Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2015

Ana Sofia Nunes Almeida

Jogadas estudadas dinâmicas para a equipa CAMBADA

Dynamic Set Pieces for CAMBADA team

Universidade de Aveiro Departamento de
Electrónica, Telecomunicações e Informática,
2015

Ana Sofia Nunes Almeida

Jogadas estudadas dinâmicas para a equipa CAMBADA

Dynamic Set Pieces for CAMBADA team

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica de Prof. Doutor António Neves e Prof. Doutor Nuno Lau, Professores do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president	**Professor Doutor José Luis Costa Pinto de Azevedo** Professor Auxiliar da Universidade de Aveiro
vogais / examiners committee	**Doutor João Manuel Leite da Silva** Coordenador da área de robótica, Fábrica Centro Ciência Viva Aveiro
	Professor Doutor António José Ribeiro Neves Professor Auxiliar da Universidade de Aveiro (orientador)

Agradecimentos

Aos membros da equipa CAMBADA, pelo tempo que dispensaram comigo, sempre que lhes pedi auxílio, em especial ao Professor António Neves, que além da ajuda prestada me concedeu a oportunidade de participar neste projecto, ao Professor Nuno Lau, Ricardo, Filipe e João.

À minha família (Mãe, Pai, Zé, Carlos, Zira) pelo apoio incondicional e pela presença constante ainda que tantas vezes separados por milhares de quilómetros.

Ao Hermínio, Lurdes e Sónia por me terem acolhido e tratado como mais uma na família ao longo destes últimos anos.

Àqueles que me acompanharam ao longo destes anos por Aveiro, com quem tive o prazer de criar tantas (e boas!) memórias.

Ao Eduardo pelos abraços quando tenho frio, pelos sorrisos quando estou triste, pela calma quando estou nervosa, pelo ralhete quando preciso, por ser quem é.

A todos um muito obrigada.

Resumo

A robótica é um ramo emergente da engenharia que envolve a conceção, fabrico e controlo de robôs. É uma área multidisciplinar que conjuga conhecimentos de mecânica, design, eletrónica e computação, inteligência artificial e nanotecnologia. A sua evolução resulta em máquinas capazes de realizar tarefas com alguma complexidade. Sistemas multi-agente, são um dos temas de pesquisa dentro da robótica, pois permitem a realização de tarefas de maior complexidade, através da execução de rotinas simples.

O futebol robótico permite o estudo e desenvolvimento de robótica e de sistemas multi-agente, uma vez que os agentes tem de trabalhar em equipa, tendo em consideração grande parte dos problemas que encontramos no nosso quotidiano, como por exemplo a adaptação a um ambiente extremamente dinâmico como o de um jogo de futebol. CAMBADA é a equipa de futebol robótico pertencente ao grupo de investigação IRIS, do IEETA, constítuida por docentes, investigadores e alunos da Universidade de Aveiro, que anualmente tem como principal objetivo a participação no RoboCup na Middle Size League.

Este trabalho tem como principal objectivo melhorar a coordenação da equipa em situações de bola parada. Esta tese introduz um novo comportamento e a adaptação dos já existentes para situações ofensivas, assim como propõe um novo método de posicionamento a ser usado em situações defensivas.

O trabalho desenvolvido foi incorporado no software de competição dos robôs, o que permite nesta dissertação apresentar resultados experimentais obtidos através de simulação e de testes efetuados nos robôs em laboratório.

Abstract

Robotics is an emergent branch of engineering that involves the conception, manufacture, and control of robots. It is a multidisciplinary field that combines electronics, design, computer science, artificial intelligence, mechanics and nanotechnology. Its evolution results in machines that are able to perform tasks with some level of complexity. Multi-agent systems is a researching topic within robotics, thus they allow the solving of higher complexity problems, through the execution of simple routines.

Robotic soccer allows the study and development of robotics and multi-agent systems, as the agents have to work together as a team, having in consideration most problems found in our quotidian, as for example adaptation to a highly dynamic environment as it is the one of a soccer game. CAMBADA is the robotic soccer team belonging to the group of research IRIS from IEETA, composed by teachers, researchers and students of the University of Aveiro, which annually has as main objective the participation in the RoboCup, in the Middle Size League.

The purpose of this work is to improve the coordination in set pieces situations. This thesis introduces a new behavior and the adaptation of the already existing ones in the offensive situation, as well as the proposal of a new positioning method in defensive situations.

The developed work was incorporated within the competition software of the robots. Which allows the presentation, in this dissertation, of the experimental results obtained, through simulation software as well as through the physical robots on the laboratory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multi-Agent systems have been conquering field as a research topic, since they can be considered for a broad class of applications from low to high degree of complexity, ranging from environmental monitoring, robotics, search and rescue operations, security systems to the technological industry. And also the possibility of performing tasks with high level of complexity, through the execution of several simpler behaviors [1].

According to Russel and Norvig [2] an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. Surprisingly, there is no universally accepted definition of the term agent or multi-agent. Behind the fact that autonomy is central to its notion, there is no consensus.

Usually, an agent has a set of possible actions which have associated to each of them pre-conditions, that define the possible situation in which they are triggered. In this way not all actions can be performed in all situations. This brings the key problem facing an agent, that is deciding which of its actions it should perform in order to best satisfy its design objectives. Agent architectures are software architectures for decision making systems that are embedded in an environment [4], which can be classify having in consideration some properties such as autonomy, cooperation, mobility, learning, communication, application, function, class or capability.

A Multi-Agent System (MAS) consists in a set of agents (that can be different from one another) that communicate, cooperate and coordinate between them in order to achieve a common goal. Further it is a system where there are constrains, such that agents may not at any given time knows everything about the world that other agents knows (including the internal states of the other agents themselves), thus they have limited perceptive systems and data is decentralize [5].

Some of the main advantages of using Multi-agent systems are highlighted by Matsubara et al [6] and Stone [7]:

- Efficiency of cooperation.
- Adaptation.
- Robustness.
- Real-time.
- Parallelism.
- Simpler programming.

1.1 Multi-agent Systems in robotic soccer

Robotic soccer provides a good research context for subjects such as real-time sensor fusion, reactive behavior, strategy acquisition, learning, real-time planning, multi-agent systems, context recognition, vision, strategic decision-making, motor control, intelligent robot control, and so on. It is a team game where strategy is involved and occurs in a highly dynamic environment.

From the MAS perspective, a soccer game is a good example of problems in the real world, because it presents the following characteristics:

- Robustness: if one agent fails, another agent can be assigned for its task.
- Adaptability: it is required for dynamic change of plans according to the operations of the opposing team.
- Communication and coordination: fundamental in order to the team play can be achieved.

As highlighted before, cooperation and coordination between agents still within a soccer game (as in other multi-agent environments) are one of the topic that requires most attention.

This work was developed with the aim of tackling those topics in order to achieve better coordination in Set Pieces situations. Namely, this thesis introduces a new behavior and the adaptation of the already existing ones in the offensive situation, as well as the proposal of a new positioning method in defensive situations.

1.2 Thesis structure

This thesis is divided in 6 chapters. In the first Chapter, the introduction, an explanation of Multi-agent Systems and the reason why its used in Robotic soccer is made. On Chapter 2 it is shown an overview of the RoboCup competition Leagues, giving special detail to the Soccer league. Chapter 3 has a detailed description of offensive and defensive Set Pieces, taking into account the roles and behaviors used by each one. In Chapter 4 it is introduced a new offensive Set Piece behavior. The Chapter 5 is focused in the explanation of utility maps and configuration tool, its importance to this work, and the changes made in the existing algorithm. Finally in the Chapter 6, the conclusions of the developed work under this dissertation are presented.

Chapter 2

RoboCup Leagues and Cambada team

In this chapter it is done a brief explanation of RoboCup objectives and Leagues. Followed by an overview of some team coordination approaches used on RoboCup Soccer league and a brief resume of the achievements made on the Middle Size League. Ending with an explanation of the agent architecture of CAMBADA software agent.

2.1 Robocup

RoboCup is an international initiative that intends to promote robotics and artificial intelligence research, by providing its participants exciting challenges each year in a motivational environment where beside the competition, provides a place to implement and test new ideas and promotes the discussing between teams so they can share the breakthroughs achieved during that year, resulting in a faster technological growth. Research topics include design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics and sensor-fusion [3]. The first edition took place in 1997 in Nagoya, Japan.

In order to make the initiative even more appealing the organization has set a milestone: it is to have a team of fully autonomous humanoid robot soccer players winning a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup, by the year of 2050 [3]. Even if, by the state of art of technology may sound overly ambitious for today, the existence of a long-term (common) goal, leads to a series of sub-goals accomplishments that can be put to use in "real-life" existing problems. Although in the beginning it was only a soccer league, as new necessities/ideas were emerging, new leagues were added to the competition. RoboCup, nowadays, has several leagues:

- RoboCup Rescue.
- RoboCup@Home.
- RoboCup@Work.
- RoboCup Logistics.
- RoboCup Junior.
- RoboCup Soccer.

This work was developed in the framework of the Middle Size League, a sub-league of RoboCup Soccer which has a total of five major sub-leagues: Simulation (2D and 3D); Small-Size; Middle-Size; Standard Platform and Humanoid (Teen Size, Kid Size and Adult Size), each one tackling a specific problem.

2.1.1 Simulation

The Simulation League focus are artificial intelligence and team strategy, allowing the test of high-level multi-agent research issues, not having the hardware limitations in consideration. It is composed by two sub-leagues:

- 2D Simulation League

  Two teams of eleven autonomous software programs (called agents), play in a two-dimensional virtual soccer stadium represented by a central server called `SoccerServer` (Figure 2.1), this server has all the information about the game. It uses that information to simulate real conditions during plays, adding noise to virtual sensor readings of each agent, as well as imperfections to each agent move (performing basic commands such as dashing, turning or kicking).

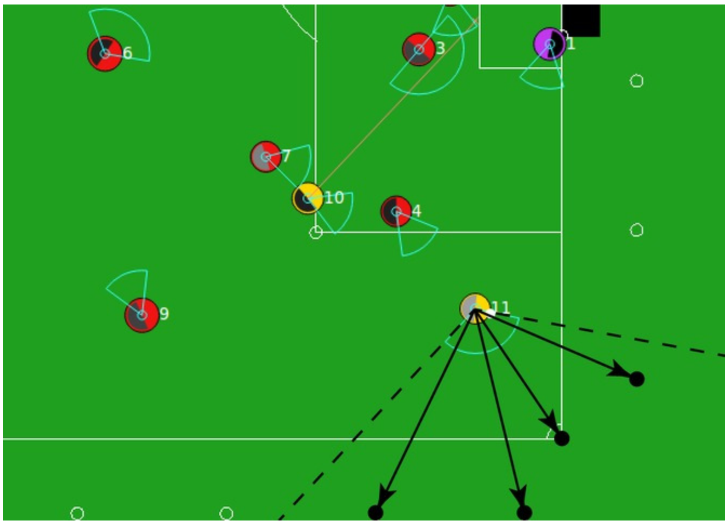


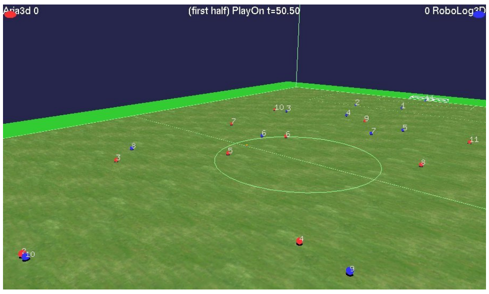
Figure 2.1: RoboCupSoccer, 2D Simulation [9].

- 3D Simulation League

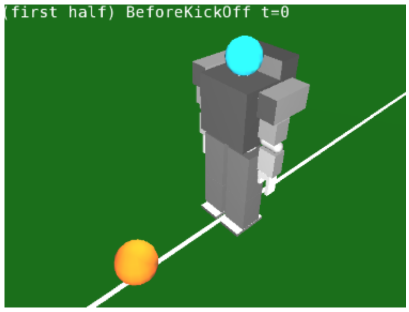
  Introduced in 2004, a totally new 3D simulator [14] adds a new dimension, increasing the realism of the games and the complexity of the physics applied to the agents, allowing the growth of agent actions (closer to the real robots). Players were represented as spheres, Figure 2.2a, only by 2006 a simple humanoid model robot was made available, Figure 2.2b, being the first time that humanoid models were used in the simulation league, Figure 2.2c. Finally in 2008 the models of NAO robots were introduced, Figure 2.2d (the official robot used in Standard Platform League), that brought another perspective to the league, allowing researchers to test their algorithms and ideas before trying them into real robots. By 2012 the number of robots for team reached the eleven elements. In 2013 each team could have robots from different types (variations of the standard NAO robot [8]) and by 2014 happened the first running robot challenge. The goal was to boost other leagues where it is necessary for the hardware robot to be able to run.

2.1.2 Small Size League

Small size league is played by two teams composed by six robots each, controlled by a hybrid centralized/distributed system. Each robot must fit within an 180mm diameter circle



(a) RoboCup, Spheres model



(b) First humanoid model used in simulation

(c) The Soccerbot humanoid soccer simulation used in Atlanta 2007

(d) Nao simulation as used in Hefei 2015

Figure 2.2: 3D Simulation League.

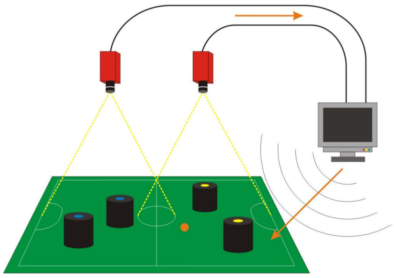
---

and a maximum height of 15cm, Figure 2.3b. The game is played using an orange golf ball on a green carpeted field that is 6.05m long by 4.05m wide. Two cameras attached to a bar 4m above the field, provide data to the vision system. Off-field computers are used to process vision information, sending out commands to the robots and to the referee using wireless communication, as presented in Figure 2.3a.

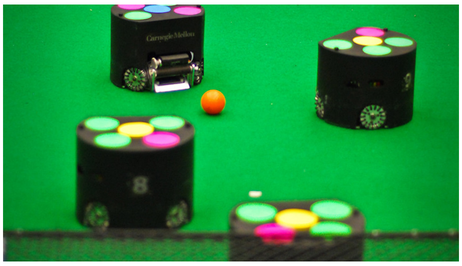
The main focus of this league deals with the problem of intelligent multi-agent cooperation, robot design, control in a highly dynamic environment.

2.1.3 Standard Platform League

Started in the 2008 edition of RoboCup, on Standard Platform League, each team can have a maximum of four players, and it is played in a 9×6m field. All teams must compete with identical robots, NAO [8], as shown in Figure 2.4, completely autonomous (with no external control, neither by humans nor by computers). Using the state of art robots forces



(a) Small size league schematic [12]



(b) CMU Small size league team. [13]

Figure 2.3: Small Size League.

the teams to focus attention on the software development.

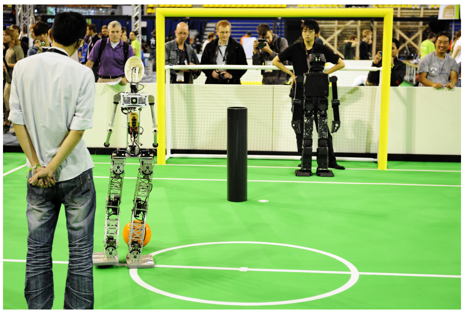
Figure 2.4: Standart Platform Team [9]

2.1.4 Humanoid League

The Humanoid League is played by autonomous, human-like structure and senses robots. Robots are design and built independently by each team. The league is composed by three sub-leagues that differ according to the size of the robot, number of robots on the field and task to perform.

On the Kid Size soccer competition teams can have four robots with a size varying for 40 cm to 90 cm of height, compete in a 6×4m field. In Teen Size soccer competition teams of two robots, with height range from 80cm to 140cm, compete in a 9×6m field, Figure 2.5b. At the Adult Size a striker robot plays against a goal keeper robot (from another team) first, then the same robots play with exchanged roles against each other, robots must be bigger than 130cm and smaller than 180cm, Figure 2.5a.

Major research issues in the humanoid league are dynamic walking, running, kicking the ball while maintaining balance, visual perception of the ball, other players, the field, self-localization and team play. To test this issues individually, along with the soccer competition, technical challenges take place.



(a) Japan team against Tech United (2013 finals) [10]

(b) Team NimbRo TeenSize [11]

Figure 2.5: RoboCup Humanoid League.

2.1.5 Middle Size League

Middle Size League combines most of the research challenges presented in Simulation Leagues, once it requires multi-agent cooperation, team strategy in a highly dynamic environment. Each team can have up to 5 players, with no standard format, that must fit inside a $50 \times 50 \times 80$ cm box and have maximum weight of 40kg, the height above 60cm can have a maximum diameter of 25cm. Robots are predominantly black with a body marker (that can be blue or magenta) identifying the team and the player number, as shown in Figure 2.6 .

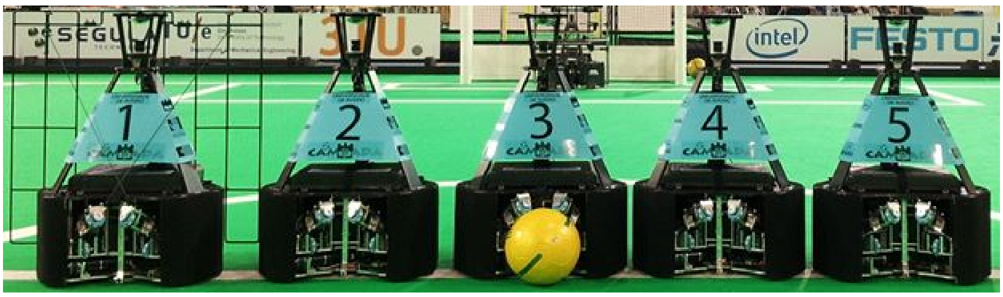


Figure 2.6: Cambada team at RoboCup.

The game is played in a 18×12m field, in two 15 minutes halves, with an official FIFA ball obliging adapted FIFA rules. Players have to be totally autonomous, having on board all sensors and actuators, no human interaction is permitted during the matches, communication between team-mates and coach is allowed via wireless network. The referee decisions are communicated to the team through the referee box, which is an application running on a separate computer, responsible for making the bridge between the human referee and the teams.

Brief history and accomplishments

The technical challenge and the fact that MSL rulebook is updated annually, leads to a driven effort of all teams to overcome a common challenge in the same period of time. This particularity allows, from a scientific and a technical point of view, the identification of well-constrained epochs, during the history of the league [15]:

- First epoch (1997 through 2001):

  Field dimensions were 9×5m, limited by surrounding walls that kept the ball always inside the field, artificial lightning was required to assure small variations, goals were colour coded (yellow and blue) and the ball was bright orange. Each team had up to four robots playing according to basic, FIFA based, rules. Main research issues were basic navigation and vision (colour-based classification and detection of objects). At hardware level the focus were traction solutions and electro-mechanical kickers.

  Some examples of main accomplishments obtained during this epoch are the self-localization methods using laser-range finders by J.S.Gustav et al. [16], the development and usage of omni-directional cameras (Iocchi and Nardi [19], Marques and Lima [18] and A. Bonarini et al. [17]).

- Second epoch (2002 through 2006):

  During this time walls outside the field were removed, instead were added coloured posts on the four corners. The field dimensions were increased to 12×8m, including a penalty and a goal area. A referee box was introduced, allowing a team independent control of the game.

  The research became focus on real-time adaptive colour segmentation, stronger and more precise kicking devices based on pneumatic or solenoid actuation, solutions for catadioptric vision systems (Marques and Lima [20]), efficient omni-directional driving, open loop dribbling devices, early sensor-fusion techniques, self-localization [21] and first solutions for team coordination [22].

- Third epoch (2007):

  Artificial uniform lightening was no longer necessary and field dimensions increase to the current official size, 18×12m. The first rule alteration in order to boost team strategic play was made, a goal could only be validated on the first ten seconds, after a restart, only if the ball was touched by a second team member.

  This changes directed the research to develop and explore concepts like dynamic role-changes and team formations, adjustable kicking systems, world modelling [23], ball-tracking, path planning and distributed real-time databases.

- Fourth epoch (2008 through 2011):

  Coloured corner posts were removed and goals became white. Teams had to apply self-localization methods without external visual aid. The ball had no longer to be orange. Bandwidth use by each team became limited. In restart situations was imposed a minimum distance to the ball, for both own team and opponent team robots, the distance changed being settled at 2m to own team and 3m to opponent team members. Goals could only be validated when the shot was taken within the opponent side of the field, and in situations of direct dispute of the ball only one player of each team could be in direct contact with the ball. The introduction of those new rules made teams to address high-level problems, and focus more aggressively in multi-agent coordination.

  Resulting in improvements on real-time communication, world modelling and role assignment.

- Fifth epoch(2012 until today):

  With the improvements achieved until 2011 teams were able to perform fast dribbles in a controlled way, resulting in some teams research leaning towards the speed instead of a more cooperative solution. To revert that trend a new rule was introduced, robots could not dribble the ball over the mid-line when progressing from their side to the opponent side, they were forced to pass the ball to a team mate on the other side of the field. That rule eventually was changed to a more reasonable one where to score a valid goal the ball has to be received or touched by a team mate within the opponent side of the field after rolling freely for at least one meter. Furthermore continuous dribble was limited to a limit of three meters from the point the robot receives the ball. This changes resulted in a reduction of game speed, pushing offences and favoured the appearing of new strategies such as man-to-man cover, zone-cover or mixes of both. The use of utility maps became more effective, as well as active ball interception.

2.2 Positional coordination approaches adopted in RoboCup

In robotic soccer as in all team sports a good (strategical) coordinate positioning of the players is crucial in order to reach a positive outcome. Allowing the team to evolve faster (in order to score points), or to defend better. As result, positional strategies are an important research focus in the RoboCup soccer league. In this section it will be described some of the developed approaches adopted on the league [29].

Introduced by Stone [24] Strategic Positioning by Attraction and Repulsion (SPAR) is a method that allows players to achieve coordinated positioning through attraction or repulsion to some game elements. When positioning itself using SPAR the agent has to evaluate several forces: repulsion from opponents and team-mates, attraction to the active team-mate, ball and opponents goal, as well as having in consideration being inside the field, stay near its home position, avoid being offside, and be in a position where it is possible to receive a pass. Figure 2.7 represents four possible areas for a robot to move, computed using SPAR. Then the agent moves to the one closest to its base position. This condition ensures that the player with the ball will have multiple passing options distributed around the field.

The Situation Based Strategic Positioning (SBSP) was introduced by Lau, Reis and Oliveira in [25]. Using this method, an agent is able to define its base strategic positioning, through the analysis of the tactic, formation, self positioning in the formation and player type. Each player type has defined strategic characteristics like ball attraction, admissible regions in the field, specific positional characteristics for some zones in the field, tendency to stay behind the ball, alignment in the offside line, and attraction by specific points in the field in particular situations. The resulting position is then adjusted accordingly with the game situation. This methods works in a best distribution of players, when compared with SPAR.

In Figure 2.8 is shows an example of a team strategy, composed by 6 different tactics,

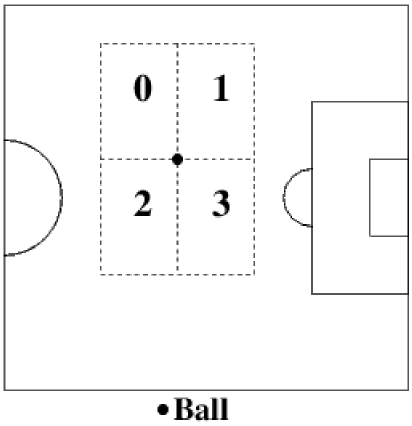


Figure 2.7: Four possible rectangles considered positions calculated using SPAR [24].

each including diverse formations to be used in distinct game situations. To each formation different agent types have assigned different positions.

Later was proposed the Dynamic Positioning based on Voronoi Cells (DPVC) [26], where players are positioned along the field, based on attraction vectors that represent players' attraction towards objects, depending on the match's current situation and players roles. The first step is to compute each agent Voronoi Cell, followed by the calculation of the centre of each cell. Then, it is created a vector from each agent position to its cell centre, Voronoi Vector, when that vector as a value close to zero it means that the agent distance from other agents is near optimal, an example of this interaction between agents is presented in Figure 2.9.

This solved some of SBSP limitations, once it was not needed a base position and the number of players could be variable, for each role.

Delaunay Triangulation [27] shares principles with SBSP. The soccer pitch is divided into triangles based on training data and a map is built from a focal point (e.g. ball position) to the positioning of players (Figure 2.10). A unique Delaunay Triangulation is obtained if more then three points are used. Constraints are used to solve topological relations between different sets of training data, in order to attain more flexible formations. Though simple, this positioning method manages to obtain reasonable approximation accuracy, and is fast running, adjustable, and scalable.

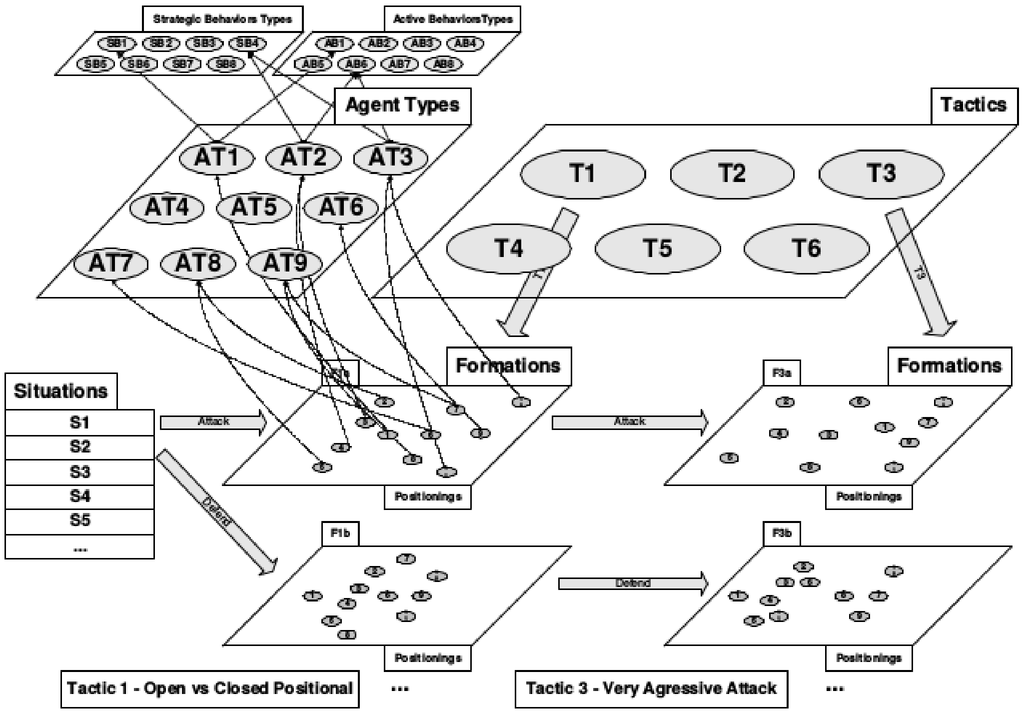


Figure 2.8: Example of a team strategy tactic using SBSP [25].

In 2008 it was presented for the four-legged league a positioning based on potential fields [28], which can be considered in line with SPAR, described earlier. Charges are placed around the field positions, attractive charges at the point that is desirable for the robot to move to, and repulsive at places that should avoid. By aggregating all potential information results in a map with the position to move and information of how to move there as it is shown in Figure 2.11. Besides defining positioning, the algorithm also determines the participants roles.

2.3 CAMBADA Software Agent

The CAMBADA software agent is composed by several well define modules, that when combine lead to a more strategic solution, as represented in Figure 2.12.

The `Integrator` is the module responsible for gathering information from the sensors, filtering it and updating the agent's internal state of the world.

World state information such as robot's position, velocity, role, behavior, perceived ball

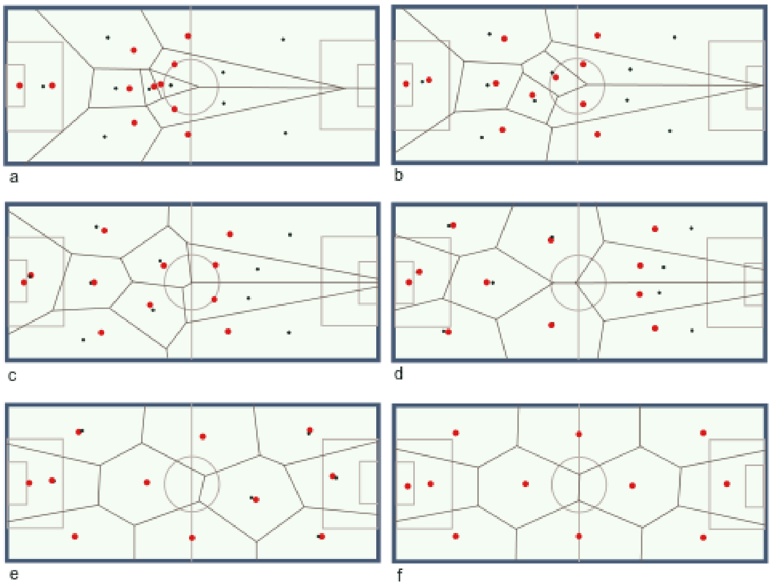


Figure 2.9: Example of agents' movement using DPVC without attraction [26].

position among other information is hold on the `WorldState` module.

The module that is responsible for selecting the `Role`, based on the current game conditions and some history, is the `Decision`.

Each `Role` has an `Arbitrator`, which is a behavior management module, that calls `Behaviors`.

`Roles` use `Behaviors` to achieve their objectives. In the constructor of each `Role` a set of `Behaviors` is added to the `Arbitrator` queue, that during the game selects which one will trigger, basing its decision on the `Invocation Condition (IC)` and `Commitment Condition (CC)` conditions of the behavior [30].

In November of 2008 during the first RoboCup MSL Workshop, held in Kassel, Brainstormers Tribots (Neuroinfromatics Group, from University of Osnabruck) presented their Behavior-Based approach. In their implementation were included two additional conditions, `IC` and `CC`, to the interface extension of the behavior. The IC specifies the conditions that will trigger the behavior, thus the CC checks if the conditions allow the permanence in the behavior.

This approach was adapted in CAMBADA team architecture, resulting in a `Behavior` base class, composed with three methods that have to be implemented by the derived behaviors:

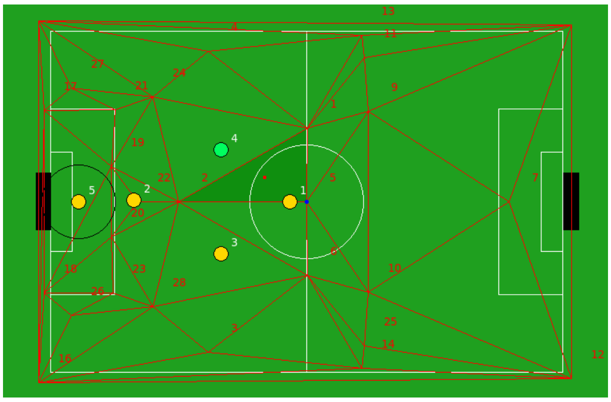


Figure 2.10: Example of a Delaunay Triangulation map.

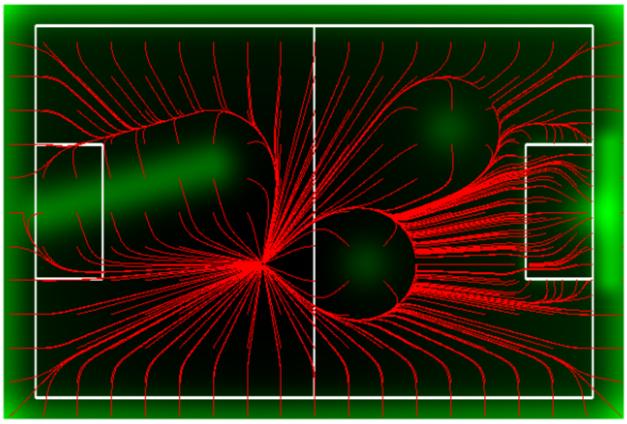


Figure 2.11: An example of the output of the potential fields visualizer. Lighter green indicate regions of higher potential, red lines indicate expected paths of the robots [28].

- virtual void calculate(DriveVector* dv)

  The main block of the Behaviors, where it calculates a DriveVector with the desired velocities and the kicker and grabber states.

- virtual bool checkIC()

  The derived Behaviors have to implement this method and define the Invocation Conditions inside.

- virtual bool checkCC()

  The derived Behaviors have to implement this method and define the Commitment

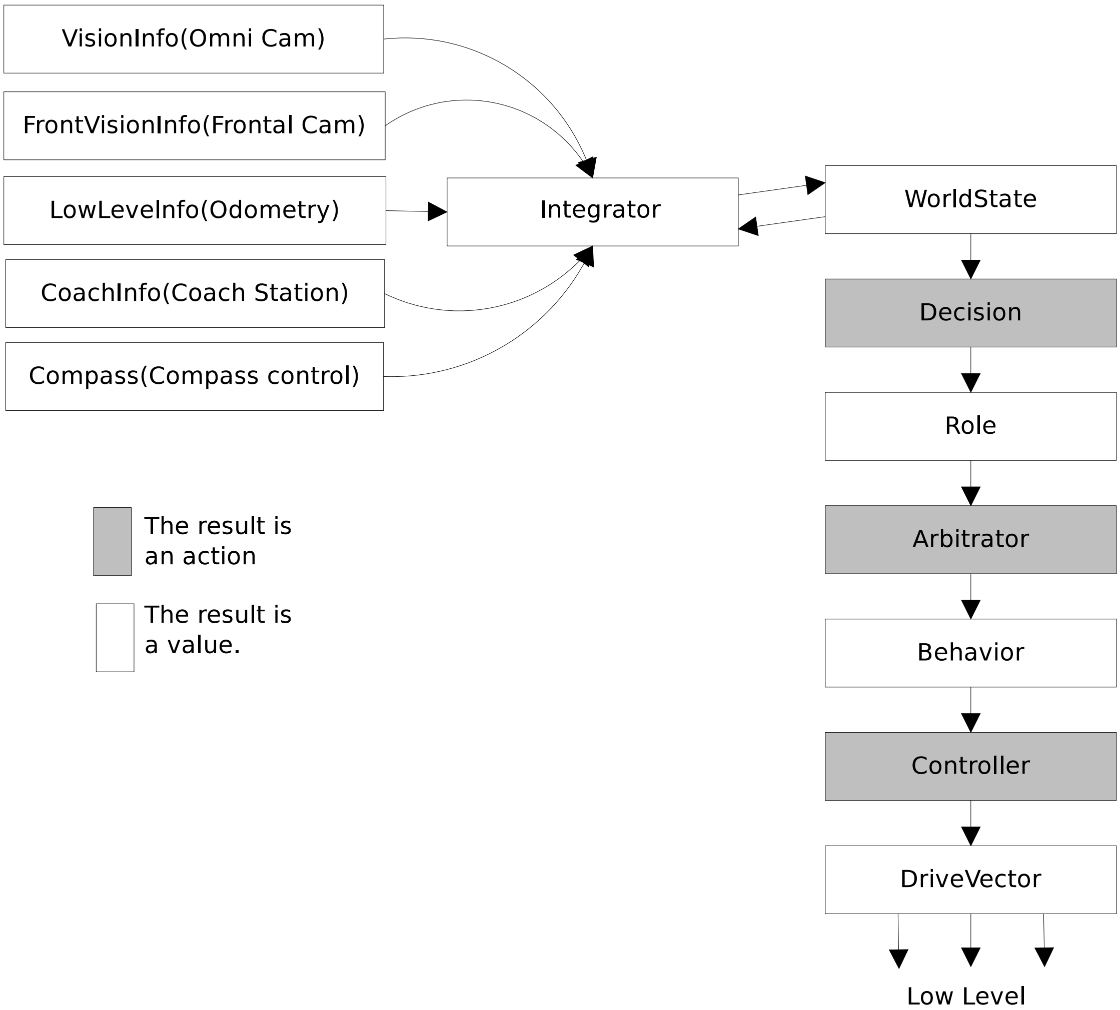


Figure 2.12: Visual representation of the relation between the different modules composing the CAMBADA software agent.

Conditions inside.

- virtual void gainControl()

  The derived Behaviors can implement this method to be notified when they gain control. The callback can be used, for instance, to initialize variables.

- virtual void loseControl()

  The derived Behaviors can implement this method to be notified when they lose control to other Behavior. The callback can be used, for instance, to clean up variables.

In CAMBADA architecture each `Role` contains an `Arbitrator` to select a `Behavior` that then calls a certain `Controller` to calculate a `DriveVector` with all the required low-level information [30]. In this thesis, the work developed was made at the Behavior and Role level.

Chapter 3

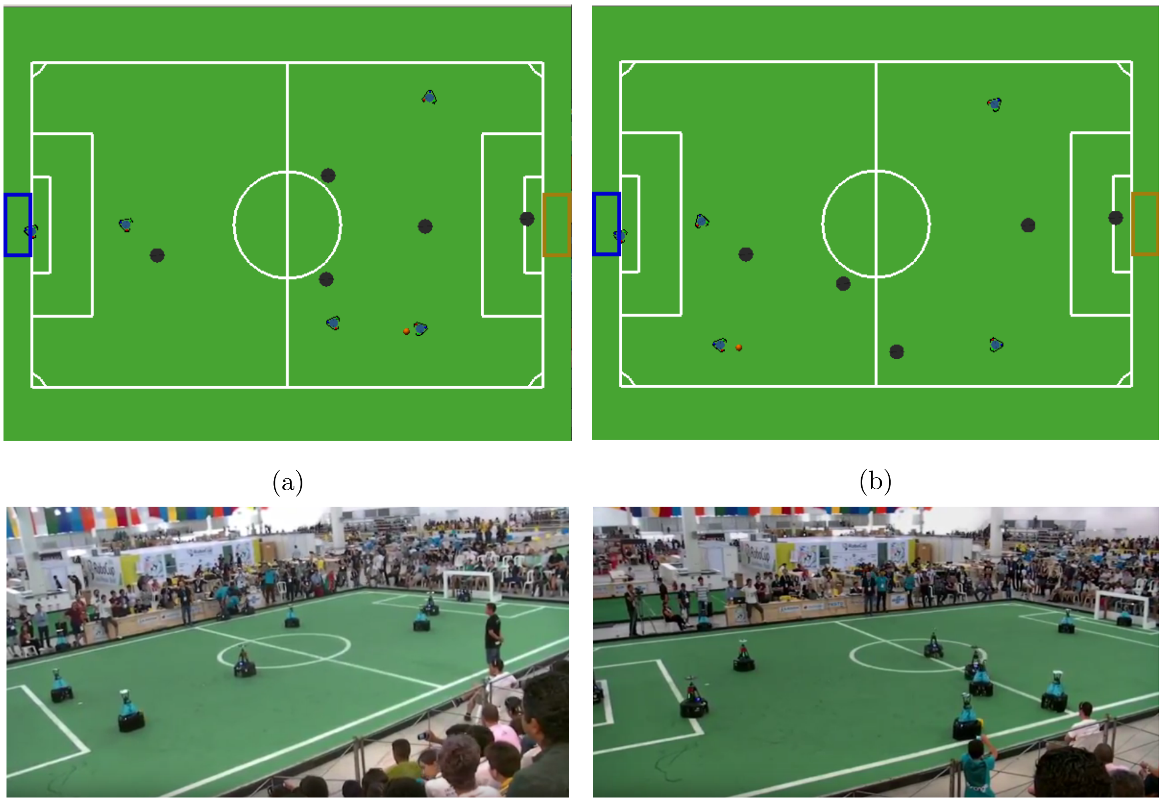
Set Pieces

In a robotic soccer game, as it happens in human soccer, there are two distinct game situations: Free play and Set Pieces. Free play occurs when the ball is played continuously, and the players have to adapt, in the moment, their actions to the conditions and the situation of the game. Set Pieces, which is a studied play, happens whenever the players are confronted with a specific situation (the one studied where the ball position is for all purposes stationary), take special roles and act as planned in advance. The adopted strategy and implementation, also transversal to the human soccer, varies from one team to another.

Set Pieces can be classified as: offensive or defensive. In the CAMBADA team offensive Set Pieces situations there are two roles involved: `Role Replacer` and `Role Receiver`. Otherwise, when defensive Set Pieces happens the role involved is the `Role Barrier`. In the next section is explained in a simplified way how the behaviors are selected inside the roles. Followed by an explanation of the roles presented in the Set Pieces, as well as the behaviors contained in each one of them.

3.1 Offensive Set Pieces

Towards offensive Set Pieces situation the agents assume two different roles, one being responsible for the placement of the ball back in the game, the other three (since one of the agents is always the goalkeeper) position themselves in strategic positions across the field, in order to maximize a good reception of the ball as well as ensure a good attack after the reception. In Figures 3.1a and 3.1b are displayed two examples of offensive Set Pieces,



(a) (b) (c) (d)

Figure 3.1: Offensive Set Pieces situations.

---

obtained using the simulator, being the black circles representing the opponent team players, and the blue "triangles" representing CAMBADA team players. The remaining two Figures (3.1c and 3.1d) represent real offensive Set Piece situations, that occurred during RoboCup 2014, against MRL (Mechatronic Research Laboratory) team.

3.1.1 Role Replacer

In offensive Set Pieces situations this role is assumed by the field robot closest to the ball. After assuming the role, the robot can perform the following list of behaviors according to the game status as well as the priority of the behavior:

- BStopRobotGS

  The robot stops immediately, no matter what is doing. According to RoboCup rules the robots have to be prepared to be stopped at every moment of the game, when a stop signal is sent. To satisfy this rule this behavior is present in every role with highest priority. He only start moving again after a new signal, enabling the movement, is sent.

- BReplacerBallPassedStop

  The robot only assume this behavior if the ball was already passed. It goes back one meter and stop, until it assumes another role.

- BSearchBall

  When the position of the ball is unknown to all the team robots, the replacer goes for a tour inside the field trying to find the ball.

- BReplacerPos

  Once the ball position is known, the robot moves to a position close to the ball.

- BReplacerAlign

  After the start signal is given, the robot chooses the point to pass the ball. That point is chosen through the analysis of the information shared by the receivers. Having into consideration the priority of each receiver and the fact that exists line clear between the replacer and selected receiver. If after 8 seconds none of robots signalizes line clear, the replacer aligns with a default point, defined in the configuration of the Set Piece.

- BReplacerPass

  If the robot is already aligned with the chosen point to pass the ball, approach the ball, engage it and checks if there is line clear to the point. In the case that there is line clear, the replacer pass the ball, otherwise abort the behavior and goes back to the previous behavior.

- BStop

  When none of the conditions for the behavior above are match, the robot stop, this is use as default behavior.

3.1.2 Role Receiver

After the `Replacer` is defined, all the other field robots, beside the goalkeeper, assume the `Role Receiver`. After assuming the role, the robot can perform one of the following list of behaviors according to the game situation, as well as the priority of the behavior:

- BStopRobotGS

  The robot stops immediately, no matter what is doing. According to RoboCup rules the robots have to be prepared to be stopped at every moment of the game, when a stop signal is sent. To satisfy this rule this behavior is present in every role with highest priority. He only start moving again after a new signal, enabling the movement is sent.

- BAvoidTheirGoalArea

  If the robot is inside or moving into the opponent goal area, it forces the robot to move outside, aligned with the ball.

- BReceiveBall

  When the ball is passed the robot assigned to receive it, aligns with the ball. If the trajectory of the ball changes from what was initially calculated the robot will recalculate the point which it as to move in order to have a better ball reception.

- BCallThreeMeters

  One of the Receivers enter this behavior when none of them has line clear in order to receive the ball, after a certain time. This behavior was introduced in this thesis and is better explained in Chapter 4.

- BReceiverPosition

  Indicates the position that the robot should move to according to the set play. Once the start signal is given, it calculates alternative positions and gives then to the Replacer, so it can choose the best position to pass the ball, having in consideration the state of the game. When chosen the position, the robot signed to be the receiver moves there.

- BStop

  When none of the conditions for the behavior above are match, the robot stop, this is use as default behavior.

3.2 Defensive Set Pieces

Defensive Set Pieces take place whenever the opponent team has control of the ball after the start signal in a free kick, corner kick, goal kick or throw in scenario. In Figure 3.2 four examples of defensive Set Pieces are shown, two (Figures 3.2a and 3.2b) were obtained

(a) (b) (c) (d)

Figure 3.2: Defensive Set Pieces situations.

through the CAMBADA simulator, as the others (Figures 3.2c and 3.2d) were taken from the 2014 RoboCup's game, played against MRL (Mechatronic Research Laboratory) team.

3.2.1 Role Barrier

This role has a maximum duration of 10 seconds, time during which the other team is forced to put the ball back into the game. After the ball is put back into the game, the robots role switch into two other roles Midfielder or Striker depending on their distance to the ball. When assuming this role the robot can perform one of the following list of behaviors according to the game situation, as well as the priority of the behavior:

- BStopRobotGS

  The robot stops immediately, no matter what is doing. According to RoboCup rules the robots have to be prepared to be stopped at every moment of the game, when a stop signal is sent. To satisfy this rule this behavior is present in every role with highest priority. He only start moving again after a new signal, enabling the movement, is sent.

- BSearchBallBarrier

  When the ball has not been seen since the start of the role, the robot enters this behavior and searches for it.

- BBarrier

  This behavior is responsible for the movement of the robot to its assigned defensive position. That position can be obtained by two ways: either is given by the coach, or calculated on the agent. When given by the coach there is still two options it could be obtained by strategy or by cover. The strategic position is calculated using a tool that combines Delaunay Triangulations method with rules restrictions. The cover position is calculated using height maps and the having in consideration the position of all obstacles on the field, the goal is never allow the opponent team to have a clear pass line with the ball.

  A main objective of this thesis was to test an only cover-strategy in a defensive Set Piece, thus a main problem becomes clear: if both teams are playing with 5 players, at least one element of the defending team will not have a player to cover (as one of the elements of the opponent team is replacing the ball) therefore its position on the field becomes uncertain, and it may decide in a non strategic one. This grows into a worse condition if the opponent team lacks a player. Some changes to this Behavior were developed, and are explained in the Chapter 5.

- BStop

  When none of the conditions for the behavior above are match, the robot stop, this is use as default behavior.

Chapter 4

Alterations in offensive Set Pieces

In an attempt to promote the evolution of the competition and the game, every year slight changes are introduce in the RoboCup rules. In MSL case these changes also intend to gradually make the game rules as similar as possible to the human soccer.

In this chapter its explained one attempt to adapt to one of the general changes: to score a valid goal the ball has to be received or touched by a team mate within the opponent side of the field after rolling freely for at least one meter.

With the aim of maintaining the competitive strategic previously implemented, this behavior was introduced. It grants the movement of one player from its base position to a place where it has a clear line to the ball, so it can receive it successfully.

4.1 BCallThreeMeters

As explained before the Robots during the role receiver indicate their position as well as some alternative positions, near them where they have a clear line to the ball. In some situations even those alternative points do not have line clear. In such cases one receiver will enter this behavior, that was included in the `Role Receiver`, after a certain period of time, defined in the configuration tool.

When entering the `BCallhreeMeters` the first method called is the `checkInvocation-Condiction()` that will check if the robot has all the conditions to perform this behavior: the play time (checks if the time passed after the start signal is above the specified one), the fact that the ball was not yet been passed, there is only one robot assigned with the role replacer, and the robot ID is equal to the chosen one (`ChooseReceiver()` method presented in the algorithm, explained next).

The receiver's ID is selected using *ChooseReceiver()* method, that having the list of all receivers, choses the one that is closer to the ball position. To ensure that the receiver that is in a more defensive position is not allowed to be chosen, an extra condition was added to the method. If the receiver is near our penalty area, even if it is the closest one, it will not be the selected.

Finally the chosen receiver move to the calculated point, situated 3 meters from the place the ball will be passed, as no member from the opponent team can be closer than 3 meters, the receiver has line clear and the replacer can make the reposition of the ball within the rules.

If during the movement to the calculated point, the `Receiver` has no obstacles between him and the ball for a period of time of approximately half second, the flag indicating line clear is changed to true and the finish point changes to its present location. The change in the line clear condition, leads to the fail of the `checkCommitemmentCondition`, resulting in abandon of the behavior by the robot.

In the Figure 4.1, it is shown a simple schematic, with the representation of the behaviors since the start signal until the game situation shift to free play (that occurs when the ball is kicked, or 10 seconds after the start signal), the times on each behavior varies accordantly with the game situation.

Figure 4.1: Behaviors selected during the Role Receiver on the chosen robot for the approximation.

The time between the 0 and t0, is the time passed since the referee marks the fault until he gives the start signal, t1 represents the time defined in the configuration file, where if there is no line clear a robot enters in the `BCallThreeMeters` behavior, t2 represents the time the robot signalizes a clear line (trigging the `BReceiver` behavior), t3 is the time where the ball was already passed and it enters the ball handling behavior, finally t4 represents the end of the Set Piece (t4 can be at maximum 10 seconds).

Is easier to understand the behavior change represented in Figure 4.1 when associated with images taken from a simulation:

- At the beginning of the Set Piece, the player closest to the ball assumes the `Replacer` (player number 2), letting the other players to be `Receivers` (players number 3 ,4 and 5), Figure 4.2.

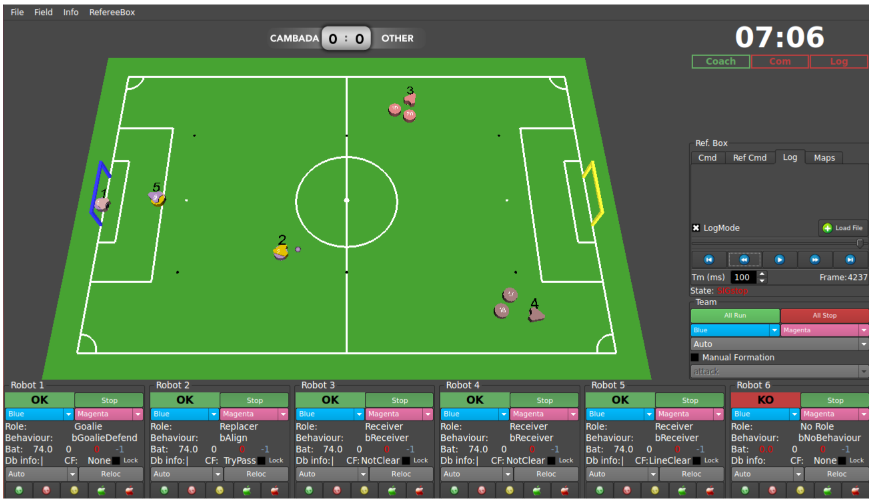


Figure 4.2: Beginning of the set piece.

- After t1 seconds, if none of the receiver signalizes clear line to the ball, the closest one (witch is not defending the team goal), enters the `BThreeMetersCall`, and starts moving until it find a place with clear line to the ball. In Figure 4.3 is shown that the chosen robot was the number 3.

- Having signalized clear line to the ball, at t2, the player number 3, enters the `BReceiverPosition`, and sends to the `Replacer` its new position, Figure 4.4.

- The `Replacer` aligns with the new coordinates, checks if it is indeed line clear, and passes the ball. When this occurs and the `Receiver` perceives that the ball is going

Figure 4.3: Chosen receiver moving to selected point.

---

in is direction, at the instant t3, changes into the `BReceiveBall`, and handles the ball reception, as it is shown in Figure 4.5.

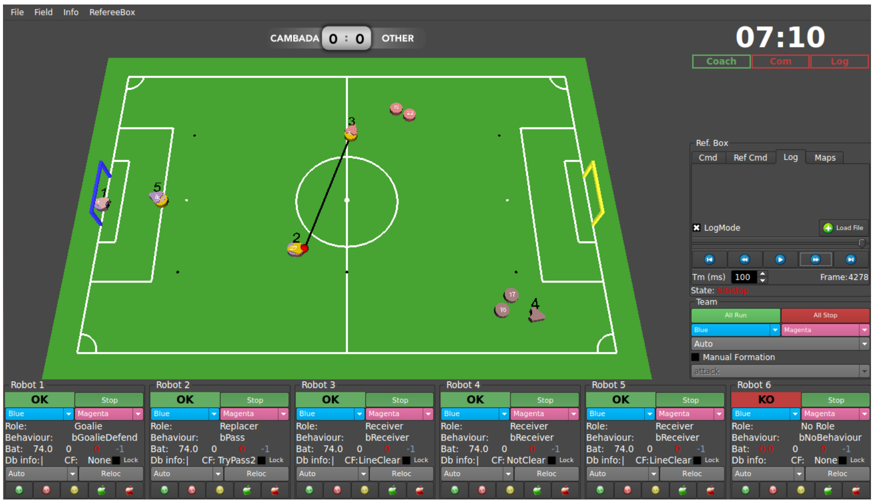


Figure 4.4: Receiver has line clear for more than the assigned time, so stops and wait for the pass.

Figure 4.5: Replacer passing the ball.

Chapter 5

Alterations in defensive Set Pieces

Grid-based representations are used in robotics for more than 20 years thus they are simple to construct, are able to represent arbitrary shape obstacles and can incorporate complex cost functions. Allowing through the merge of all percept information the construction of simple world models with high accuracy. The final representation is then used on planning and decision making purposes, usually oriented to a specific goal.

In CAMBADA team, utility maps were introduced with the intention of improve collective behavior in some specific game situations. Through the merging of all the relevant information about the environment (teammates, obstacles or ball position), conditions, restrictions and used metrics in map, easy decision and positioning is achieved over the analysis of the map values [31].

5.1 Height Maps

In robotic soccer height maps are used to calculate the relevance of certain positions depending on the game situation. They can be used to calculate the alternative position for which the player can move in order to receive the ball in a offensive situation, as well as the position a robot has to move in order to cover one player from the opponent team in a defensive situation. Gathering all information collected by every agent, the maps are built trough the attribution of values to every position on the pitch. The value attribution changes from map to map, thus different types of maps are used in different game situations. The Maps representation is colour coded with the range going from blue to red, being the red the highest utility position, and the blue the lowest. An example of a height map used in CAMBADA is shown in Figure 5.1

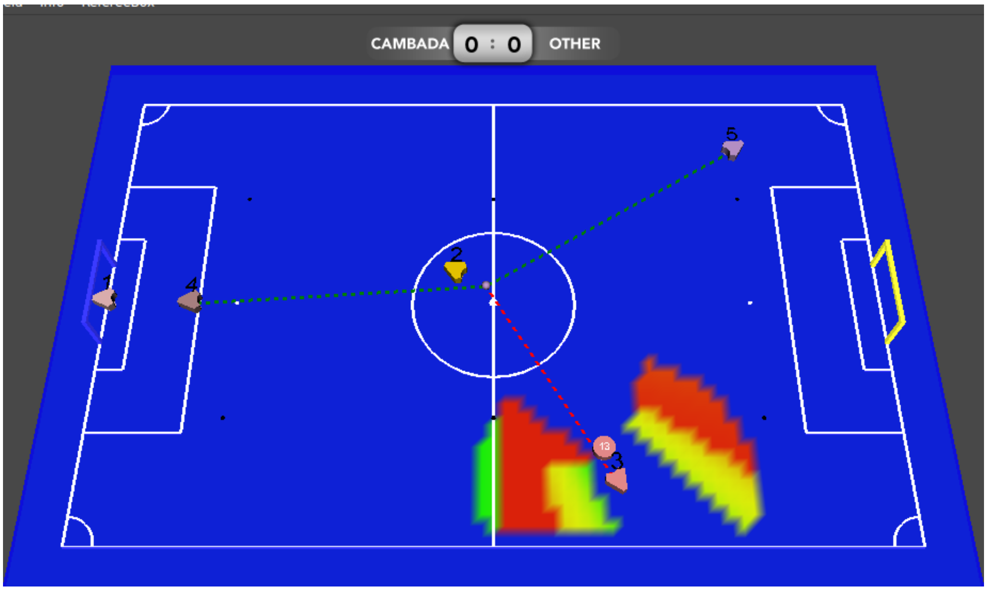


Figure 5.1: 3D visualization height map, use to calculate alternative positions.

Cover Positions

The objective in covering the opponent is preventing it from having a successful reception of the ball, so the agent must be located in between the ball and the possible Receiver.

The cover positions are the result of the combination of two height maps: one responsible for representing, in one map, the obstacles perceived by all agents as a valley in the direction of the ball, the other is responsible for corresponding to each position of the field a cover priority (Figure 5.2). When combined it gives the robots the most relevant places they must move for in order to cover the opponents.

5.2 Configuration Tool

In earlier works, within the project, an user interface called configuration tool was developed. The tool allowed the insertion of new data and the easy manipulation of already existing one, permitting this way the fast adaptation of some variables without having to go into the main code.

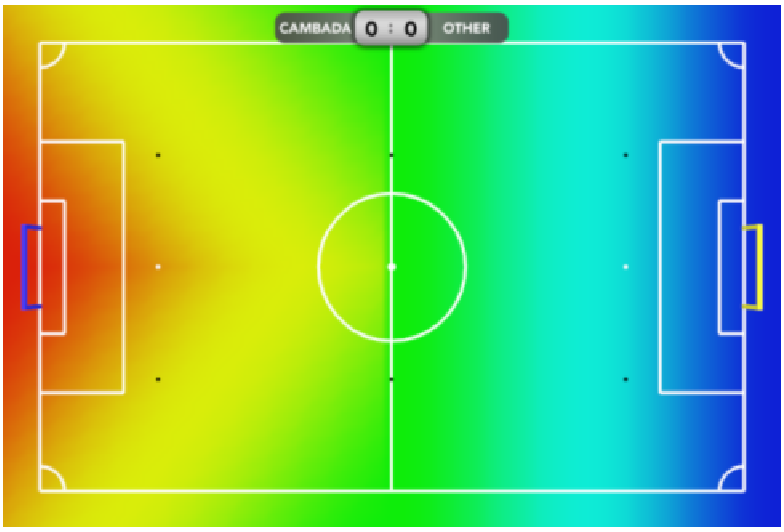


Figure 5.2: Map of the cover priorities.

Within this thesis the configuration tool was edited, so the addition of points of interest distributed by the field would be possible, to be used in defensive Set Pieces scenarios. The main goal is remove positioning by DT, and obtain it only through the usage of height maps.

The Configuration tool allows the edition of five different types of data, in five separated tabs:

- **Set Pieces**, where is possible to change the base position of team players in a offensive Set Piece situation according on the zone on the pitch.
- **Control**, allows the edition of control parameters.
- **Parameters**, where variables that are normally changed from game to game are defined (for example cover distance or maximum speed).
- **Field**, permits the fast definition of the field limits, when not playing in official sized fields.
- **Barrier**, tab was edited during this work, now allows the definition of imaginary obstacles, in chosen positions into the field in defensive Set Piece situation.

When saved all changes done in those tabs are written into a .xml file, a data binding compiler is used, where C++ classes are generated (representing the given vocabulary). Allowing the creation of methods to manipulate the information (write, read, add and change data) in the main (C++) code.

The graphical output was changed using QT, once it provides a large set of libraries as well as the GUI (Graphical User Interface) components(e.g. XML parsing, threads management, network support), all in a consistent style and all multi-platform.

In Figure 5.3 a simple diagram of the events chain that lead to a change in the configuration XML file is presented.

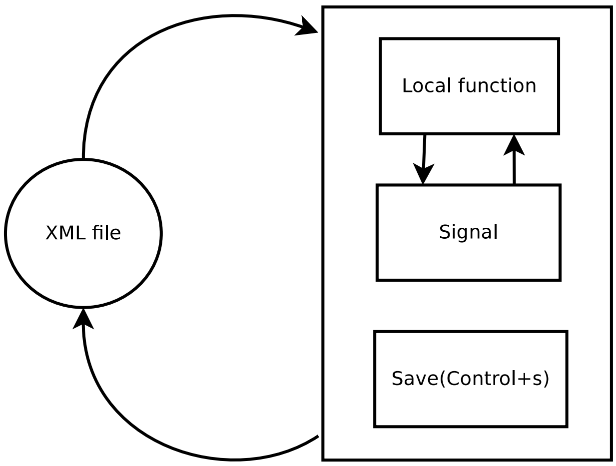


Figure 5.3: Configuration tool, with the Barrier tab open.

When the tool is open, all information in the XML file is read and written into temporary variables. Every time an element is selected, it triggers a signal that is connected with a specific function assigned to it, that allows the edition of its conditions locally. However this alterations may be visible, they only are written in the XML file when saved.

5.2.1 Tab Barrier

In an attempt to define strategical positioning using only cover-positioning strategy in a defensive Set Piece situation, Barrier tab was edited. It allows the definition of the position of imaginary objects through the field, having in consideration its distance to a reference point: that can be the ball or the center of the goal. This leads to the positioning of the players into the field depending only on the game situation, the positioning of the robots from the opponent team and the position of the ball. Letting out the need of the extensive calculations to assign strategical positions.

On the image (Figure 5.4) the ball is fixed in the center of the field, only positions with a minimum 3 meters distance to the ball are valid in a defensive situation, a yellow circle is draw to symbolize that mark. It is possible to define up to 4 positions. As each obstacle as its own attributes in Figure 5.4 all obstacles are shown (for better understanding of the following images).

Figure 5.4: Configuration tool, with the Barrier tab open.

When saved, the positions of the obstacles will be saved in real coordinates, as well as the reference point assigned for each one. The imaginary obstacles are then insert into the defensive height map, as any other (real) obstacle, to be perceived by the robots has place of interest to defend as shown in Figures 5.5 and 5.6. When there is no communication the number of imaginary obstacles added, to each agent maps may vary with the number of opponents detected on the field, which in some cases may not be the real one, thus depending on the distance it is, the opponent, may not be perceived by all team members.

As the ball position is closer to a side line, a problem becomes clear, if the obstacle was define on that side off the ball, is reference position will be outside the field, therefore it is not a valid point to defend. As all the reference points must be valid, the outside point is then recalculated to a place inside the field.

In the pursue of the better method to calculate a new valid position several ones were tested. The main problem with most of the solutions was the overlapping of points and the positioning of imaginary elements in the opponent side of the field, which is clearly not a good defensive positioning when the reposition of the ball is being done in our side of the field.

After the static obstacles are added to the map, the final position of each robot will be assigned by the coach, as the imaginary elements are treated as obstacles the robot final position will not be the one saved in the .xml file but one in between the ball and the saved point.

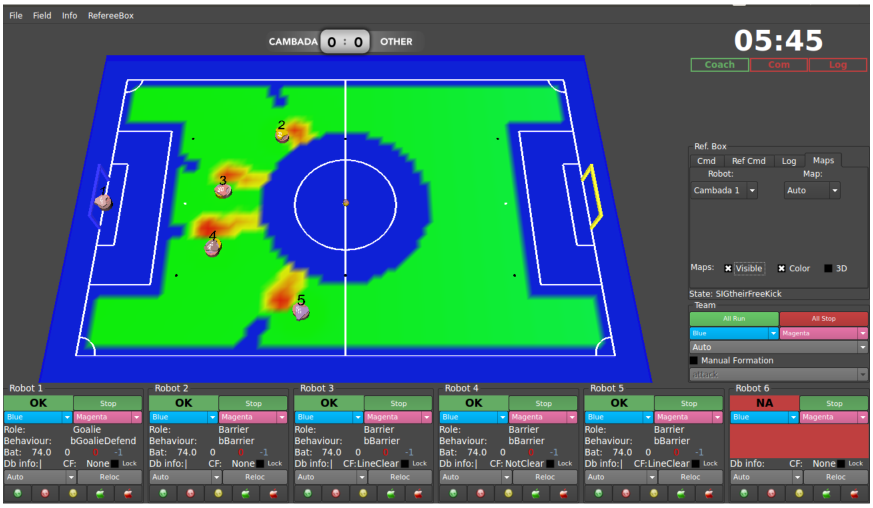


Figure 5.5: Defensive map using with the imaginary obstacles positions define as showed in Figure 5.4.

Through the analysis of the Figure 5.5 we can see that the obstacles are well positioned on the map. In the Figure 5.6 is perceptible that their changed accordantly with the ball position, to the exception of the obstacle number one which had as reference position the centre of the goal, and the obstacle number three that had to be relocated in the field, because maintaining its relative position to the ball, the point would be out of the pitch.

5.2.2 BBarier alterations

The base positions that before were calculated using Delaunay Triangulations were removed from the BBarrier, being the positioning done only by covering perceived points of

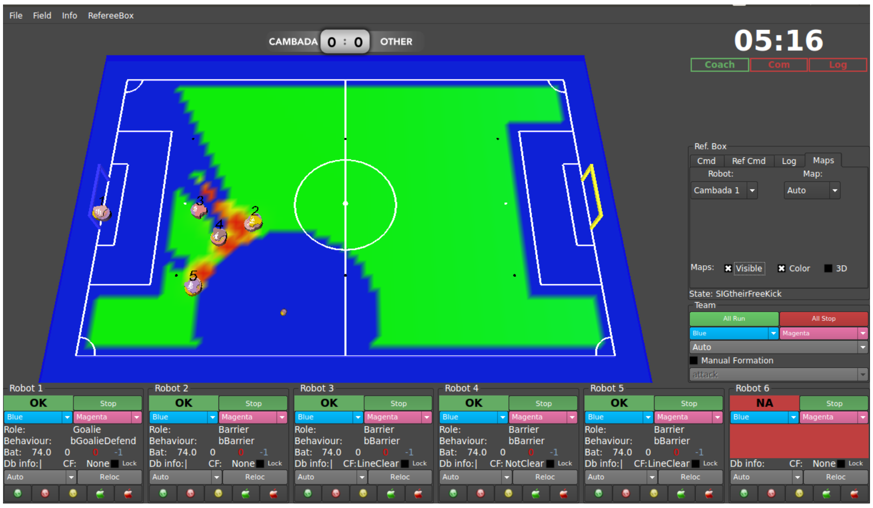


Figure 5.6: Defensive map using whit the imaginary obstacles positions define as showed in Figure 5.4, with the ball in on the side.

---

interest.

That adds extra dynamism to the plays, since the positioning is adapted to the game situation and the other team members. This alteration allows the simplification of the positions attribution, thus the calculation as well as the assignment of positions depends only on the number of players in the field (from our team and from the opponents).

Though the imaginary obstacles addiction, and robots positioning worked well when there was good communication with the coach. When it failed the robots would be lost, hence the assigned positions could be totally different. To avoid those situations, the maps and the coach library were added to the `BBarrier` behavior so it could calculate locally the same imaginary positions that were calculated by the coach.

Using this approach, even if the communication with coach failed totally, the robots would have a strategic position to go for. Comparing Figures 5.6 and 5.5 where positions are assigned by the coach with the Figures 5.7 and 5.8 where there is no coach is noticeable that the positioning is done correctly.

Figure 5.7: Players positioning using defined places in Figure 5.4 without coach.

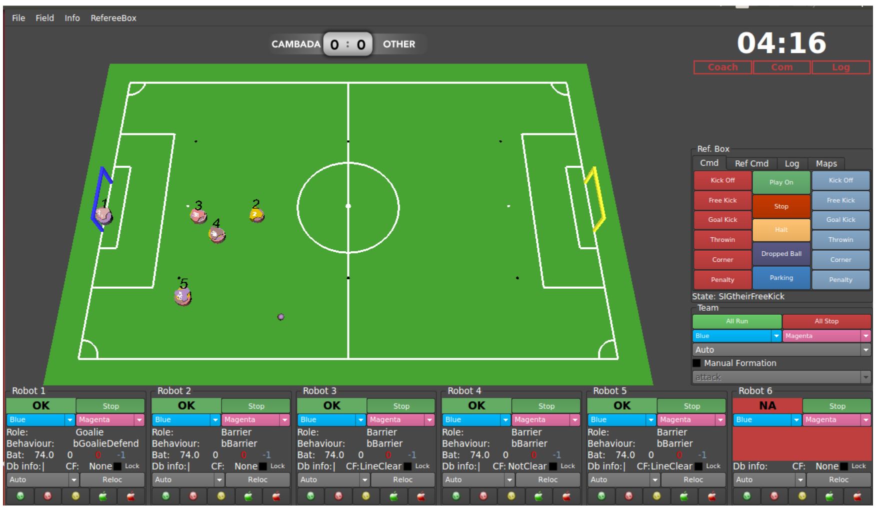


Figure 5.8: Players positioning using defined places in Figure 5.4 without coach.

Chapter 6

Conclusions

This thesis had two main objectives from the beginning, one being the implementation of a behavior that improved the success rate of the ball reposition in offensive Set Pieces and the other being the implementation of an alternative positioning method for defensive Set Pieces.

In the beginning of this work, the reposition of the ball in Set Pieces situations, although highly dynamic, did not contemplate the fact of the receivers players could not have line clear to the ball, resulting in a pass to a default position after 8 seconds, which could be, by the rules, an infraction. Now to score a valid goal the ball has to be played for at least two players of the team and a teammate must be close to the point the ball was passed (at least 3 meters).

Within this work that situation changed, thus when there is no condition to make a pass, one robot moves until it finds a position where there is a clear line, or in the worse case scenario moves to a point three meters from the ball. Although it was not tested in official games, the solution proved to be working in all simulations, as well as in the testes made in the CAMBADA training field.

The positioning in a defensive Set Piece scenario used in CAMBADA is a mix of covering opponents distributed by the field, and attributed positions calculated using the Delaunny Triangulations, being the number of the opponents to cover defined in the begging of each match. That had two problems, one being the fact that if all players were assign to be covering the opponent, at least one would not have a opponent to cover, the other being the case where there has more players to cover than the assigned number, lead to a possible advantage to the opponent team.

The proposed method assigns to all players cover positions even if there is no obstacle on the pitch, thus the number of imaginary obstacles vary with the number of real detected ones, leading to a better defence strategy, not allowing opponents distributed on the field to have clear view to the ball. The method proved to be working both in simulation as in the CAMBADA laboratory. When tested with no obstacles in the field some problems appear in corner situation, thus one or two point overlap, other then that when at least two obstacles were add to the field it proved to be working. In Figure 6.1, it is illustrated the evolution of positioning methods, in defensive situations within CAMBADA Team, along with the proposal of this work.

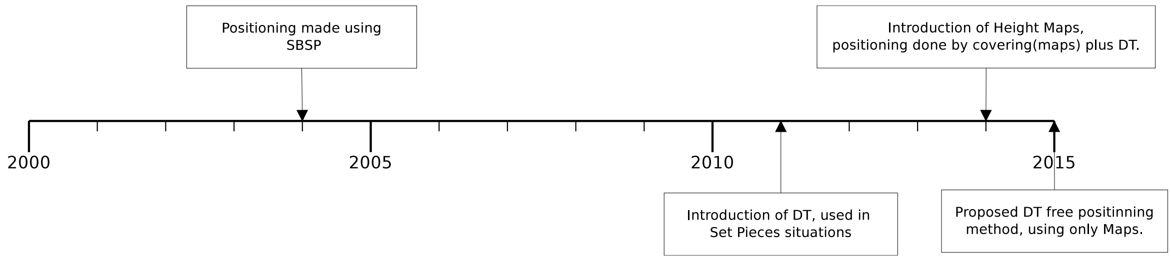


Figure 6.1: Positioning methods introduced (used) in CAMBADA through the years.

6.1 Future Work

The offensive Set Pieces has already a strategy that adapts to the opponent team game, but the positioning as well as the set pieces itself is quite limited thus when the ball is passed the game changes almost immediately to free play situation. It would be very interesting the addition of extra layers of movements and the continuation of the set play with more levels of complexity.

On the positioning method on the defensive Set Pieces situation presented would be very interesting to study/adjust a, more suitable, algorithm to relocate the players into the field, so problems mentioned before such as overlaps do not occur. That could lead to a better distribution of players in the field, maintaining the simplicity and the dynamism of the proposed method.

Bibliography

[1] Jess S. Shamma. *Cooperative Control of Distributed Multi-Agent System*, John Wiley & Sons Ltd, West Sussex, 2007.

[2] Stuart Russel, Peter Norvig. *Artificial inteligence: a modern aproach*, Pearson Education International, second edition, 1995.

[3] RoboCup. *http://www.robocup.org/about-robocup/objective/*, Accessed on 26 September, 2015.

[4] Gerhard Weiss. *AMultiagent Systems A Modern Approach To Distributed Artificial Intelligence*, Massachussets Institute of Technology, third edition, 1999.

[5] Liviu Panait and Sean Luke. *Cooperative Multi-Agent Learning:The State of the Art*, George Mason University.

[6] Hitoshi Matsubara, Itsuki Noda and Kazuo Hiraki. *Learning of Cooperative actions in multi-agent systems: a case study of pass play in Soccer*, AAAI-96, 1996.

[7] Peter Stone and Manuela Veloso. *Multiagent systems: A survey from a machine learning perspective.*, *Autonomous Robots*, 8:345383, 1997.

[8] RoboCup. *https://www.aldebaran.com/en/humanoid-robot/nao-robot*, Accessed on 16 September, 2015.

[9] *https://www.robotics-erlangen.de/software*, Accessed on 16 September, 2015.

[10] RoboCup. *https://www.aldebaran.com/en/humanoid-robot/nao-robot*, Accessed on 16 September, 2015.

[11] Humanoid League kids size. *http://www.nimbro.net/OP/*, Accessed on 16 October, 2015.

[12] Small Size League squematic. *https://www.robotics-erlangen.de/software*, Accessed on 16 October, 2015.

[13] Small Size League CMU team. *http://www.cmu.edu/news/stories/archives/2015/july/images/robocup_-853x480-min.jpg*, Accessed on 16 October, 2015.

[14] SimSpark, 3D simulation League. *http://simspark.sourceforge.net/wiki/index.php/Soccer_-Simulation*, Accessed on 16 October, 2015.

[15] Robin Soetens , René van de Molengraft, Bernardo Cunha. *RoboCup MSL - History, Accomplishments, Current Status and Challenges Ahead*, 2014.

[16] JS Gutmann, W Hatzack, I Herrmann, B Nebel, F Rittinger, A Topor, and T Weigel. *Reliable Sel-Localization, Multirobot Sensor Integrartion, Accurate Path-Planning and Basic Soccer Skills: Playing an Effective Game Of Robotic Soccer*, 1999.

[17] Andrea Bonarini, Paolo Aliverti, and Michele Lucioni. *An omnidirectional vision sensor for fast tracking for mobile robots* In *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE, volume 1, pages 151–155.* IEEE, 1999.

[18] Carlos F Marques and Pedro U Lima. *A localization method for a soccer robot using a vision-based omni-directional sensor.* In *Robocup 2000: Robot Soccer World Cup IV*, pages 96–107. Springer, 2001.

[19] Luca Iocchi and Daniele Nardi. *Self-localization in the robocup environment.* In *Robocup 99: Robot Soccer World Cup III*, Springer, 2000.

[20] Pedro Lima, Andrea Bonarini, Carlos Machado, Fabio Marchese, Carlos Marques, Fernando Ribeiro, and Domenico Sorrenti. *Omni-directional catadioptric vision for soccer robots.* Robotics and Autonomous Systems, 2001.

[21] Hans Utz, Alexander Neubeck, Gerd Mayer, and Gerhard Kraetzschmar *Improving vision-based self-localization.*, In *"RoboCup 2002: Robot Soccer World Cup VI"*, pages 25–40, Springer, 2003.

[22] Matthijs TJ Spaan and Frans CA Groen. Team coordination among robotic soccer players *Team coordination among robotic soccer players.*, In *"RoboCup 2002: Robot Soccer World Cup VI"*, pages 25–40. Springer, 2003.

[23] Stefan Schiffer, Alexander Ferrein, and Gerhard Lakemeyer. *Qualitative world models for soccer robots.*, In *Qualitative Constraint Calculi, Workshop at KI* volume 2006, pages 3–14, 2006.

[24] Peter Stone. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer.* MIT Press, 2000.

[25] Luis Paulo Reis, Nuno Lau, and Eugénio Oliveira. *Situation based strategic positioning for coordinating a simulated robosoccer team.*, In M. Hannebauer, J. Wendler, and E. PagelloIn, editors, *Balancing React. and Social Deliberation* in MAS 2013 of *Lecture Notes in Artificial Intelligence* pages 175–197. Springer, 2001.

[26] Hesam Addin Torabi Dashti, Nima Aghaeepour, Sahar Asadi, Meysam Bastani, Zahra Delafkar, Fatemeh Miri Disfani, Serveh Mam Ghaderi, Shahin Kamali, Sepideh Pashami and Alireza Fotuhi Siahpirani. *Dynamic Positioning based on Voronoi Cells (DPVC).*, Ansgar Bredenfeld, Adam Jacoff, Itsuki Noda, and Yasutake Takahashi, editors. *RoboCup-2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Artificial Intelligence* pages 219–229. Springer, 2006.

[27] Hidehisa Akiyama and Itsuki Noda. *Multi-Agent Positioning Mechanism in the Dynamic Environment.*, Ubbo Visser, Fernando Ribeiro, Takeshi Ohashi, and Frank Dellaert, editors. *RoboCup-2007: Robot Soccer World Cup XI*, volume 5001 of *Lecture Notes in Artificial Intelligence* pages 377–384. Springer, 2008.

[28] Henry Work, Eric Chown, Tucker Hermans, Jesse Buttereld, and Mark McGranaghan. *Player positioning in the four-legged league.*, Luca Iocchi, Hitoshi Matsubara, Alfredo Weitzenfeld, and Changjiu Zhou, editors. *RoboCup-2007: Robot Soccer World Cup XII*, volume 5399 of *Lecture Notes in Artificial Intelligence* pages 391–402. Springer, 2009.

[29] Luís Mota *Multi-robot Coordination using Flexible Setplays: Applications in RoboCups Simulation and Middle-Size Leagues.* In *Faculdade de Engenharia da Universidade do Porto*, 2011.

[30] Ricardo Dias. *Agent Architecture of the CAMBADA Robotics Soccer Team.* In *Universidade de Aveiro*, 2014.

[31] R. Dias, F. Amaral, J. L. Azevedo, R. Castro, B. Cunha, J. Cunha, P. Dias, N. Lau, C. Magalhães, A. J. R. Neves, A. Nunes, E. Pedrosa, A. Pereira, J. Santos, J. Silva, and A. Trifan. *CAMBADA'2014: Team Description Paper.* In http://robotica.ua.pt/CAMBADA/docs/qualif2014/CAMBADA-tdp-2014.pdf .

[32] Hiroaki Kitano and Satoshi Tadokoro. *RoboCup Rescue A Grand Challenge for Multiagent and Intelligent Systems*, AI Magazine Volume 22 Number 1, 2001.