Mechatronics xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Robot team coordination using dynamic role and positioning assignment and role based setplays

Nuno Lau *, Luis Seabra Lopes, Gustavo Corrente, Nelson Filipe, Ricardo Sequeira

Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA), Dep. Electrónica Telecomunicações e Informática (DETI), Universidade de Aveiro, Portugal

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Multi-robot team coordination
Strategic positioning
Dynamic role assignment
Coordinated procedures

ABSTRACT

The coordination methodologies of CAMBADA, a robotic soccer team designed to participate in the RoboCup Middle-Size League (MSL), are presented in this paper. The approach, which relies on information sharing and integration within the team, is based on formations, flexible positionings and dynamic role and positioning assignment. Role assignment is carried out locally on each robot to increase its reactivity. Positioning assignment is carried out at a lower frequency by a coach agent following a new priority-based algorithm that maintains a competitive formation, covering the most important positionings when malfunctions lead to a reduction of the team size. Coordinated procedures for passing and setplays have also been implemented. With this design, CAMBADA reached the 1st place in RoboCup'2008 and the 3rd place in RoboCup'2009. Competition results and performance measures computed from logs and videos of real competition games are presented and discussed.

1. Introduction

As robots become increasingly available in different areas of human activity, researchers are naturally prompted to investigate how robots can cooperate with each other in order to perform different tasks. Moreover, progress in wireless communication technologies enables information sharing and explicit coordination between robots. These are basic capabilities needed to support sophisticated cooperation and coordination algorithms. Given this increasing availability of robots and communication technologies, multi-robot systems have, in the last two decades, been receiving more and more attention from researchers [1–3].

Interest on multi-robot systems is further justified by the advantages they offer with respect to single robots. First, some tasks are simply too difficult or impossible to be carried out by a single robot. In other cases, by providing a larger work force, multi-robot systems can carry out tasks faster. Multi-robot systems also facilitate scalability, as larger problems can often be solved by adding more robots to the team. Finally, through their inherent redundancy, multi-robot systems offer robustness, as they may still work when a team member is damaged or malfunctioning.

These advantages make multi-robot systems useful in a variety of domains, such as exploration of unknown or changing environments [4–6] (including such diverse applications as ecological monitoring, rescue, de-mining or planetary exploration), mapping [7], foraging [8], transportation [9], manufacturing [10], intrusion detection and patrolling [11,12], or even entertainment [13].

The development of multi-robot systems raises many new research issues, not found in isolated robots. These new issues are concerned with how the individual robots can coordinate their actions to carry out the assigned tasks as efficiently as possible. Among other issues, the following can be mentioned: How are different sub-tasks assigned to different robots [14,8,15]? How can different roles be assigned to different robots [16–18]? If robots need to move in formation, how can the formation be controlled [2,19,20]? How can multi-robot plans be generated and/or executed [21,22]? Which information should robots exchange in order to enable coordination [23,24]? How can multi-robot systems be debugged [25,26]?

The authors have been addressing several of these issues in the robotic soccer domain, currently a popular scenario and application for research in multi-robot systems. In particular, the authors contributed to the development of CAMBADA, a RoboCup Middle-Size League (MSL) team (Fig. 1). The MSL is one of the most challenging leagues in RoboCup. Robotic players must be completely autonomous and must play in a field of 12 m × 18 m [27]. Teams are composed of at most five robots with a maximum height of 80 cm. Human interference is allowed only for removing malfunctioning robots and re-entering robots in the game.

Building a team for the MSL is a very challenging task, both at the hardware and software levels. To be competitive, robots must be robust, fast and possess a comprehensive set of sensors. At the software level they must have an efficient set of low-level skills and must coordinate themselves to act as a team. Research

---

* Corresponding author.

E-mail addresses: nunolau@ua.pt (N. Lau), lsl@ua.pt (L.S. Lopes), gustavo@ua.pt (G. Corrente), nelson.filipe@ua.pt (N. Filipe), rps@ua.pt (R. Sequeira).

0957-4158/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2010.05.010

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

Fig. 1. CAMBADA robotic team.

conducted within CAMBADA has led to developments concerning hardware [28], computational and communications infrastructure [29–31], vision system [32,33], monitoring/debugging [25] and high-level deliberation and coordination [24,34]. This paper focuses on the last aspect, providing a detailed and up-to-date account of the currently used algorithms and their performance.

The complexity inherent to the MSL and, in particular, the difficulty of developing robots with robust sensorimotor capabilities and informative perception capabilities explains why most teams have implemented relatively simple coordination capabilities. The more advanced teams achieve coordination through the assignment of different roles to the robots [35,36,18]. Typically there is, at least, an attacker, a defender, a supporter and a goalie. As perception and sensorimotor capabilities become more sophisticated it will be possible to develop more sophisticated coordination algorithms. This trend is pushed further by the increase in team size (number of robots) as well as field size. A natural source of inspiration is the RoboCup Soccer Simulation League, where teams have been using coordination layers with strategy, tactics and formations [37,16], coordination graphs [38] and reinforcement learning [39,40].

CAMBADA participated in several national and international competitions, including RoboCup world championships (5th place in 2007, 1st place in 2008, 3rd pace in 2009) and the Portuguese Open Robotics Festival (3rd place in 2006, 1st place in 2007, 2008 and 2009). The excellent results obtained in RoboCup'2008 and RoboCup'2009 are largely due to the developed coordination methodologies, as the CAMBADA robots are among the slowest in the international competitions.

This paper is organized as follows: Section 2 presents the hardware and software architectures of CAMBADA players and provides details on the main software components involved in individual decisions of the players. Section 3 describes how players share information with teammates and how they integrate shared information. Sections 4 and 5 describe the adopted coordination methodologies. Section 6 presents and discusses competition results and various performance measures. Section 7 concludes the paper.

2. Player architecture

CAMBADA robots (Fig. 1) were designed and completely built at the University of Aveiro. Each robot fits into a cylindrical envelope with 485 mm in diameter. The mechanical structure of the players is layered and modular. Each layer can easily be replaced. The components in the lower layer, namely motors, wheels, batteries and an electromechanical kicker, are attached to an aluminum plate placed 8 cm above the floor. The second layer contains the control electronics. The third layer contains a laptop computer, at 22.5 cm from the floor, a catadioptric omnidirectional vision system, a frontal vision system (single camera) and an electronic compass, all close to the maximum height of 80 cm.

The players are capable of holonomic motion, based on three omnidirectional roller wheels. With the current motion system, the robots can move at a maximum speed of 2.0 m/s. As mentioned, this is less than in many of the other MSL teams, which can currently move at speeds typically between 2.5 and 4.0 m/s (e.g. [41–44]). The mentioned vision system allows detecting objects, the ball, players, and field lines on a radius of 5 m around each player. The frontal camera allows detecting the ball further away but is currently not used due to software stability problems encountered when using both cameras simultaneously. Each player also carries encoders, battery status sensors and, for detecting if the ball is kickable, an infra-red presence sensor.

The computational system in each robot is a set of processing nodes (several small microcontrollers for basic perception and sensorimotor control plus a laptop for high-level deliberation) connected through a Controller Area Network (CAN). All communications within the team are based on the standard wireless LAN protocol IEEE 802.11x profiting from large availability of complying equipment. The team receives referees instructions through a wired LAN TCP link.

On the main processing node (laptop), CAMBADA players run several software processes that execute different activities, such as image acquisition, image analysis, integration/deliberation and communication with the low-level modules (Fig. 2). The order and schedule of activation of these processes is performed by a so-called process manager (Pman [31]). Pman stores the characteristics of each process to activate and allows the activation of recurrent tasks, settling phase control (through the definition of temporal offsets), precedence restrictions, priorities, etc. The Pman services allow changes in the temporal characteristics of the process schedule during run-time.

The top-level processing loop starts by integrating perception information gathered locally by the player. This includes information coming from the vision processes, odometry information coming from the holonomic base, compass information and ball presence information. All this information is stored in a shared data structure called Real-Time Data Base (RTDB) [29]. The RTDB has a local area, shared only among local processes, and a global area, where players share their world models to the other players. The global area is transparently updated and replicated in all players in real-time. Every 100 ms the shared area of the RTDB of each robot (and of the coach) is communicated to the other robots using

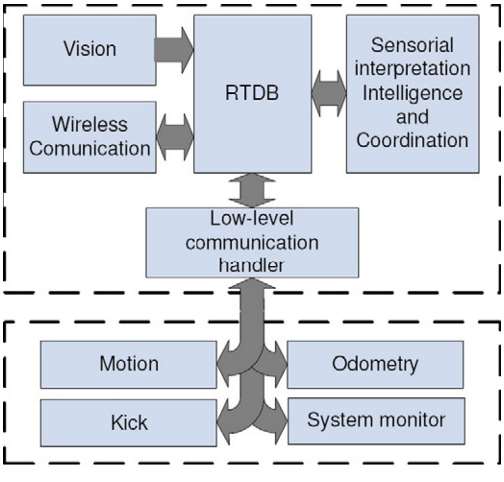

Fig. 2. Layered software architecture of CAMBADA players, from [28].

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

UDP multicast. Sending times are chosen using an adaptive TDMA algorithm that tries to avoid collisions among packets [45].

Selflocalization uses a sensor fusion engine based on the publicly available engine described in [46]. By integrating information from field line detection, this engine produces self position estimates with a high level of confidence. Compass information is used to resolve ambiguities and detect self-localization errors. The final fusion step is to integrate local information with information shared by teammates. After this integration, part of the world state is written to the global area of the RTDB.

Deliberation in CAMBADA considerably relies on the concepts of role and behavior. Behaviors are the basic sensorimotor skills of the robot, like moving to a specific position or kicking the ball. The set of behaviors that are implemented in the CAMBADA agent are adapted to its catadioptric omnidirectional vision and holonomic driving systems. The combination of these technologies enhances the set of possible behaviors when compared to a differential drive robot or to an holonomic drive robot with a limited field of view. In brief, the current set of behaviors is the following:

bMove uses two symbolic parameters: the target position where to move; and the position which the CAMBADA player should be facing in its path to the target. The symbols used are OBall, TheirGoal and OurGoal. This behavior may activate the functions of avoiding obstacles and avoiding the ball (used during the game repositions to avoid collisions with the ball).

bMoveToAbs is another moving behavior; it allows the movement of the player to an absolute position in the game field, and also allows the player to face any given position. Obstacle avoidance is also included.

bPassiveInter moves the player to the closest point in the ball trajectory and waits there for the ball.

bDribble is used to dribble the ball towards a given relative player direction.

bCatchBall is used to receive a pass. The player aligns itself with the ball path and, when the ball is close, moves backwards to soften the impact and more easily engage the ball.

bKick is used to kick the ball accurately to one 3D position, either for shooting to goal or passing to a teammate. Preparing for the kick involves determining the kick direction and power. Polynomial functions, whose coefficients were determined by experimentation, are used to compute kick power based on distance to target. Different functions are used according to the expected number of ball bounces, given the distance.

bGoalieDefend is the main behavior of the goalie.

Roles select the active behavior at each time step. During open play, the CAMBADA agents use only three roles: `RoleGoalie`, `RoleSupporter` and `RoleStriker`. The `RoleGoalie` is activated for the goalkeeper. Further details about the developed roles and respective coordination mechanisms will be presented in Sections 4 and 5.

Another important component of the deliberation process in CAMBADA is based on a coach agent that runs in an external computer. The coach communicates with the robot agents using the RTDB. This agent is used to define the positionings of the robots inside the current formation, as will be explained in Section 4.3.

3. Information sharing and integration

Sharing perceptional information in a team can improve the accuracy of world models and, indirectly, the team coordination [23]. Therefore, information sharing and integration is one of the key aspects in multi-robot teams.

In CAMBADA, each robot uses the information shared by the other robots, obtained through the RTDB, to improve its knowledge about the current positions and velocities of the other robots and of the ball. It is very important for our coordination model that each robot keeps an accurate estimate of the absolute position of the ball, its own position and its teammates positions. The role assignment algorithm is based on the absolute positions of the ball, the robot and its teammates. The teammates positions are not obtained through the vision system. They are obtained from the teammates themselves through the RTDB.

Each agent communicates its own absolute position and velocity to all teammates as well as its ball information (position, velocity, visibility and engagement in robot), current role and current behavior.

Multi-robot ball position integration has been used in the Middle-Size League by several teams [35,47]. In CAMBADA, multi-robot ball position integration is used to maintain an updated estimate of the ball position, when the vision subsystem cannot detect the ball, and to validate robot's own ball position estimate, when the vision subsystem detects a ball.

Currently, a simple integration algorithm is used. When the agent does not see the ball, it analyzes the ball information of playing teammates. The analysis consists in the calculation of the mean and standard deviation of the ball positions, then discarding the values considered as outliers of ball position, and finally using the ball information of the teammate that has a shorter distance to the ball. To determine if the agent sees a fake ball, i.e., to validate the robot's own perception, we use a similar algorithm.

Communication is also used to convey the coordination status of each robot allowing robots to detect uncoordinated behavior (e.g., several robots with the same exclusive role) and to correct this situation reinforcing the reliability of coordination algorithms.

The communication between the base station and the robots informs the team of the active play mode (decided by the referee). During development, the base station can be used to control several robotic agent characteristics like fixed roles, manually activated self-positioning, etc, all managed through the RTDB.

4. Positionings and roles in open play

For open play, CAMBADA uses an implicit coordination model based on notions like strategic positioning, role and formation. These notions and related algorithms have been introduced and/or extensively explored in the RoboCup Soccer Simulation League [16,17].

The concept of formation adopted in CAMBADA is mostly the same as the one presented in [16]. The model that was used to define the strategic positions of the formation members for each situation is derived from [17]. However, these and other methods developed in the Simulation League assume that team size is constant. In the MSL we must deal with incomplete formations, resulting from referee orders or malfunctioning robots.

4.1. Formations and strategic positionings

A formation defines a movement model for the robotic players. Formations are sets of strategic positionings, where each positioning is a movement model for a specific player. The assignment of players to specific positionings is dynamic, and it is done according to some rules described below. Each positioning is specified by three elements:

Home position which is the target position of the player when the ball is at the center of the field.

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

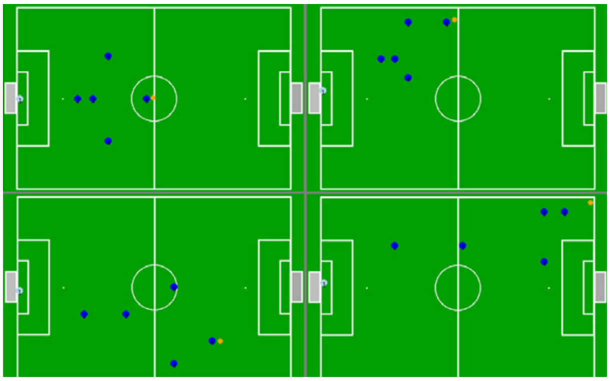

Fig. 3. Target player positions for several different ball positions.

Region of the field where the player can move.
Ball attraction parameters used to compute the target position of the player in each moment based on the current ball position.

All these items of information are given in a strategy configuration file. Using different home positions and attraction parameters for the positionings allows a simple definition of defensive, wing, midfielder and attack strategic movement models. Fig. 3 shows the formation of the team used in RoboCup'2008 for several ball positions.

The definition of a formation in terms of strategic positionings was introduced in the SBSP model [17] for the Soccer Simulation League. This model also introduced specific notions of tactic and strategy, which are currently not used in CAMBADA.

4.2. Roles in open play

Each role has an associated hierarchical finite-state machine that decides the behavior to be used for each of its states based on the current world state [24]. Using an hierarchical finite state machine enables the modeling of transitions between states with different levels of importance. Certain super-states (states that include other states) may be exited by the activation of one of these high priority conditions, without having to consider an identical transition from each of the their substates.

As mentioned before, the CAMBADA players use only three roles in play-on mode: `RoleGoalie`, activated for the goalkeeper, `RoleSupporter` and `RoleStriker`. `RoleStriker` is an "active player" role. It tries to catch the ball and score goals. The striker activates several behaviors that try to engage the ball (bMove, bMoveToAbs), get into the opponent's side avoiding obstacles (bDribble) and shoot to the goal (bKick). The bDribble behavior can perform 180 degrees turns while keeping possession of the ball.

In a consistent role assignment, only one player at a time takes on the role of striker. The striker is helped by other teammates which take on `RoleSupporter` [24]. Supporters maintain their target positions as determined by their current positioning assignments and the current ball position. To this end, they use essentially the bMoveToAbs behavior. As a result, supporters accompany the striker as it plays along the field, without interfering. In case the ball is captured by the opponent, some supporter hopefully will be in a good position to become the new striker. Occasionally, supporters can take a more active behavior. This happens when the striker cannot progress with the ball towards the opponent goal and, instead, the ball remains behind the striker for more than some pre-defined time (2 s in the adopted configuration). In this case, the closest supporter to the ball also approaches the ball, acting as "backup striker".

4.3. Role and positioning assignment

Previous work on role assignment algorithms for robotic soccer is based on the concept of role exchange, measuring the utility of that exchange to decide its activation [37,16]. However, in MSL the number of available players varies as a result of several common situations, namely hardware and software malfunctions and referee orders. As the number of robots is small and varies a lot, the usefulness of role exchanges is reduced. The algorithms used in CAMBADA for role and positioning assignment are based on considering different priorities for the different roles and positionings, so that the most important ones are always covered [34].

In CAMBADA, the algorithms for role assignment and positioning assignment are separated and run at different rates. Role assignment is decided locally by each robot, every cycle (40 ms), based on its current world model. The positioning assignment is decided by the coach and communicated to the agents, through the RTDB, every second. We believe this is an improvement over previous approaches [37,16], in which role and positioning assignment were integrated. The adopted separation provides a very reactive role assignment to cope with the high dynamics of MSL games, and a more stable and consistent positioning assignment. In case the coach fails, robots are prepared to run the positioning assignment algorithm locally.

During open play, from the robots that see the ball, the one that estimates having the closest distance to the ball takes on `RoleStriker`, and all others, except the goalie, take on `RoleSupporter`. Because world models are not identical, in some situations more than one robot may be assigned `RoleStriker`, but the results provided in Section 6 show that this situation is very rare.

The positioning assignment algorithm decides the place in the formation that each robot should occupy (see Fig. 4). Consider a formation with N positionings and a team of $K \leqslant N$ available field players (not counting the goalkeeper which has a fixed role). To assign the positioning to each robot, the distances of each of the robots to each of the target positions are calculated.

Then the closest robot to the highest priority strategic positioning is assigned to that positioning, which is in turn the closest to the ball. From the remaining $K-1$ robots, the closest to the defensive positioning (second highest priority) is assigned to this positioning, then the closest to the third level priority positioning

```
Algorithm: positioning assignment
Input:
  POS - array of N positionings
  BallPos - ball position
Input/output:
  PL - array of K active players (K =< N)
Local:
  TP - array of N target positions
{
  clearAssignments(PL);
  TP = calcTargetPositions(POS,BallPos);
  for each POS[i], i = 1..N, in
            descending order of priority
  {
    if there is no free player
      then return;
    p = the free player closest to TP[i];
    PL[p].positioning = i;
    PL[p].targetPosition = TP[i];
  }
}
```

Fig. 4. CAMBADA positioning assignment algorithm.

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

is assigned next and the algorithm continues until all active robots have positionings assigned. The robot assigned to the highest priority positioning will in most cases be locally assigned to `RoleStriker` and will not move to that positioning, but will position itself close to the ball assuring the stability of the assignment. This algorithm results in the `RoleStriker` having top priority, followed by the defensive positioning, followed by the other supporter positionings.

5. Coordinated procedures

Coordinated procedures are short plans executed by at least two robots. These plans in some cases involve communication resulting in explicit coordination. In the case of CAMBADA coordinated procedures are used for passes and set plays.

5.1. Passes

Passing is a coordinated behavior involving two players, in which one kicks the ball towards the other, so that the other can continue with the ball. Until now, MSL teams have shown limited success in implementing and demonstrating passes. In RoboCup'2004, some teams had already implemented passes, but the functionality was not robust enough to actually be useful in games [13,48]. The CoPS and Tribots team also support pass play [49,40].

Two player roles have recently been developed for coordinated passes in the CAMBADA team. In the general case, the player running `RoleStriker` may decide to take on `RolePasser`, choosing the player to receive the ball. After being notified, the second player takes on the `RoleReceiver`.

These roles have not been used yet for open play in international competition games, but they have been demonstrated in RoboCup'2008 MSL Free Technical Challenge and a similar mechanism has been used for corner kicks (see below). In the free challenge, two robots alternately took on the roles of passer and receiver until one of them was in a position to score a goal (Fig. 5).

The sequence of actions on both players is described in Table 1. They start from their own side of the field and, after each pass, the passer moves forward in the field, then becoming the receiver of the next pass. The coordination between passer and receiver is based on passing flags, one for each player, which can take the following values: READY, TRYING_TO_PASS and BALL_PASSED. In the case of a normal game, another pass coordination variable would identify the receiver.

5.2. Set plays

Another methodology implemented in CAMBADA is the use of coordinated procedures for set plays, i.e. situations when the ball is introduced in open play after a stoppage, such as kick-off, throw-in, corner kick, free kick and goal kick. Set play procedures define a sequence of behaviors for several robots in a coordinated way. For that purpose, the involved players take on specific roles. This role-based implementation of set plays not only was easy to integrate within the previous agent architecture, but also facilitated the test and tune of different possibilities allowing for very efficient final implementations.

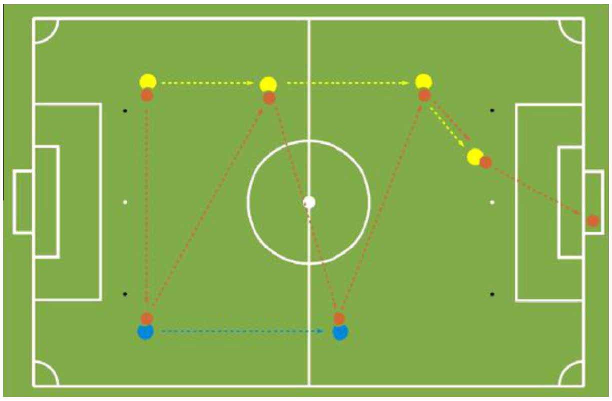

Fig. 5. Sequence of passes demonstrated in the free challenge of RoboCup'2008.

Table 1
Coordinated actions in a pass.

`RolePasser`	`RoleReceiver`
PassFlag TRYING_TO_PASS	
Align to receiver	Align to Passer
	PassFlag READY
Kick the ball	
PassFlag BALL_PASSED	
Move to next position	Catch ball

`RoleToucher` and `RoleReplacer` are used to overcome the 2008 MSL indirect rule in the case of indirect set pieces against the opponent [27]. The purpose of `RoleToucher` is to touch the ball and leave it to the `RoleReplacer` player. The replacer handles the ball only after it has been touched by the toucher. This scheme allows the replacer to score a direct goal if the opportunity arises.

Two toucher–replacer procedures are implemented. In the case of corner kicks, the toucher passes the ball to the replacer and the replacer continues with the ball (see pseudo-code in Fig. 6). The passing algorithm is as explained above.

Another toucher–replacer procedure is used in the case of throw-in, goal kick and free kick set plays. Here, the toucher approaches and touches the ball pushing it towards the replacer until the ball is engaged by the replacer, then withdraws leaving the ball to the replacer. The replacer also moves towards the ball, grabs it, waits that the toucher moves away and then shoots to the opponent goal. It should be noted that both the toucher and the replacer position themselves on the shoot line, so that, as soon as the toucher moves away, the replacer is ready to shoot. For the kick-off, a similar procedure is followed, but without reference to the shoot line, since the involved robots must be in their own side of the field.

This scheme has been updated in 2009 to comply with the new rule that only allows one robot of the team performing the set piece (and none from the opponent team) within the 1 m circle around the ball and obliges the ball to be immediately kicked and to roll free on the field for at least 0.5 m. In 2009, the `RoleReplacer` passes the ball to one of, possibly multiple, robots acting as `RoleReceiver`. Before passing, an evaluation of the passing corridors is performed jointly by the Replacer and all Receivers and results are shared through the RTDB. It is the responsibility of the Replacer to choose the destination of the pass, which is also communicated through the RTDB before pass execution.

Finally, in the case of set pieces against CAMBADA, `RoleBarrier` is used to protect the goal from a direct shoot. The line connecting

```
Algorithm: RoleReplacer // for corner kicks
{
if I have Ball then shoot to opponent goal
  else if Ball close to me
    then move to Ball
    else if Toucher already passed ball
           then catch Ball
           else wait that Ball is passed
}
```

Fig. 6. Replacer role algorithm for corner kicks.

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

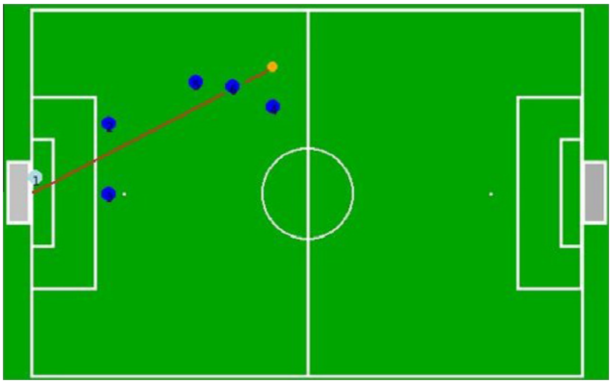

Fig. 7. Placement of `RoleBarrier` players.

the ball to the own goal defines the barrier positions. One player places itself on this line, as close to the ball as it is allowed. Two players place themselves near the penalty area. One player is placed near the ball, 45 degrees from the mentioned line, so that it can observe the ball coming into play and report that to teammates. Finally, one player positions itself in such a way that it can oppose to the progression of the ball through the closest side of the field. The placement of players is illustrated in Fig. 7.

The assignment of the `RoleBarrier`, `RoleReceiver`, `RoleReplacer` and `RoleToucher` roles is executed by sorting the agents according to their perceived distances to the ball and selecting the closest ones, up to the maximum number of agents in each role. When selecting a role like the `RoleReplacer`, which is exclusive, the agent looks at the other teammates role decisions and if it finds a `RoleReplacer` with a lower uniform number it will never select that role. A similar approach is performed for the other exclusive roles. This assignment is always performed locally by each robot. Robots that are not assigned setplay specific roles are assigned the supporter role with a positioning that does not interfere with the setplay.

As soon as the setplay finishes, either because of a timeout or because all the setplay actions have been performed with success, the robots assigned with specific setplay roles return to an open play role using the role assignment algorithm previously described.

6. Performance evaluation

The CAMBADA team participated and won the MSL world championship in RoboCup'2008 (Suzhou, China, July 2008) and achieved a distinct 3rd place in RoboCup'2009 (Graz, Austria, July 2009). Most performance evaluation measures presented in this Section were obtained by analyzing log files and videos of games in the RoboCup championships. The logs are created by the coach agent. At 1 s intervals, the coach takes a snapshot of relevant information retrieved from each robot, including current role, strategic positioning, behavior, self position and ball position. A software tool was developed to analyze game logs and extract relevant evaluation measures. Most of the information presented below was extracted from the RoboCup'2008 logs. As the CAMBADA team made it to the final, it was scheduled to play 13 games. One of them was not played due to absence of the opponent. For two other games, the log files were lost. Thus, the results presented below are extracted from log files of the remaining 10 games. Some additional results were extracted from the semi-final game in RoboCup'2009. Finally, RoboCup'2008 and RoboCup'2009 competition results will also be presented.

Table 2
Time distribution for different classes of game states.

Game state	% Time
Open play	53.1
Set piece for	21.5
Set piece against	25.4

6.1. General game features

Three main classes of game states are open play, set piece against CAMBADA and set piece for CAMBADA. Table 2 shows the respective time distribution in percentage of full game duration, computed over the 10 game logs mentioned above. The time spent in set pieces, considerably higher than what might be expected, results from the dynamics in MSL games. In fact, robots fast moving capabilities (up to 4 m/s) and powerful ball kicking capabilities are not accompanied by sufficiently effective ball control capabilities, thus causing various types of set pieces. The time spent in set pieces justifies the investment in the development of the replacer/toucher combination in CAMBADA. A high efficiency rate in set pieces makes a real difference in the final team performance.

Another common feature in MSL teams is that, due to reliability issues, the number of playing field robots is often less than the maximum of five. Table 3 shows the average percentage of game time (in the 10 mentioned game logs) for different numbers of playing field robots in the CAMBADA team.

The average number of running field robots for the CAMBADA team was 3.98. This reveals the reliability problems that were experienced mostly in the beginning of the championship. These were solved to some extent during the championship and reliability improved in later games. In the final game the average number of running field robots was 4.33.

Capabilities for shooting to goal, although not directly based on coordination methodologies, are essential for a team's success. Fig. 8 shows the location from where the ball was shot to goal in the RoboCup'2008 MSL final (CAMBADA-TechUnited). CAMBADA showed good scoring abilities in the competition. Table 4 shows the results of all the shots made in the final game within 9 m of

Table 3
Percentage of game time for different numbers of playing field robots.

Number of robots	0	1	2	3	4	5
Time (%)	0.3	4.5	3.5	16.1	39.3	36.3

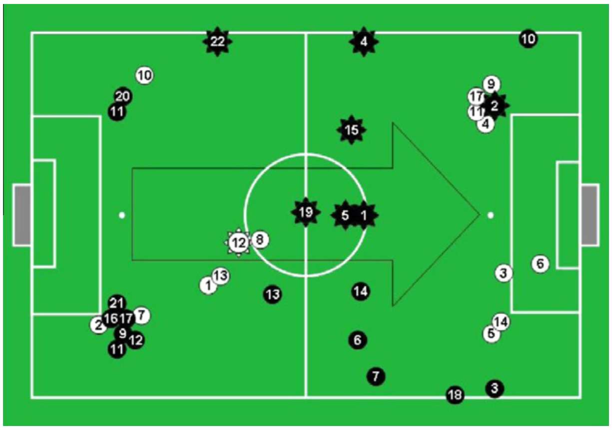

Fig. 8. Shoot locations in the final CAMBADA (black, on the left)–Tech United (white, on the right) game in RoboCup'2008 (shoots are circles and goals are sun-like forms).

Table 4
Goal scoring performance.

Result	Number
Missed	1
Post/bar	2
Defended	5
Goal	7
Total	15

Table 5
Average time spent by players in different roles (in %) and respective standard deviation.

Role	% Time
`RoleStriker`	10.4 ± 5.2
`RoleSupporter`	45.2 ± 10.0
`RoleToucher`	5.9 ± 4.1
`RoleReplacer`	5.6 ± 4.6
`RoleBarrier`	28.4 ± 6.5
`RoleParking`	4.4 ± 6.4

the opponent goal (for larger distances, a shot does not have enough power to pose a real threat to the opponent team). A total of 15 shots were made, of which 1 was missed, 1 hit the post and another hit the bar. The remaining 12 hit the intended target within the goal. This gives an accuracy rating of 80%. From all the 15 shots made, 7 resulted in a goal being scored. This gives a goal scoring success rate (within 9 m) of 46.7%.

This high success rate is the result of accurate ball placing when kicking. In 5 of the 7 scored goals, the goalkeeper was actually well positioned and in the path of the ball. However, the accurate calibration and power selection for each kick made the ball reach the opponent goal at an height slightly above 80 cm which effectively caused it to go over the goalkeeper, and thus creating a shot that is very difficult to defend.

6.2. Roles and behaviors

Table 5 shows the average percentage of time any given player spends in each role, with respect to the total time the player is active in each game. It can be observed that players spend a considerable amount of time (45.2%) as `RoleSupporter`. This is to be expected since there may be up to four players with the Supporter role in open play, while there is at most one player acting as `RoleStriker`. This largely explains why the `RoleStriker` time is approximately 1/4 of the `RoleSupporter` time. The small deviation from the exact 1/4 relation is explained by two main factors: first, `RoleSupporter` is also taken by some players during set plays for CAMBADA; and, second, the number of field robots is often less than the maximum of five, as described above.

It can also be seen that more time is spent in set plays against CAMBADA (28.4%, since usually four players take the Barrier role in these situations) than in set plays against the opponent (11.5% in Toucher and Replacer roles). `RoleParking` moves robots outside of the field at the end of the first half and at the end of the game.

Table 6
Average time spent by players in different roles (in %) for different game states.

Role	Open play	Set piece for	Set piece against
`RoleStriker`	24.3	0.3	0.4
`RoleSupporter`	75.3	51.5	0.3
`RoleToucher`	0.4	23.7	0.0
`RoleReplacer`	0.0	24.5	0.0
`RoleBarrier`	0.0	0.0	99.3

A more in-depth perspective is given by Table 6, which shows the role time distribution across the three classes of game states. It can be seen that in open play basically only `RoleStriker` and `RoleSupporter` are used. In set pieces for CAMBADA, players take the roles of `RoleReplacer`, `RoleToucher` and `RoleSupporter`. In set pieces against CAMBADA, all field robots act is `RoleBarrier`. Underlying the numbers in Table 6 is the fact, already mentioned above, that CAMBADA had an average of nearly four field players. That explains why the time spent as supporter in open play is approximately three times that of striker, and the time spent as supporter in set pieces for CAMBADA is approximately two times that of toucher or replacer.

Table 7 shows the average percentage of time any given player spends running each implemented behavior. The second column of the table shows such percentages irrespective of the role taken. The third column shows the percentages of time considering only the periods in which players are acting as `RoleStriker`.

These values highlight the specificity of `RoleStriker`: much less time moving to absolute positions, since the striker most of the time ignores its strategic positioning assignment; much more time in moving (to the ball), dribbling and kicking.

6.3. Coordination

In the final game of RoboCup'2008 (CAMBADA-TechUnited), the ball was in the opponent's side 73% of time, mainly in the center of the field towards the opponent's side. While this certainly results from the combination of several factors, the CAMBADA's coordination approach has certainly helped in achieving such high field dominance.

Some measures of coordination performance have been extracted. According to the logs, players change roles 2.02 ± 1.02 times per minute. As role assignment is distributed (implicit coordination), it occasionally happens that two players take on `RoleStriker` at the same time. On average, all inconsistencies in the assignment of the Striker role have a combined total duration of 20.9 ± 27.4 s in a game (~30 min), i.e., the mean inconsistency time is about 1.2% of game duration. The high standard deviation results mainly from one game in which, due to magnetic interference, localization errors were higher than normal. In that game, role inconsistencies occurred 45 times for a combined total of 101 s.

Concerning strategic positionings, relevant mainly to supporters, the average distance of the player to its target position is 1.38 ± 0.48 m. The strategic positioning assignment for each player is changed on average 9.83 ± 2.23 times per minute.

As the CAMBADA players do not track the positions and actions of the opponent players, it is not possible to compute an exact

Table 7
Average time (±standard deviation) spent by players running different behaviors.

Behavior	% Time (any role)	% Time (striker)
bMove	4.9 ± 3.0	43.7 ± 4.4
bMoveToAbs	74.7 ± 12.6	25.3 ± 4.7
bDribble	1.4 ± 1.2	13.4 ± 4.5
bKick	1.8 ± 1.5	14.6 ± 7.7
bCatchBall	0.2 ± 0.3	

Table 8
Measures related to ball possession (average ± standard deviation).

Average minimum distance to the ball (m)	1.25 ± 0.33
Time with ball visible (%)	91.7 ± 3.5
Time with ball engaged (%)	9.8 ± 4.7

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

Table 9
Set-piece performance in the RoboCup'2008 final game.

Set piece	# Occurrences	# Correct
Kick-off	2	2
Free kick	1	1
Throw in	6	5
Goal kick	10	8
Corner kick	2	2
Total	21	18

measure of ball possession. However, the game logs enable to compute related measures, as shown in Table 8. The closest player to the ball is at an average distance of 1.2 m from the ball (the field is 18 m × 12 m). The ball is perceived by at least one robot of the CAMBADA team 91.7% of the time. The ball is engaged in a robots grabber device 9.8% of the time.

Some additional analysis was carried out based on the logs of the RoboCup'2008 final game. Table 9 provides information on set pieces, identifying the total number of times each set piece was executed as well as the number of times it was correctly executed.

In RoboCup'2008 final game there were 21 set pieces, of which 18 were correctly executed (85.7%). The failed throw-in occurred due to magnetic interference in one area of the field, causing the robot to mislocalize itself. The two missed goal kicks occurred because the movement of the robot acting as `RoleToucher`, while pushing the ball towards the Replacer, was not accurately aligned and did not succeed in delivering the ball to the Replacer. This can be due to some small localization errors experienced near the goal kick marker.

Table 10 provides information on goal scoring success in set piece situations in which the set piece procedure was correctly executed and the distance to the opponent goal was less than 9 m. In the six set pieces for CAMBADA, carried out under these conditions, four resulted in a goal being scored. This is a very good success rate. It should be noted that from the seven goals scored in this game, four resulted from set pieces. This shows the importance of having accurate, reliable and swift set pieces in MSL games. These high values were observed consistently throughout the whole championship. They were crucial in the teams success, proving to be a powerful asset for achieving victories.

An identical analysis was performed based on the logs of the RoboCup'2009 semi-final game, in which CAMBADA played against the same opponent of the 2008 final: Tech United. Table 11 shows the obtained results.

Table 10
Goal scoring performance in set piece situations.

Set piece	# Occurrences	# Success
Kick-off	2	2
Free kick	1	0
Throw in	3	2
Total	6	4

Table 11
Set-piece performance in the RoboCup'2009 semi-final game.

Set piece	# Occurrences	# Correct
Kick-off	3	3
Free kick	6	4
Throw in	13	11
Goal kick	2	1
Corner kick	1	1
Total	25	20

Table 12
Outcomes of set pieces correctly executed in the RoboCup'2009 semi-final game.

Outcome	# Occurrences	%
Receiver blocked by opponent	12	60
Ball off the field through goal line	3	15
Ball hits goal framework	2	10
Goalkeeper defense	2	10
Receiver dribbling with ball	1	5
Total	20	100

Table 13
RoboCup'2008 competition results.

	# Games	# Goals scored	# Goals suffered	# Points
Round-robin 1	5	41	2	15
Round-robin 2	4	16	3	9
Round-robin 3	2	5	2	3
Semi-final	1	4	3	3
Final	1	7	1	3
Total	13	73	11	33

Table 14
RoboCup'2009 competition results.

	# Games	# Goals scored	# Goals suffered	# Points
Round-robin 1	6	38	6	15
Round-robin 2	4	23	2	12
Round-robin 3	2	7	2	6
Semi-final	1	0	2	0
3rd Place	1	3	1	3
Total	14	71	13	36

In RoboCup'2009 the number of set pieces in the semi-final (against Tech United) was 25, of which 20 were correctly executed (80%). It should be noticed that 2009 rules make it much more difficult to control the ball during the execution of the coordinated procedure following a set piece. While in 2008 the ball moved very little during the execution of the coordinated procedure (robots moved to touch the ball and then the replacer shoots to goal), the 2009 MSL rules make it obligatory for the ball to roll free for at least 0.5 m. This gives more time for the opponent team to react and forces the interception of a moving ball, a capability that is still not perfectly performed with the current robots. The outcomes of the 20 set pieces that were correctly executed in that semi-final can be observed in Table 12.

6.4. Competition results

Tables 13 and 14 present the competition results of CAMBADA in RoboCup'2008 and RoboCup'2009. In 2008, the team won 11 out of 13 games, scoring a total of 73 goals and suffering only 11 goals. The participation in 2009 also resulted in the team winning 12 of the 14 played games, scoring a total of 71 goals, far more than any other team, and suffering 13 goals.

7. Conclusion

This paper presented and evaluated the coordination methodologies of CAMBADA, one of the top teams in RoboCup MSL world championships (champion in RoboCup'2008, 3rd place winner in RoboCup'2009).

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

During open play, an implicit coordination approach, based on formations, flexible positionings and dynamic role and positioning assignment, is used. The positioning of the team adapts to the external game conditions and maintains a strong defense and a good backup to the striker role. This is achieved through priority-based positioning/role assignment algorithms that maintain a competitive formation even when robot malfunctions decrease the number of field players. The positioning assignment algorithm is focused on covering the most important roles/positionings and differs substantially from previously presented algorithms that were based on role exchange. The success of the approach can be seen, not only from the competition results, but also from the detailed analysis of game logs and videos, as presented in the paper. More importantly, and this is one of the clearest evidences, the good competition results were obtained despite the fact that CAMBADA robots clearly move at low speed (2 m/s), when compared to most of the main competitors which move faster (2.5–4 m/s).

The development of pre-defined role-based set plays proved to be very efficient both during the development phase, and during their execution in games. More than half of the 73 scored goals are direct result of these set plays.

One of the most significant aspects of this work is the integration of the described coordination methodologies in a complex multi-robot system and their validation in the challenging RoboCup MSL competition scenario. This contrasts with many other approaches described in the literature, which are often validated in more controlled robotic environments, if not in simulation.

Acknowledgments

The CAMBADA team was funded by the Portuguese Government – FCT-POSI/ROBO/43908/2002 (CAMBADA) and currently FCT, PTDC/EIA/70695/2006 (ACORD). We would also like to thank the rest of the CAMBADA team for an excellent work environment.

References

[1] Noreils F. Toward a robot architecture integrating cooperation between mobile robots: application to indoor environment. Int J Robotics Res 1993(12):79–98.
[2] Wang P. Navigation strategies for multiple autonomous mobile robots moving in formation. J Robotic Syst 1991;8(2):177–95.
[3] Balch T, Parker L. Robot teams: from diversity to polymorphism. Natick (Massachusetts): A K Peters Ltd.; 2002.
[4] Low KH, Gordon GJ, et al. Adaptive sampling for multi-robot wide-area exploration. In: Proc IEEE int conf on robotics and automation, Rome, Italy; 2007. p. 755–60.
[5] Yamauchi B. Frontier-based exploration using multiple robots. In: Proc second int conf on autonomous agents; 1998. p. 47–53.
[6] Burgard W, Moors M, Stachniss C, Schneider F. Coordinated multi-robot exploration. IEEE Trans Robotics Autom 2005;21(3):376–86.
[7] Carpin S. Fast and accurate map merging for multi-robot systems. Auton Robots 2008;25(3):305–16.
[8] Dahl T, Mataric M, Sukhatme G. Emergent robot differentiation for distributed multi-robot task allocation. In: Proc 7th international symposium on distributed autonomous robotic systems, Toulouse, France; 2004. p. 201–10.
[9] Figueiredo L, Jesus I, Machado J, Ferreira J, de Carvalho JM. Towards the development of intelligent transportation systems. In: Proc IEEE intellig transport syst; 2001. p. 1206–11.
[10] Tewolde G, Wu G, et al. Distributed multi-robot work load partition in manufacturing automation. In: Proc IEEE int conf on autom science and eng, Arlington, VA, USA, 2008. p. 504–9.
[11] Fagiolini A, Pellinacci M, et al. Consensus-based distributed intrusion detection for multi-robot systems. In: Proc IEEE int conf robotics and automation, Pasadena, CA, USA; 2008. p. 120–7.
[12] Agmon N, Kraus S, Kaminka G. Multi-robot perimeter patrol in adversarial settings. In: Proc IEEE int conf on robotics and automation, Pasadena CA, USA; 2008. p. 2339–45.
[13] Lima P, Custdio L, Akin I, Jacoff A, Kraezschmar G, Ng BK, et al. RoboCup 2004 competitions and symposium: a small kick for robots a giant score for science. AI Mag 2005;6(2):36–61.
[14] Gerkey BP, Mataric M. A formal analysis and taxonomy of task allocation in multi-robot systems. Int J Robotics Res 2004;23(9):939–54.
[15] Michael N, Zavlanos M, et al. Distributed multi-robot task assignment and formation control. In: Proc IEEE int conf on robotics and automation, Pasadena CA; 2008, pp. 128–33.
[16] Stone P, Veloso M. Task decomposition, dynamic role assignment and low bandwidth communication for real time strategic teamwork. Artif Intell 1999;110(2):241–73.
[17] Reis L, Lau N, Oliveira E. Situation based strategic positioning for coordinating a team of homogeneous agents. In: Hannenbauer M et al., editors. Balancing reactivity and social deliberation in multiagent sytems: from RoboCup to real word applications. LNAI, vol. 2103. Springer-Verlag; 2001. p. 175–97.
[18] Pagello E, DGAngelo A, Menegatti E. Cooperation issues and distributed sensing for multirobot systems. Proc IEEE 2006;94(7):1370–83.
[19] Balch T, Arkin R. Behavior-based formation control for multirobot teams. IEEE Trans Robotics Autom 1998;14(6):926–39.
[20] Cheah C, Hou S, Slotine J. Region following formation control for multi-robot systems. In: Proc IEEE int conf on robotics and automation, Pasadena CA, USA; 2008. p. 3796–801.
[21] Joyeux S, Alami R, Lacroix S. A plan manager for multi-robot systems. In: Proc 6th int conf on field and service robotics, Chamonix, France; 2007. p. 443–52.
[22] Lesser V, Decker K, Wagner T, Carver N, Garvey A, Horling B, et al. Evolution of the GPGP/TAEMS domain-independent coordination framework. In: Autonomous agents and multi-agent systems, vol. 9(1); 2004. p. 87–143.
[23] Dietl M, Gutmann J-S, Nebel B. Cooperative sensing in dynamic environments. In: Proc IEEE/RSJ int conf on intelligent robots and systems (IROS'01), Maui, Hawaii; 2001.
[24] Lau N, Seabra Lopes L, Corrente G. Cambada: information sharing and team coordination. In: Autonomous robot systems and competitions: proc of the 8th conference. Aveiro (Portugal): Universidade de Aveiro; 2008. p. 27–32.
[25] Figueiredo J, Lau N, Pereira A. Multi-agent debugging and monitoring framework. In: Proc first IFAC workshop on multivehicle systems (MVS'06), Brazil; 2006.
[26] Rosa MD, Campbell J, et al. Distributed watchpoints: debugging large multi-robot systems. In: Proc IEEE int conf on robotics and automation, Rome, Italy; 2007. p. 3723–9.
[27] M.T.C. 1997–2009. Middle size robot league rules and regulations for 2009. Version – 13.1 20081212 [December 2008].
[28] Azevedo J, Cunha M, Almeida L. Hierarchical distributed architectures for autonomous mobile robots: a case study. In: Proc ETFA2007 – 12th IEEE conference on emerging technologies and factory automation, Patras, Greece; 2007. p. 973–80.
[29] Almeida L, Santos F, Facchinetti T, Pedreira P, Silva V, Seabra Lopes L. Coordinating distributed autonomous agents with a real-time database: the CAMBADA project. In: C A, et al., editors. Computer and information sciences – ISCIS 2004: proc 19th international symposium, LNCS, vol. 3280, Antalya, Turkey; 2004. p. 876–86.
[30] Pedreiras P, Teixeira F, Ferreira N, Almeida L, Pinho A, Santos F. Enhancing the reactivity of the vision subsystem in autonomous mobile robots using real-time techniques. In: Noda I, A J, et al., editors. RoboCup-2005: robot soccer world cup IX, LNAI, vol. 4020. Berlin: Springer; 2006. p. 371–83.
[31] Pedreiras P, Almeida L. Task management for soft real-time applications based on general purpose operating systems. In: Lima P, editor. Robotic soccer. Vienna (Austria): Itech Education and Publishing; 2007. p. 598–607.
[32] Neves A, Corrente G, Pinho A. An omnidirectional vision system for soccer robots. In: Progress in artificial intelligence. LNCS, vol. 4874. Berlin: Springer; 2007. p. 499–507.
[33] Cunha B, Azevedo J, Lau N, Almeida L. Obtaining the inverse distance map from a non-svp hyperbolic catadioptric robotic vision system. In: Visser, U, et al., editors. RoboCup-2007: robot soccer world cup XI, LNAI. Berlin: Springer Verlag; 2008.
[34] Lau N, Seabra Lopes L, Corrente G, Filipe N. Multi-robot team coordination through roles, positionings and coordinated procedures. In: Proc of the 2009 IEEE/RSJ international conference on intelligent robots and systems – IROS 2009, St. Louis, MO, USA; 2009.
[35] Weigel T, Auerbach M, et al. Cs freiburg: doing the right thing in a group. In: P S., et al., editors. RoboCup-2000: robot soccer world cup IV, LNAI, vol. 2019. Springer-Verlag; 2001. p. 52–63.
[36] M A, et al. Creating a robot soccer team from scratch: the brainstormers tribots. In: Proc of RoboCup 2003, Padua, Italy; 2003.
[37] Reis L, Lau N. FC Portugal team description: RoboCup 2000 simulation league champion. In: Stone P, et al., editors. RoboCup-2000: robot soccer world cup IV, LNCS, vol. 2019. Springer; 2001. p. 29–40.
[38] Kok J, Spaan M, Vlassis N. Non-communicative multi-robot coordination in dynamic environments. Robotics Auton Syst 2005;50(2-3):99–114.
[39] Riedmiller M, Gabel T. On experiences in a complex and competitive gaming domain: reinforcement learning meets RoboCup. In: Proc of the 3rd IEEE symposium on computational intelligence and games (CIG 2007). Honolulu (Hawaii):IEEE Press; 2007. p. 17–23.
[40] H M, et al. Making a robot learn to play soccer using reward and punishment. In: KI 2007: Advances in artificial intelligence, LNCS, vol. 4667. Springer; 2007. p. 220–34.
[41] Oubbati M, Schanz M, Buchheim T, Levi P. Velocity control of an omnidirectional RoboCup player with recurrent neural networks. In: A B, et al., editors. RoboCup 2005: robot soccer world cup IX, LNAI, vol. 4020. Springer; 2006. p. 691–701.
[42] Hafner R, Lange S, Lauer M, Riedmiller M. Brainstormers tribots team description; 2008.

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010

[43] Sato Y, et al. Hibikino-musashi team description paper; 2008.
[44] E.T. Group. Ethercat robots win german open, May 2008.
[45] Santos F, Almeida L, Seabra Lopes L, Azevedo JL, Cunha MB. Communicating among robots in the RoboCup middle-size league. In: Baltes J, Lagoudakis MG, Naruse T, Ghidary SS, editors. RoboCup 2009: robot soccer world cup XIII, LNAI, vol. 5949. Berlin/Heidelberg: Springer; 2010. p. 320–31.
[46] Lauer M, Lange S, Riedmiller M. Calculating the perfect match: an efficient and accurate approach for robot self-localisation. In: Bredenfeld A, et al., editors. RoboCup 2005: robot soccer world cup IX, LNAI, vol. 4020. Springer; 2006.
[47] Ferrein A, Hermanns L, Lakemeyer G. Comparing sensor fusion techniques for ball position estimation. In: A B, et al., editors. RoboCup 2005: robot soccer world cup IX, LNAI, vol. 4020. Springer; 2006. 154–65.
[48] van der Vecht B, Lima P. Formulation and implementation of relational behaviours for multi-robot cooperative systems. In: RoboCup 2004: robot soccer world cup VIII, LNAI, vol. 3276. Springer; 2005. p. 516–23.
[49] Zweigle O, Lafrenz R, Buchheim T, Kppeler U-P, Rajaie H, Schreiber F, et al. Cooperative agent behavior based on special interaction nets. Intell Auton Syst 2006:651–9.

Please cite this article in press as: Lau N et al. Robot team coordination using dynamic role and positioning assignment and role based setplays. Mechatronics (2010), doi:10.1016/j.mechatronics.2010.05.010