the simulator offered, we were able to set, besides the normal game, scenarios where the robots were placed in certain positions, with specific roles, score, and time; to test situations like approaching to the ball, or self-positioning. For the self-positioning behaviour 85% of trials was successful. This is because we placed the striker inside the goal keeper area or in one of the corners, and the robot should go back to its half of the field, in the way back the robot should avoid other robots, but sometimes it collides with them, falling and not reaching the position on time.

6. Conclusion

Humanoid robot soccer is getting popular recently, and the advances in the technology have made it possible to see exciting displays from the humanoid robots. Nevertheless, most of the teams have achieved a position where the robots are able to show several skills. It is challenging to have a team of robots displaying team behaviour without interfering with each other and supporting their objectives. The work presented here uses a dynamic role assignation but is based on the strategy that the team has for the game. Besides, other factors like remaining time and goal difference are used to determine the new formation of the team. The roles of the players underlie team behaviour, while the strategy defines the objective of the team. Experimental results support our proposal, showing that the percentage of wrong role selection among the robots is low less than 20%. Results also prove that this wrong role selection is soon corrected by the robots after negotiation. Future work is to deal with situations of negotiation when having more players, and to define more team behaviours with this framework.

7. References

Acosta Calderon, C.A.; Zhou, C.; Yue, P.K.; Wong, M. & Mohan, R.E. (2007). A Distributed Embedded Control Architecture for Humanoid Soccer Robots. *Proceedings of Advances in Climbing and Walking Robots*, pp. 487-496, Singapore, July 2007.

Acosta Calderon, C.A.; Mohan, R.E. & Zhou, C. (2007). A Humanoid Robotic Simulator with Application to RoboCup, *Proceedings of IEEE Latin American Robotic Simposium*, Mexico, November 2007.

Acosta Calderon, C.A.; Mohan, R.E.; Zhou, C.; Hu, L.; Yue, P.K. & Hu, H. (2008) A Modular Architecture for Soccer Robots with Distributed Behaviour Control, *International Journal of Humanoid Robotics*, Vol. 5, No. 3, September, pp. 397–416.

Behnke, S.; & Stueckler, J. (2008). Hierarchical Reactive Control for Humanoid Soccer Robots. *International Journal of Humanoid Robots*, Vol 5, No 3, September, pp. 375-396.

Coelho, A.L.V.; Weingaertner, D. & Gomide, F.A.C. (2001). Evolving Coordination Strategies in Simulated Robot Soccer. *Proceeding of the 5th International Conference on Autonomous Agents*, pp. 147-148, 1-58113-326-X, Montreal, Canada, May-June 2001.

Friedmann, M.; Kiener, J.; Petters, S.; Thomas, D. & von Stryk, O. (2006). Modular Software Architecture for Teams of Cooperating, Heterogeneous Robots. *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pp 24-29, Kunming, China, December 2006.

Kitano, H. & Asada, H. (2000). The RoboCup Humanoid Challenge As The Millennium Challenge for Advanced Robotics. *Advanced Robotics*, Vol. 13, No. 8, pp723–736.

Phillips, M. & Veloso, M. (2009). Robust Support Role in Coordinated Two-Robot Soccer Attack, In: *RoboCup 2008, LNAI 5399*, Iocchi, L, Matsubara, H., Weitzenfeld, A. & Zhou, C., pp. 235-246, Springer-Verlag, 3-642-02920-5, Germany.

Risler, M. & von Stryk, O. (2008). Formal Behavior Specification of Multi-robot Systems Using Hierarchical State Machines in XABSL. *Proceedings of AAMAS08-Workshop on Formal Models and Methods for Multi-Robot Systems*, Estoril, Portugal, May 2008.

Work, H.; Chown, E.; Hermans, T.; Butterfield, J. & McGranaghan, M. (2009). Player Positioning in the Four-Legged League, In: *RoboCup 2008, LNAI 5399*, Iocchi, L, Matsubara, H., Weitzenfeld, A. & Zhou, C., pp. 391-402, Springer-Verlag, 3-642-02920-5, Germany.

Zickler, S. & Veloso, M. (2009). Playing Creative Soccer: Randomized Behavioral Kinodynamic Flanning of Robot Tactics, In: *RoboCup 2008, LNAI 5399*, Iocchi, L, Matsubara, H., Weitzenfeld, A. & Zhou, C., pp. 414-425, Springer-Verlag, 3-642-02920-5, Germany.

Zhou, C. & Yue, P.K. (2004). Robo-Erectus: A Low Cost Autonomous Humanoid Soccer Robot. *Advanced Robotics*, Vol. 18, No. 7, pp. 717–720.