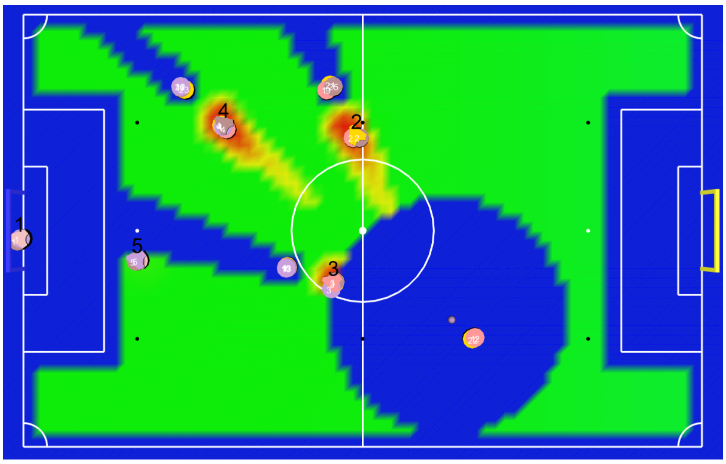
Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2014

Filipe Alves Amaral

Posicionamento estratégico nas jogadas de bola parada para a CAMBADA

Strategic positioning during Set Pieces for the CAMBADA soccer team



Universidade de Aveiro Departamento de
Electrónica, Telecomunicações e Informática,
2014

Filipe Alves Amaral

Posicionamento estratégico nas jogadas de bola parada para a CAMBADA

Strategic positioning during Set Pieces for the CAMBADA soccer team

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Nuno Lau (orientador), Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor António Neves (coorientador), Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president	**Prof. Doutor Luís Filipe de Seabra Lopes** Professor Associado da Universidade de Aveiro
vogais / examiners committee	**Prof. Doutor Eduardo Alexandre Pereira da Silva** Professor Adjunto do Instituto Superior de Engenharia do Porto
	Prof. Doutor António José Ribeiro Neves Professor Auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

Aos meus orientadores e todos os membros da equipa CAMBADA pelo bom ambiente de trabalho e constante disponibilidade para ajudar, um obrigado.
Ao Ricardo e João, um especial agradecimento pelo apoio, esclarecimentos e troca de ideias.
Aos meus amigos e colegas que me acompanharam nestes anos.

Aos meus pais pelo apoio incondicional.

Por último, um muito obrigado à Cláudia pela ajuda, apoio e motivação nestes ultimos tempos.

Resumo

A robótica estuda o design, construção e uso de máquinas e ou agentes, designados por robôs, para executar tarefas com algum nível de complexidade. Dentro desta vasta área de conhecimento científico, a coordenação em sistemas multi-robô tem sido alvo de especial atenção, merecendo um papel de destaque no domínio do futebol robótico. A maneira como cada equipa coordena os seus robôs para o desempenho de acções cooperativas define a base da sua estratégia e em grande parte o sucesso das suas jogadas.

CAMBADA (Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture) é a equipa de futebol robótico da Universidade de Aveiro que participa na RoboCup Middle Size League. Esta equipa foi criada e é desenvolvida por alunos e docentes pertencentes à unidade de investigação IEETA e ao DETI.

Este trabalho tem como principal objectivo melhorar o posicionamento dos robôs nas jogadas de bola parada, tornando-as assim mais dinâmicas e facilmente adaptáveis às diferentes estratégias usadas pelas outras equipas em ambiente de jogo. As diversas alterações efectuadas aos algoritmos já existentes, assim como as novas condições incluídas e a sua aplicabilidade, são descritas ao longo desta dissertação.

O trabalho desenvolvido foi testado em laboratório e utilizado nos robôs na competição Robótica 2014. Embora o desenvolvimento não estivesse ainda completamente concluído à data da competição, é de salientar a maior eficácia verificada na estratégia da equipa nas situações de jogadas de bola parada, tornando a equipa mais competitiva e capaz de defender e atacar melhor.

Abstract

Robotics studies the design, construction and use of machines and or agents designated by robots, to perform tasks with some level of complexity. Within this scientific domain, the coordination in multi-robot systems has been receiving a special attention, deserving a prominent role in the robot soccer issues. The way that each team coordinates its robots individually and together in order to perform cooperative tasks is the base of its strategy and in large part dictates the success of the team in the game environment.

CAMBADA (Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture) is the robot soccer team of the University of Aveiro, which participates in the RoboCup Middle Size League. This team was created and developed by students and teachers belonging to the IEETA research unit and the DETI.

The aim of this work is the improvement of the strategic positioning in the set pieces situations, thus making it more dynamic and easily adaptable to the different strategies used by the opponent teams. The various changes made to existing algorithms, as well as new conditions included and their applicability are described throughout this dissertation.

The work developed has been tested in the laboratory and used in robots in the Robótica 2014 competition. Although the development had not yet been fully completed by the date of the competition, it is important to emphasize the increased efficacy observed in team strategy in situations of set pieces, making the team more competitive and able to defend and attack better.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the recent years, the multi-agent systems has been acquiring high importance, once this type of decentralized systems are extremely useful in the resolution of typical complex problems and situations in a wide variety of technological and industrial environments, such as robotics, sensor networks, distributed control, collaborative decision support systems and supply chains, among others. These systems are based in the well *coordination* and *cooperation* of multiple agents that using joint behaviors with some degree of complexity are able to solve joint tasks and maximize their capacity.

This thesis was developed with the aim to improve the effectiveness, coordination and dynamics of robotic soccer team CAMBADA[1].

1.1 Multi-agent Systems

The concepts of agent and multi-agent continue to be very controversial and some different definitions are admitted. According to Panait and Luke [4], an agent can be described as a computational mechanism that exhibits a high degree of autonomy, performing actions in its environment based on information received from the environment (sensors, feedback). Multi-agent domain is characterized by the existence of more than one agent interacting all together and in which the agents may not, at a given time, know everything about the world that other agents know, including the internal states of the other agents. This condition permits the agents to act as if they were really mere appendages of a single

[1]CAMBADA is an acronym for Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture.

master controller.

Moya and Tolk [1] say that an agent is an entity authorized to act for another and capable to exist situated in a certain environment (defined as the set of circumstances, objects or conditions that surround the agent and influence its behaviors), perceiving that environment and able to perform autonomous action [5][6]. Some agent properties that need to be taken into account and allow their classification are autonomy, cooperation, mobility, learning, rationality, communication, application, function, class or even capability. Figure 1.1 shows a schematic representation of a typical agent.

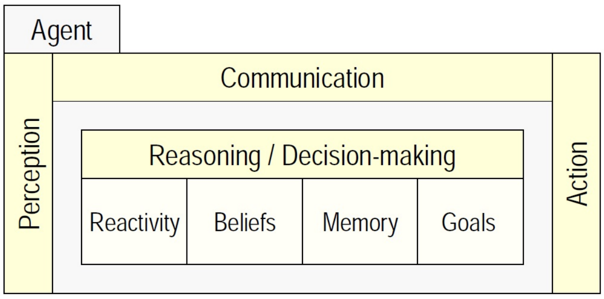


Figure 1.1: Representation of a typical agent from [1].

Considering that a single agent is an autonomous entity situated in an environment, a multi-agent system can be described as a one that consists of multiple autonomous entities that build a population of agents. In these systems, the agents have a limited point of view and the data is decentralized. In Figure 1.2, a scheme of a multi-agent system is represented with their integrant parts.

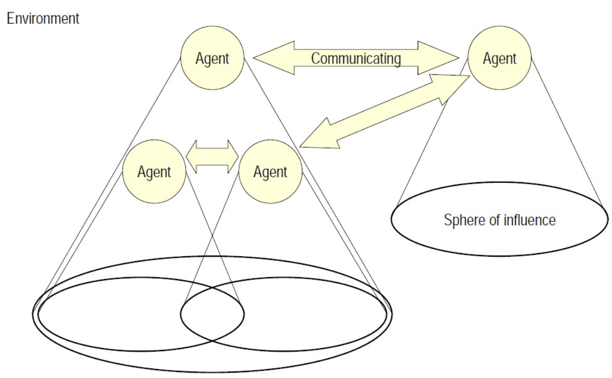


Figure 1.2: Representative multi-agent system from [1].

The agents present in a certain environment can be identical or different in their characteristics and capacities such as size, diversity, homogeneity, overall goal structure and cooperativity (Figure 1.3).

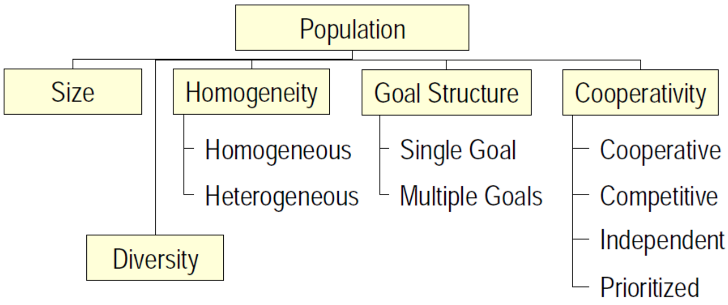


Figure 1.3: Different properties of agent populations from [1].

According to size (the number of system agents) a distinction can be made between single agent populations and multi-agent populations. Diversity takes into account the type of agents in a system regarding their characteristics.

Homogeneity refers to whether the agent population consists of agents of the same type. A system is homogeneous if every agent has uniform structure and composition. This includes but is not limited to goals, rules and architecture. An heterogeneous system is characterized by the presence of different type of agents, with different construction, functions, goals or position in system hierarchy. An heterogeneous system can include both reactive and deliberative agents. Also, an heterogeneous system can include only reactive agents differing in reactive structures.

In a system it is also important to define the goals to be achieved and whether they are structured. In case of a single goal system, all the agents act to achieve a single goal. Sometimes, this goal could be layered in some sub-goals that may vary in the population, not implying that all the agents cooperate in the same way to achieve the goal. In a system with multiple goals, each agent has its own goal to achieve. Also, a single agent could have multiple goals.

Being cooperative within a system permits a mutual benefit for all the agents. The cooperative mechanisms can be implicit or explicit being related with the agents characteristics and influence the overall behavior of the multi-agent system. A set of agents could be considered:

- actively cooperative, giving special importance to the system objective over their individual goals;

- competitive, e.g., in a case where all the agents involved have the same goals and resources to achieve them;
- independent, acting on goals independently without cooperative mechanisms;
- prioritized, when the goals present a prioritization structure that guides pursuit and cooperation.

According to Arkin and Balch [7], a team of simple robots is simpler in terms of individual physical design than a larger/more complex robot and the resulting team is more economic, scalable and less susceptible to overall failure.

A collective or team englobes a collection of completely autonomous agents which communicate through information transference or a team can comprise a number of remotely controlled appendages making the entire collective more suitable to be described as a single large robot with distributed activators.

Robot teams are characterized by enhanced task performance and reliability (redundancy) and decreased cost when compared with more traditional robotic system. It is important that the team presents a collective behavior or a set of actions that accomplishes the same behavior action that was required of the single more complex robot. This cooperative intelligent behavior results from the effective communication established between the team agents. This communication may be done directly via an explicit communication channel or indirectly through one robot that senses a changement in others robots in its environment.

The success of a robot team is related with their information processing ability which depends on a large number of factors including the number of units, their sensing abilities, their communication mechanisms among others. [8, 9]

With the robots becoming increasingly used in a wide variety of human activity areas, its is necessary to investigate and understand the cooperation and coordination of autonomous robots to well perform tasks achieving high quality overall system performance. Communication and information sharing via wireless technologies makes possible an accurate coordination between the agents that partially explains the increasing use of multi robot systems. Multi robot systems also presents some advantages with respect to single robots. Some tasks are difficult or even impossible to be performed by a single robot. In certain situations, multi robot systems are capable to perform tasks in a faster way and present higher scalability once problems can be solved by adding more agents to the team.

Multi-robot systems can be more robust and even when a team member is damaged or malfunctioning the team can still work [10].

Multi-robot systems are widely used in various areas such as exploration of unknown or changing environments [11, 12, 13], mapping [14], foraging [15], transportation [16], manufacturing [17], intrusion detection and patrolling [18, 19] or entertainment [20].

1.2 Problem, Objective and Developed Work

RoboCup Middle Size League(MSL) competition introduce some great challenges to the robotic teams. Robots must be coordinated and cooperative in their actions and positioning in the field in order to prevent the opponent team from scoring. Also accurate skills in dribbling, shooting and passing are extremely important so that the team can successfully play the game and score.

Conditions such as game strategies, positioning and team coordination play a decisive role in MSL soccer games, being the development of stable and efficient strategies as well as accurate team coordination extremely crucial in the achievement of good results in the competitions.

The main goal of this work is the improvement of team performance specially in the set pieces situations through the upgrading of the existing CAMBADA team strategies and coordination behaviors. For this, the strategic positioning needs to be improved in these referred situations in order to make it more dynamic and adaptable to strategies used by the opponent teams.

After a careful analysis of the existing work, some improvements were done. The aim of the new developed work comprises the improvement of the player at individual level even as the team cooperation and coordination in Set Pieces situations. A more efficient coordination and a more complete overall strategy is due to the improvement of existing behaviors and roles that will integrate and ameliorate the team strategy.

The main tasks realized during this work were focused on the following topics:

- Improvement of the role responsible for the defense on defensive set pieces to a more dynamic positioning concerning the other team, especially regarding the block of the opponent pass lines;

- Upgrade the roles involved on offensive set pieces improving the passer decision as well as the receivers positioning and breakaways;

- Rework of class responsible for team strategy in order to fix known weaknesses;
- Introduction of new functionalities useful for debugging process.

1.3 Thesis structure

This thesis is divided into 6 chapters. In Chapter 1, the Introduction, an overview of the multiagent robotics is shown. Chapter 2 presents a brief description of robot world cup initiative "RoboCup" and Middle Size League competition rules. A special importance is given to CAMBADA (one of the MSL robotic soccer teams) where this work was developed. A description of CAMBADA team history and both software and hardware architecture is made. Chapter 3 emphasizes the usefulness of Utility Maps and the Library chosen to help the development of this work. In Chapter 4, the Defensive Set Pieces are detailed specifying the roles and behaviors, the choice of base positions, the cover positions and the obtained results. In Chapter 5, the Offensive Set Pieces are described taking into account the involved roles and behaviors, the choice of base positions, the receiver selection, position to receive the pass and the obtained results. Chapter 6 focus the changes made to the module responsible for team strategy and improvements done to coach and basestation applications. Finally, the Conclusions of the developed work under this dissertation are presented in Chapter 7.

Chapter 2

RoboCup, Middle Size League and CAMBADA

RoboCup Middle Size League (MSL) is one of the RoboCup Soccer leagues. This league is regulated by a set of rules, based and adapted from FIFA laws, that impose some important conditions and restrictions that needs to be well known in order to understand the game. Also important is the knowledge of the CAMBADA characteristics as well as the inherent limitations of this platform where this work has been developed. So in this chapter an introduction to RoboCup is made followed by a brief description of MSL rules and finally a brief description of the CAMBADA team.

2.1 Robocup

RoboCup ("Robot Soccer World Cup") is a scientific challenge with an annual international meeting and competition that started in 1997 and the first edition took place in Nagoya, Japan. The aim of this initiative is to promote the development of Artificial Intelligence, Robotic and all the related fields worldwide. Several different types of real environments are considered, simulated, studied and analyzed during this competition. One of the most popular is the soccer domain. Since it is a game, it makes possible the development, testing and incorporation of a great number of useful technologies such as: design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, sensor capacities, among others. Actually, the main goal of RoboCup is the development of a soccer team of fully autonomous humanoid robots that by the year of 2050 can play and win against the human world champions, complying with the official FIFA rules [21].

RoboCup englobes an annual world event and some regional events worldwide. Soccer game was chosen to be used as a central topic for further research and innovations to be applied in the fields of socially problems and industrial needs.

The first public announcement for the development of soccer-playing robots occurred in 1993 and within two years some discussions about the organization and technical issues were conducted in numerous conferences, workshops and meetings. Also the first open system simulator for the soccer domain enabling multi-agent systems research was first announced and publicly demonstrated. In August 1995, the intention of organizing the First Robot World Cup Soccer Games and Conferences in 1997 was announced. At the same time, the decision was made to organize Pre-RoboCup-96 (that took place in Osaka, Japan), in order to identify potential problems associated with organizing RoboCup on a large scale. So, the initial group of participant researchers had two years for preparation and start robot and simulation team development. In 1997, the first official conference occurred in Nagoya, Japan. In the following years other cities throughout the globe housed the event: Paris, Stockholm, Melbourne, Seattle, Fukuoka, Padua, Lisbon, Osaka, Bremen, Atlanta, Suzhou, Graz, Singapore, Istanbul, Mexico city and Eindhoven. This year the games will take place at João Pessoa, Brazil, in July.

The annual events had national and international media coverage and, over the years, attracted an increasing number of participants and spectators. in the past 17 years the number of participating teams increased greatly from 38 in Nagoya (Japan) to 310 in Eindhoven (the Netherlands), achieving 3033 participants last year in Eindhoven.

The RoboCup World Championship and Conferences is the main issue of RoboCup activities and it provides a world class moment where participant researchers from all over the world can stay together working and evaluating the global as well as the individual research progress and participate in several competitions testing their own work.

Soccer game is considered as the standard problem, but RoboCup also includes more activities such as Technical Conferences, RoboCup International Competitions and Conferences, RoboCup Challenge Programs, Education Programs and also Infrastructure Development.

Although the creation of RoboCup Project was initially focused on soccer game, during the years the range of activities has increased and now six main domains can be considered:

- RoboCup Soccer
- RoboCup Rescue
- RoboCup@Home
- RoboCup@Work Demo
- RoboCup Logistics League sponsored by Festo
- RoboCup Junior

Each competition focuses on different potential future applications of robots capacity and due to its singularities different leagues enable researchers to address problems in different areas of robotic fields such as hardware, vision or software.

2.1.1 RoboCup Soccer

As it was already said, the main issue in RoboCup activities is the soccer game. Being probably the most popular sport worldwide, soccer easily attracts a large number of people to the event. The most important topics considered in this research field are related to the use of cooperative multi-robot and multi-agent systems in dynamic adversarial scenarios and as it was expected, it brings a set of interesting challenges for researchers:

- as it is a collective game, the agents/robots will be forced to interact through cooperation and competition;
- need to well define the individualistic behavior aspects, since each agent/robot must be able to identify relevant objects, self localize, dribble;
- consider the cooperative elements that will enable the existence of passes, complementary roles and all teamwork alike elements;
- take into account the real time permanent changes in the dynamic and adversarial environment (e.g. the ball movement, the position change of teammates and opponents).

These games are important challenging opportunities for researchers to analyze different game situations and to exchange technical information leading to a great technological evolution. Actually, RoboCup comprises several different leagues which differ from each other in various aspects such as: agents/robots size and construction characteristics, type of communication used, type of environment (real or virtual), number of agents/robots and also the age and knowledge level of the participants, among others.

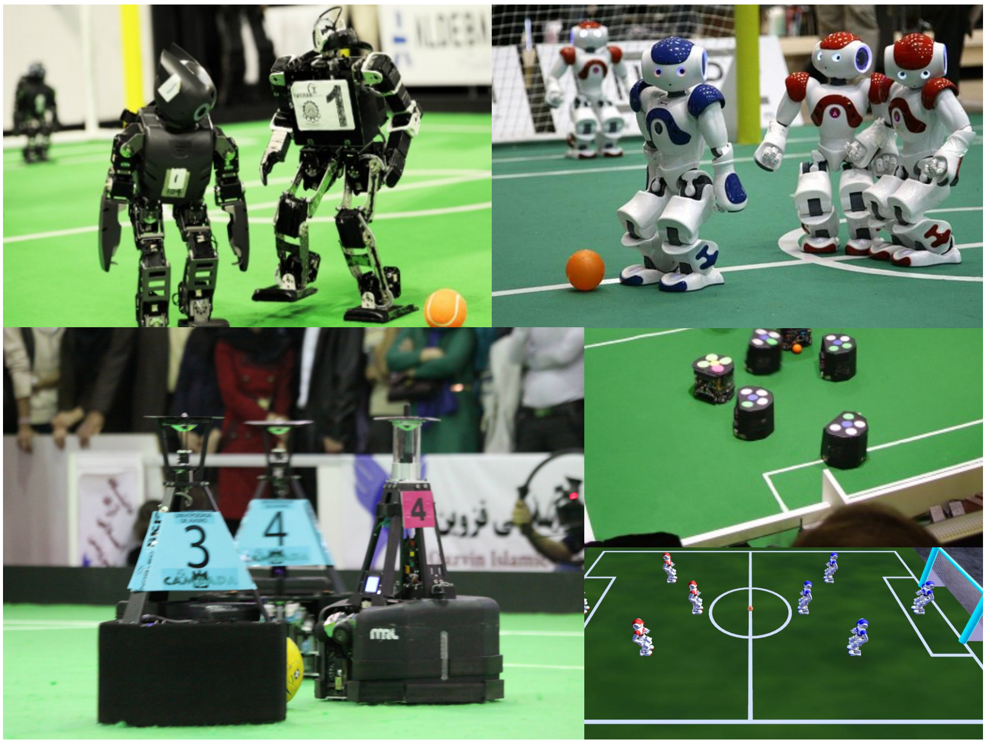


Figure 2.1: Picture of the different RoboCup Soccer Leagues.

- Soccer Humanoid League - In this league, autonomous robots with a human-like body and human-like sensors play soccer against each other. Since the main focus is the human sensorial capacity, the robots/agents need to be the as identical as possible to humans characteristics. Dynamic movements, kicking the ball when maintaining the balance, visual perception of the game environment, self-localization, team play coordination and all inherent technical aspects are the most important research issues that need to be considered. This league is divided in three subleagues, according to robots sizes:
  - KidSize, 40-90 cm height

  - TeenSize, 80-140 cm height
  - AdultSize, 130-180 cm height

- Soccer Middle Size League - The main research focus is on full autonomy and cooperation and perception capacities. Agents/Robots of no more than 52 x 52 x 80 cm and with a maximum weight of 40 Kg play soccer against others in teams of up to 5 robots and matches are divided in 15-minute halves. The game must be played in a green colored field with white lines similar to a scaled human soccer field. The ball must fit the regular FIFA size. Each match is controlled by a referee who has full authority to enforce the laws of the game being helped by as assistant referee that operates an application known as the *referee box* that send signals such as start, stop, kickoff, freekick and so on to playing teams according to the main referee decisions. Robots must be fully autonomous and have all sensors on-board (mostly cameras). For cooperation purposes, robots can communicate with team-mates and the coach through a wireless connection. Robots must be able to recognize objects and localize themselves using sensor information, decide which action need to be taken and control the motors and actuators autonomously. An autonomous coach normally in an external computer that has no sensors is responsible to take decisions only using the information provided by the players. No external human intervention is allowed with exception to the situation in which a robot needs to be inserted or removed in/from the field.

- Soccer Simulation League - Being one of the oldest RoboCup soccer leagues, its main research issues are focussed on the artificial intelligence and team strategy. Autonomous software agents/players play soccer on a virtual field inside a computer. This league can be subdivided in two subleagues: 2D and 3D. Matches with 5-minute halves are played by two teams of eleven virtual players. The game takes place in a computer simulator that provides a realistic approach of soccer robot sensors and actions. Each agent communicates with the simulation server sending information and motion commands regarding the player it represents and also receiving back information about its state, including sensor observations of the surrounding environment.

- Soccer Small Size League - The aim of this league is the exploitation of intelligence multi-robot/agent cooperation and control capacity in highly dynamic environment with hybrid centralized/distributed system technology. Matches with 10-minutes halves are played by teams of 5 robots smaller than 18 cm in diameter. The field has an overhead camera system that provides feedback to an external computer about the relevant objects in the field (e.g., ball, own and opponent player locations) which are distinguished by color and colored markers on the top of the robots. the external computer processes games information and sends the resulting commands to the team robots by wireless communications.

- Soccer Standard Platform League - This league replaced the existing Four-Legged League (which was based on Sony's AIBO dog robots). Here, all teams use identical robots (Aldebaran's Nao humanoids) which allows them to be concentrated only on software development, while still using the state-of-art robots. The robots do not include omnidirectional vision, forcing decision-making to trade vision resources for self-localization and ball localization. They are also fully autonomous with no external human or computer control.

2.1.2 RoboCup Rescue

The aim of RoboCup Rescue League is to promote the development of science and engineering issues that can be useful in disaster rescue situations involving a very large number of heterogeneous agents in an hostile environment. Technical research and development is promoted in a socially significant domain involving multi-agent teamwork coordination, physical robotic agents for search and rescue, information infrastructures, personal digital assistants, a standard simulator and decision support systems, providing evaluation benchmarks for rescue strategies and robotic systems. Some common features can be noticed with the soccer game in some aspects such as dynamic environment, incomplete and noisy information. The most important aspects, not present in soccer, are logistics, heterogeneous agents, long-range planning and emergent collaboration between different teams.

The RoboCup Rescue englobes two leagues:

- Rescue Robot League - In this league of teams the increase of awareness of the challenges involved in search and rescue applications is promoted as it is the collaboration between researchers and the evaluation of robotic implementations and capabilities (mobility, sensory perception, planning, mapping and practical operator interfaces and assistive autonomous behaviors) while searching for simulated victims in unstructured environments.

- Rescue Simulation League - The main objectives of this league are the development of simulators that form the infrastructure of the system that simulates the real environment and the development of intelligent agents/robots with capacity to interact in a disaster scenario. In urban search and rescue applications (USAR) robot competitions, a generic urban disaster simulation environment is constructed on network computers and some heterogeneous intelligent agents that conducted search and rescue activities. In this league three competitions are considered: the agent competition, the infrastructure competition and the virtual robot competition.

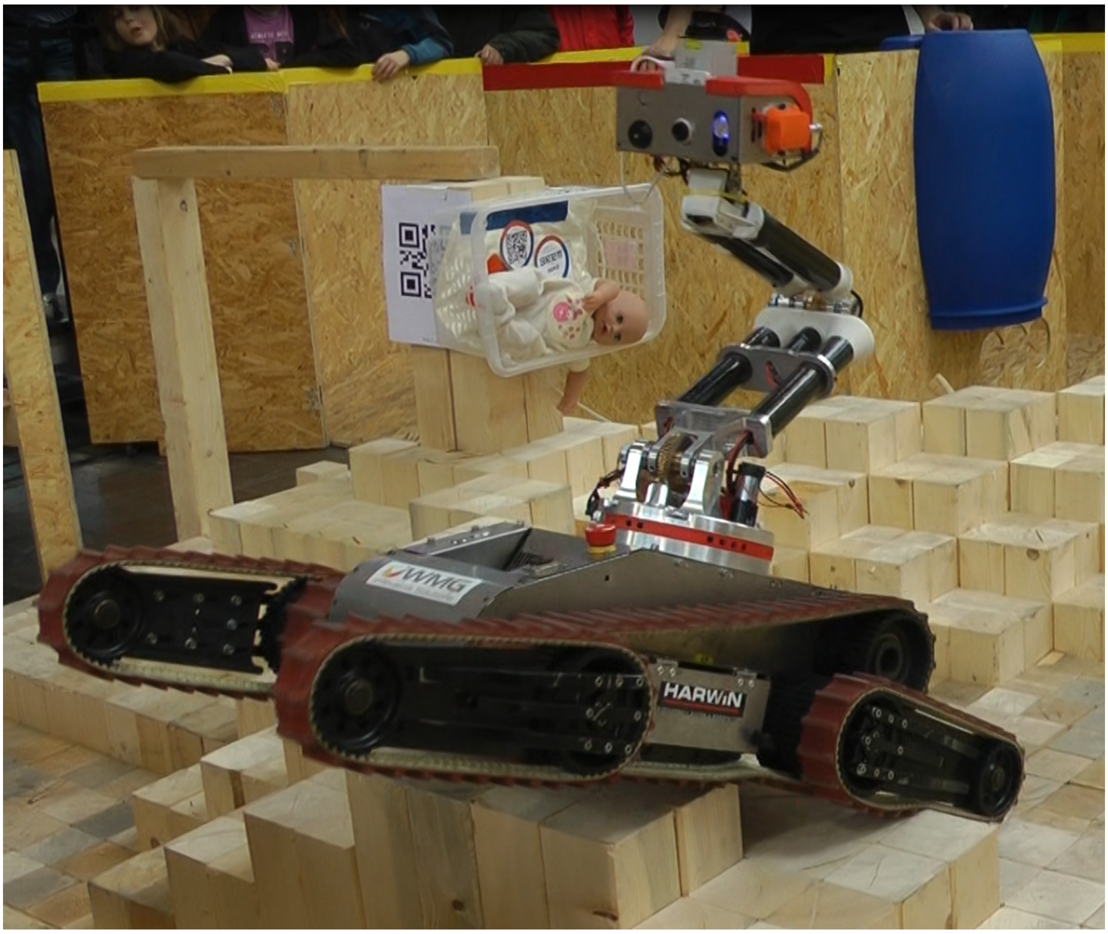


Figure 2.2: Rescue Robot from University of Warwick.

2.1.3 RoboCup@Home

This league started in 2006 and it is the largest international annual competition for autonomous service robots. The main issue is to promote the development of a service and assistive technology with high relevance for the real world itself, namely for personal domestic applications. The competition consists in a set of tests in which the robots have to use their abilities and performance to solve situations in a scenario close to a realistic home dynamic environment. Some domains are considered, such as: Human-Robot-Interaction and Cooperation, Navigation and Mapping, Computer Vision and Object Recognition under natural light conditions, Object Manipulation, Adaptive Behaviors, Behavior Integration, Ambient Intelligence, Standardization and System Integration.

Figure 2.3: AMIGO robot from Eindhoven University of Technology.

2.1.4 RoboCup@Work Demo

This league is a new competition where robots are used in work-related environments. Innovative mobile robots equipped with advanced manipulators for industrial applications are expected to cooperate with human in order to execute complex tasks such as: manufacturing, automation, and parts handling up to general logistics. In this competition, the specific targets proposed lead to several new specific challenges: mobile manipulation, logistics, cooperative mobile manipulation and multi agent planning, scheduling and multi-criteria optimization.

Figure 2.4: Kuka YouBot - The platform used in RoboCup@Work.

2.1.5 RoboCup Logistics League sponsored by Festo

The first official competition took place in 2012 with the aim of proposing a challenge for university students in a scenario close to industrial environment enabling scientific work in order to create a flexible solution of material and informational flow within industrial production using self-organising robots. Nowadays, the increasing in worldwide transport leads to a necessity of improved autonomous solutions. The evolution in Automated Guided Vehicles area requires new technologies to overcome present non feasibilities. The main objective of this competition is to use a hardware-in-the-loop simulation method to create an efficient material flow to provide a high rate of product deliveries in due time. The participant teams are composed by three robots (robot platform Robotino) with a 40 cm diameter with no limitations in terms of sensors and way of robots software programming. Robots need to be capable to solve the logistic challenges of an unknown production system. In a big production area of 12 m x 6 m, two teams have to compete for the most efficient solution resulting in the highest score. Problems such as out-of-order machines, express goods, changing delivery gates and a random machine distribution need to be solved by autonomous solutions resulting from research work in Mechatronics, Computer Science and Logistics areas.

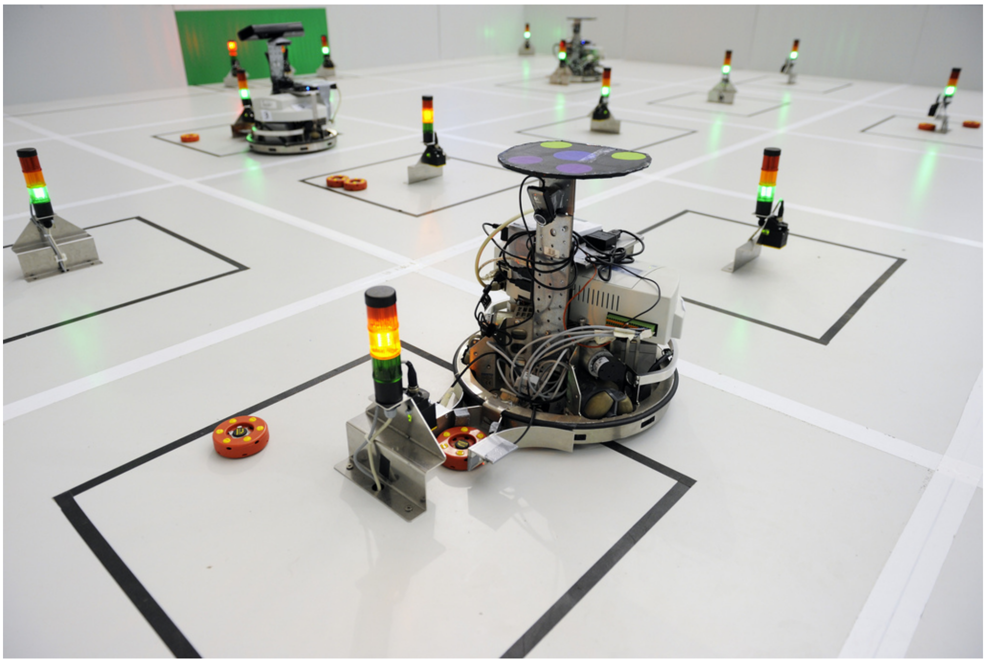


Figure 2.5: The RoboCup Logistics challenge.

2.1.6 RoboCup Junior

RoboCup Junior is an educational initiative for young students from primary to secondary school as well as for undergraduates who do not have resources to get involved in the senior leagues of RoboCup. This competition offers several challenges that emphasize the cooperation, problem-solving and task-achievement capacities. The Junior League has three different competitions:

- Soccer Challenge - Here, teams of autonomous mobile robots play games in a highly dynamic environment, tracking a special light-emitting ball in an enclosed landmarked field;

- Rescue Challenge - Robots try to identify victims quickly within re-created disaster scenarios;

- Dance challenge - One or more robots together with music, dressed in costume and moving in creative harmony.

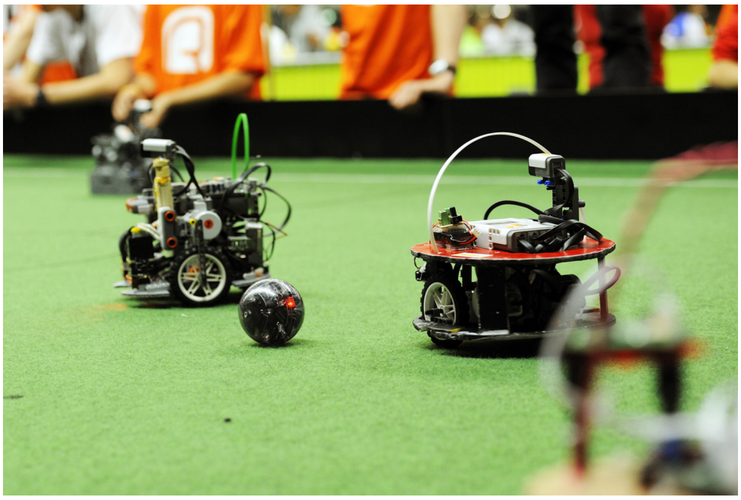


Figure 2.6: RoboCup Junior Soccer.

2.2 Summary of Middle Size League Rules

RoboCup Middle size robot league is based in the official FIFA laws. Some changes were made in order to enable the game to be played by Middle Size robots[2]. The most important rules will be briefly described:

- The Field - The field of play is rectangular, green and marked with white lines. It is 18 meters longer and 12 meters wide. The goals are 2 meters wide. Field design and markings are detailed in Figure 2.7.

- The Ball - The official tournament ball used is a FIFA standard size 5 soccer ball and cannot be mostly black, white or green.

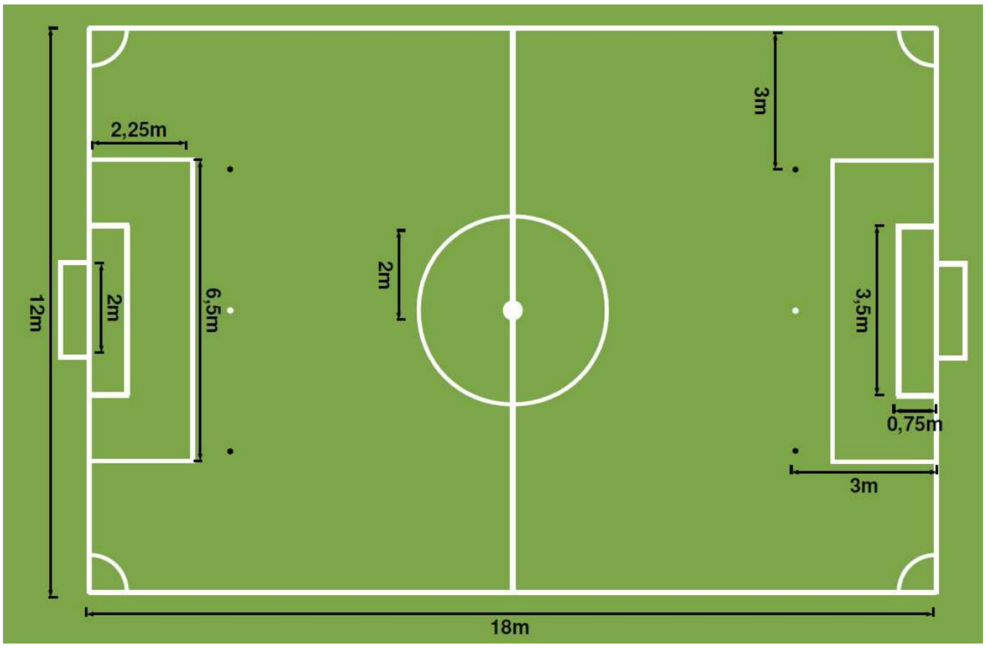


Figure 2.7: Official MSL field markings from [2].

- The Number of Players - The match is played by two teams, each consisting in no more than 5 players, one of whom is the goalkeeper. Currently substitutions are unlimited. The goalie can be exchanged with a field player during a break of the game, when goalie needs to be repaired. RoboCup players must be fully autonomous and human interference with robots is forbidden, except for substitutes and robots outside of the field of play. High level human coaching is allowed and each team may use a set of paper boards with QR codes to be directly interpreted by the robots. Only the robots that are in the field can be coached by the member designated by each team for this. Coaching is only allowed from the team leader position, in front of the teams' base station pc, and the robot(s) that is/are being coached should remain within the field and cannot be touched by human team members. Also the human that is coaching should stay always outside the of the field. No electronic devices, other than the devices already incorporated on the robot itself, can be used to transfer coaching instructions and the coaching action can only take place during the "dead time" (the 10 seconds between a stop and a start by the assistant referee).

- The Players' Equipment - Robots must be designed such they are both robust and safe, i.e., they cannot damage other robots or any objects in the field, or pose a threat to the audience, to the referees or to human team members. Also, the robots must be designed and programmed such that they try to avoid interferences concerning the operation of sensor systems and/or communication devices. Robots must be constructed so that their shape onto the floor fits into a square of size of at least 30 cm x 30 cm and at most 52 x 52cm. Only the goalie is allowed to increase his size instantaneously (at most 1 second) up to 60 x 60 cm width or 90 cm height if the goal area is endangered by an approaching ball. The robots height must be between 40 cm and 80 cm (exception for the goalie: 90 cm during extended phase). The maximum weight of a robot is 40 kg. The base color of robot's body must be black (matte black) in order to minimize reflectivity. Robots needs also to possess marks to be recognized by other robots and be distinguished by the referee. To be eligible to play, robots need to carry color markers (light blue for team A and magenta/purple to team B), number markers and top markers. Communication and information transfer between the robots of a team and between the robots and the Base Station must be done by wireless links. These communications must be established through one of the two Access Points available at the field of game and provided by the organization. the use of ad hoc wireless networking is strictly forbidden.

- The Referee - Each match is controlled by a referee in co-operation with the assistant referees, who has full authority to enforce the laws of the game. The referees are responsible for verifying the eligibility of all the equipment and resources needed for the game. Also, one of the assistant referees acts as a timekeeper and keeps a record of the match. Assisting technology (Referee Box) is used to supports the referee in particular for conveying referee decisions to robots and for the maintenance a record of the game. In case of a robot dangerous or incorrect behaviour, a single human team-member intervention to stop the robot is allowed evenly without the referee's permission. In this situation, the game is stopped by the referee and resumed with a free-kick for the other team taken from the position where the ball was when the robot was stopped or from one of the closest restart points.

- The Duration of the Match - The match lasts two equal periods of 15 minutes. Other periods of play can be defined for friendly games. In case of official tournament, any modification in the periods of play must be specified by the organizing committee. The half-time interval must not exceed 10 minutes. If this interval is exceeded by clear responsibility of one of the teams, the referee may limit the game to the first half time or in case of agreement with both team leaders, the referee may reduce the second half overall time. The allowance for time lost is of entirely discretion of the referee. If a penalty kick has to be taken or retaken, the duration of either half is extended until the penalty kick is completed.

- The Start and Restart of Play - The match must start at the scheduled time, but in case of exceptional situations the referee may re-adjust the time for starting the game in according with both team leaders. Robots start and stop procedure are done by receiving a signal through wireless communication from the outside of field. A kick-off is a way of starting or restarting play in the following situations: at the start of the match, after a goal has been scored and at the start of the second half of the match. To kick-off, it is necessary to follow some rules such as: all players are in their own half of the field, with the exception of the robot taking the kick, which may be partially inside the opponent half of the field; the opponents must remain at least 3 meters away from the ball until the ball is in play; the robot of the attacking team that is taking the kick is positioned at the ball; the players of the team taking the kick-off other than the kicking robot must remain at least 2 meters from the ball until the ball is in play; no robot, except the kicking robot, is allowed to touch the ball until the ball is in play; the ball is stationary on the center mark; the referee gives a signal; a player of the team who was awarded the kick-off kicks the ball or the player can kick the ball into its own half of the field; the robot taking the kick should either use its kicker or one of its sides to instantaneously kick the ball such that it travels freely over a distance of at least 0.5 meters; the ball is in play immediately after being kicked; after the kick, the attacking team is only allowed to touch the ball a second time after it moved over a distance of at least 0.5 meters; a goal may be scored only when the ball was touched by another player of the same team; when 10 seconds have passed since the start signal and the ball wasn't kicked by the attacking team, the defending team can approach the ball and score a goal directly, even without any contact between the ball and any other player (however, even after these 10 seconds, the attacking team can only score a valid goal after the ball has been touched by at least two of its players); if a robot of the attacking team except the kicking robot approaches the ball before the ball is in play, the kick-off will be awarded to the the other team. The above mentioned 2 and 3 meters refers to the radius of a circle centered on the ball. The robots must be completely out of each circle respectively, depending on its status (attacking or defending). The referee must restart the game within 10 seconds after the game stops. If the ball is kicked by the team that has kick-off and enters the goal without being touched by a second player of the same team before crossing the goal line, the goal is not scored and the kick-off is awarded to the opposing team. After 10 seconds of the signal and if the attacking team didn't touch the ball, a goal may be scored directly by the defending team. In a case of dropped ball, the referee gives a "stop" signal and all players stop their movement. The ball is positioned in the place where it was located when the game was stopped and the referee gives a "dropped ball" signal. All players remain 1 meter away from the ball and one robot may stay anywhere inside the penalty area (except goal area) of its own team, even if the distance to the ball is shorter than 1 meter. Now, the referee gives a "start" signal and the ball is in play immediately. A goal may not be scored directly from a dropped ball, the ball needs to be touched by at least two robots (not necessarily of the same team).

- The Ball In and Out of Play - A special "dead call" signal may be given by the referee, upon which all robots immediately have to cease operating any kind of actuator. After a dead call, the game continues with a dropped ball at the position nearest to the ball location when the game was interrupted, except when the referee issued a different call prior to the dead call. The ball is in play at all other times.

- The Method of Scoring - A goal is scored when the whole of the ball passes over the goal line, between the goalposts and under the crossbar, provided that no infringement of the laws of the game has been committed previously by the team scoring the goal. Any goal scored by a robot in the opponent goal will only be valid if the robot taking the kick is inside the opponent side of the field. This is not applied if the robot kicks into the goal of its own team. A goal is not validated if the kick is taken from the robot own half of the field and touch another robot of the same team before entering the goal and this goal is still not valid if the touched robot is in the opponent half of the field. Even if the ball is regained by one of the teams within its own half of the field a valid goal can only be scored after the ball has been received or touched by a teammate within the opponent side of the field and assuming that the ball has rolled freely for at least one meter before being received or touched. If the ball is regained in the opponent side of the field during play on, the robot regaining the ball may score directly as long as the kick is taken from within the opponent side. The winning team is that which scores the greater number of goals during a match. If both teams score an equal number of goals, or if no goals are scored, the match is tied.

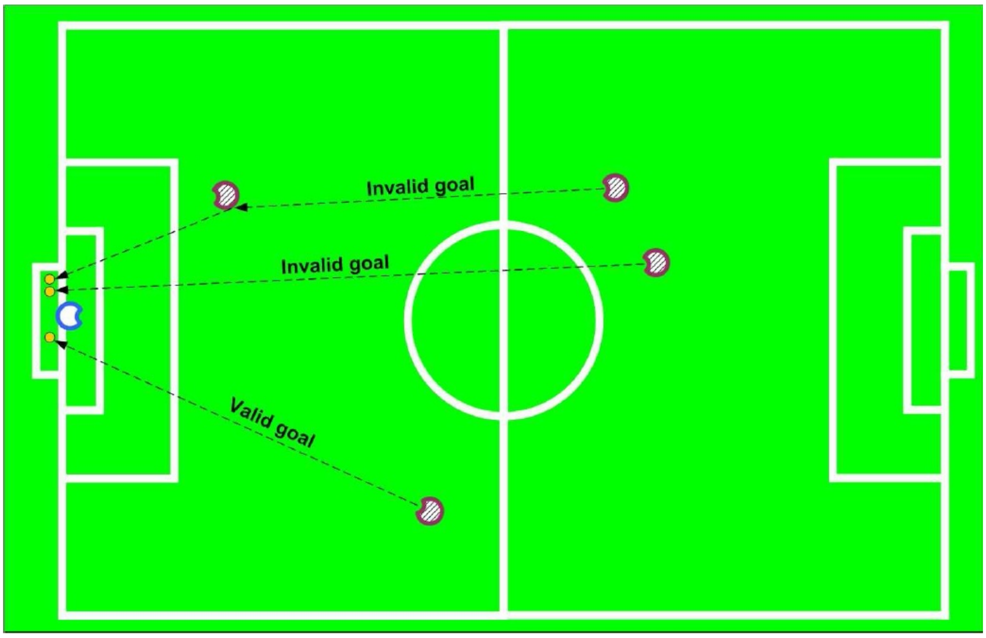


Figure 2.8: Valid methods of scoring from [2].

- Permissible Actions for Robot Soccer Players - The ball must not enter the convex hull of a robot more than a third of its diameter except when the robot is stopping the ball and in this situation the ball must not enter more than a half of its diameter. The force exerted onto the ball can only be done by direct physical contact between robot and ball. Forces exerted onto the ball that hinder the ball from rotating in its direction of rotation (natural direction of rotation) are allowed for no more than 1 second and a maximum distance of movement of 50 centimeters. Repeat this kind of forces is allowed only either after a waiting time of at least 4 seconds or if the robot has previously released the ball. Movements of the ball such as "roll-stop-roll-stop" are not considered a valid ball rotation and will be considered ball holding. For any kind of ball dribbling, direct contact between the robot and the ball can only be maintained within a circle with a radius of three meters, centered on the point where the robot last caught the ball. To move outside that circle, the robot needs to completely release the ball and this action need to be directly observable by any of the referees. Dribbling with direct contact between the robot and the ball outside this circle will be considered ball holding. Dribbling the ball backwards is allowed for a maximum of 2 meters and the ball must be rolling in its natural condition. If a robot dribbles the ball backwards for more than one meter, it can not repeat the same backward dribbling again before the ball has been completely released by this robot or until the robot has engaged a new struggle against an opponent robot. Disrespect any of the above rules is considered ball holding.

- The Free Kick, Throw-In, Goal Kick and Corner Kick - An indirect free kick is awarded in all situations where the FIFA laws specify a direct free kick. The kicker robot may touch the ball more than one time as long as the ball has not moved over a distance of more than 0.20 meters after an indirect free kick. After that, the ball must be touched by another player before the kicking robot can touch the ball again. A goal may be scored only after the ball has been touched by another player of the same team. The indirect free kick will be started from where the ball was when the offences occurred, if the ball was not inside a penalty area, and from the closest restart point if the ball was inside a penalty area. All other players of the free-kick awarded team can stay anywhere on the field except in a circle with a radius of 2 meters around the ball until the ball is in play. All players of the opponent team can stay anywhere on the field except in a circle with a radius of 3 meters around the ball until the ball is in play. One robot may stay anywhere inside the penalty area (except goal area) of its own team, even if the distance to the ball is shorter than 3 meters. The robot taking the kick should either use its kicker or one of its sides to instantaneously kick (i.e., without dribbling or dragging) the ball such that it travels freely over a distance of at least 0.5 meters. The ball is in play immediately after being kicked. When 10 seconds have passed after the start signal and if the ball was not kicked by the attacking team, the defending team can approach the ball and score a goal directly (if the ball is inside the opponent half field), even without any contact between the ball and any other player. However, even after these 10 seconds, the attacking team can only score a valid goal after the ball has been touched by at least two of its players.

- The Penalty Kick - This type of kick is currently not awarded during the two periods of play in a match. The execution of penalty kicks can be defined to decide the winner of game which ends in a draw. The penalty starts 5 minutes after the end of the game.

2.3 CAMBADA

CAMBADA is the RoboCup Middle Size League (MSL) soccer team of the University of Aveiro, Portugal. The team development started in 2003 and since then a significant progress has been noticed. During the last years, CAMBADA achieved the following important results and awards in national and international competitions:

- Portuguese Robotics Open 2014: 2nd place.
- RoboCup IranOpen 2014: 2nd place.
- RoboCup World Championship 2013:
  - Tournament: 3rd place.
  - Technical Challenge 1: 1st place.
  - Scientific Challenge 2: 3rd place.
- Portuguese Robotics Open 2013: 2nd place.
- RoboCup World Championship 2012:
  - Tournament: 4th place.
  - Technical Challenge 1: 1st place.
  - Scientific Challenge 2: 2nd place.
- DutchOpen 2012: 3rd place.
- Portuguese Robotics Open 2012: 1st place.
- RoboCup World Championship 2011:
  - Tournament: 3rd place.

  - Free Challenge 1: 1^{st} place.
- Portuguese Robotics Open 2011: 1^{st} place.
- RoboCup World Championship 2010:
  - Tournament: 3^{rd} place.
  - Technical Challenge 1: 4^{th} place.
- GermanOpen 2010: 2^{nd} place.
- Portuguese Robotics Open 2010: 1^{st} place.
- RoboCup World Championship 2009:
  - Tournament: 3^{rd} place.
  - Technical Challenge 1: 1^{st} place.
  - Technical Challenge 2: 5^{th} place.
- Portuguese Robotics Open 2009: 1^{st} place.
- RoboCup World Championship 2008:
  - Tournament: 1^{st} place.
  - Technical Challenge 1: 2^{nd} place.
  - Technical Challenge 2: 5^{th} place.
- Portuguese Robotics Open 2008: 1^{st} place.
- RoboCup World Championship 2007:
  - Tournament: 5^{th} place.
- Portuguese Robotics Open 2007: 1^{st} place.

The CAMBADA robots (Figure 2.9) were designed and entirely built in-house. A triangular aluminium chassis supports three independent DC motors (allowing for omni-directional motion), an electromagnetic kicking device, three 4 cells LiPo batteries and the motor controller modules. The other parts of the robot are located in three higher layers. The layer immediately upon the chassis is used to place the kicking mechanism, the ball handler and related electronic items. In other layer is located the PC and the top layer contains an omnidirectional vision system based on an hyperbolic mirror (AIS Fraunhofer-Gesellschaft) and the IMU module [22].

Figure 2.9: CAMBADA robot.

The robots architecture is based on a main processing unit (a PC with the Linux operating system) that is responsible for the higher-level behavior coordination (the coordination layer). This unit controls the processing of visual information acquired by the vision system, the execution of high-level control functions and the handling of external communication with the other robots. This unit is also responsible for receiving sensing information and sending actuating commands to control the robot attitude in the context of a distributed low-level sensing/actuating system.

The low level control layer is connected to the coordination layer through a CAN-USB gateway responsible for filtering interactions that could occur within both layers, passing only the information that is relevant across the layers. The low level sensing/actuating system yield the following main functions:

- Motion control - This block englobes three independent motor control boards (one for each DC motor) each of them receiving a velocity setpoint from the high-level holonomic motion controller.

- Odometry computation - This module combines the encoder readings from the three motors and provides coherent robot displacement information which is periodically sent to the high level coordination layer;

- Kicking control - This function is responsible for controlling an electromagnetic kicker with fifty levels of kicking strength. This kicker allows the robot to shoot a regular size 5 FIFA ball with speeds up to 11 m/s permitting the choice of the shoot type between lob shoot and straight shoot.

- Ball handling - This module controls the ball grabbing during dribbling actions trying to keep the ball close to the robot while avoiding ball holding situations. It is also responsible to gather the ball during ball pass actions.

- System monitoring - Being a distributed function, it is responsible for the monitoring of the robot batteries and the state of all nodes in the low-level layer.

- Inertial measurement unit (IMU) - An item that consists in a 9 DOF module with a triple-axis gyro, a triple-axis accelerometer and a triple-axis magnetometer useful in the robot heading estimation which is later on joined with the heading provided by the vision equipment.

The communication between team robots is performed using an adaptive TDMA transmission control protocol on top of IEEE 802.11x, that reduce the risk of transmission collisions between team mates reducing the communication latency [23].

The vision apparatus (Figure 2.10) comprises an omni-directional setup based on a catadioptric configuration implemented with a gigabit ethernet camera and a hyperbolic mirror. Radial search lines are used by the image processing software in order to analyze the color information [24]. Some image zones such as the robot itself, the bars that hold the mirror and the areas outside the mirror need to be excluded and are ignored using an image mask that was previously generated.

Figure 2.10: The vision catadioptric set.

An analytical method developed by the team combining the back-propagation ray-tracing approach and the mathematical properties of the mirror surface allows the establishment of a relationship between image pixels and the real world distances [25].

Sensor fusion, basic behaviors and high level decision and cooperation are the high-level decision construction basis. Sensor fusion collects the noisy information from the sensors and from the teammates and update the world state database used then by the high-level decision and coordination. Basic behaviors module defines the set of primitives used to control the robot. The high-level decision module analyses the situations and performs the decision-making processes that need to be done by each player in order to maximize the global performance and success of the team.

In Figure 2.11 the robot software architecture components and their relationship are illustrated.

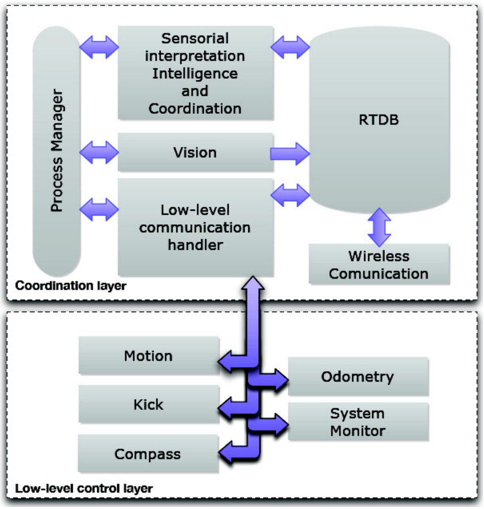


Figure 2.11: Schematic representation of the main modules in the architecture of the robots [3].

Chapter 3

Utility Maps

In order to merge all the relevant information about the environment (teammates, obstacles or ball position), conditions, restrictions and used metrics was chosen to use height maps. This approach allow easy decision just by analyzing the values of the maps.

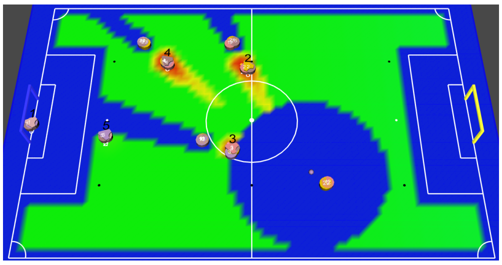
The information about the obstacles in the current version of the CAMBADA software is a list of objects containing its position on the field and their classification of being teammates or opponents. The algorithm for obstacles detection and identification is described in [26].

In this chapter are explained the concepts of Height Maps and Field of Vision, and their usefulness for the developed work. It is also presented the chosen library.

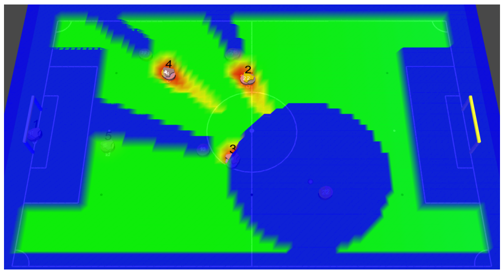
3.1 Height Maps

In this work, height maps are used in the sense of being utility maps that show the importance level of positions in the field according to the game situation in a certain moment. This type of maps assign a utility/cost value to each region of the field and can be represented with a color gradient (Figure 3.1a) or even a height gradient (in case of 3D map representation, Figure 3.1b) which allows a rapidly visualization of the relative importance of a certain position. To each possible position a value is attributed according to its importance and the map is built with it.

On the maps visualization the values are color code from blue to red, where red represents a highest utility value and blue the lowest.



(a) 2D visualization of a heightmap.



(b) 3D visualization of a heightmap.

Figure 3.1: Heightmap vizualization examples.

3.2 Field of Vision

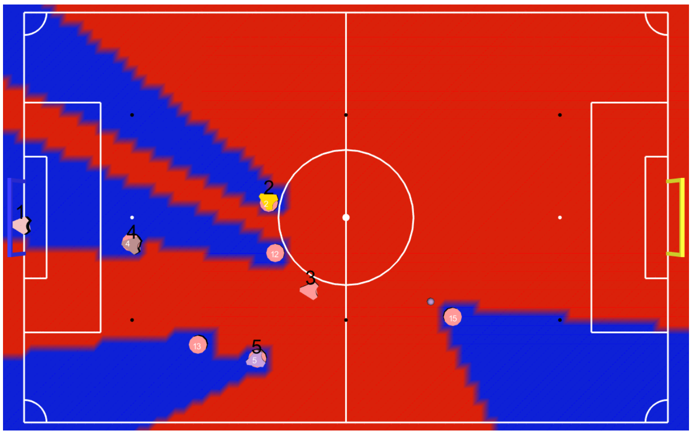
Some of the most important features on the game require that all the players have an adequate and precise visualization of the other robots, the ball and all the objects positions in the field of play. The Field of Vision (FOV) can be referred as the area that is considered visible from the point which it was calculated. In this work, the FOV is useful for instance, in the following situations:

- The FOV calculated considering the ball position rapidly allows the robot to know for which zones can the ball be passed without being intercepted by any obstacle (Figure 3.2a).
- The FOV defined taking into account the robot position gives information about which zones the robot can move on without the need to make any deviation from obstacles (Figure 3.2b).
- Joining the two above referred ways of FOV determination can be useful in situations of forward passes (Figure 3.2c).

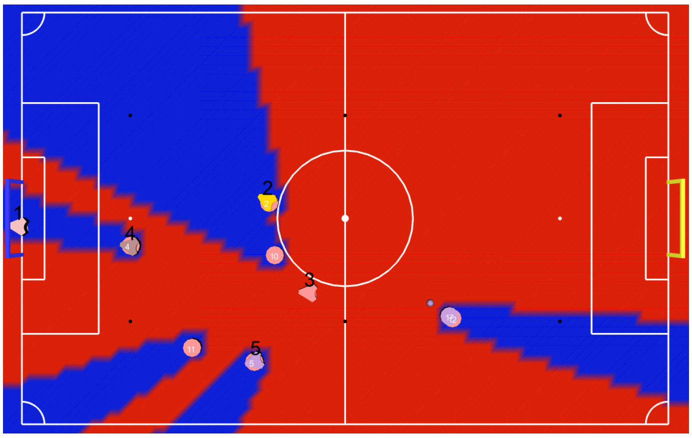
3.3 Library

After doing some research on field of vision libraries we came upon the litbtcod[1]. This library which, while not dedicated to fov, contains toolkits for field of vision calculation. Another reason for choosing the library was the simplicity of use and good documentation. As an extra it also have toolkits for management of height maps.

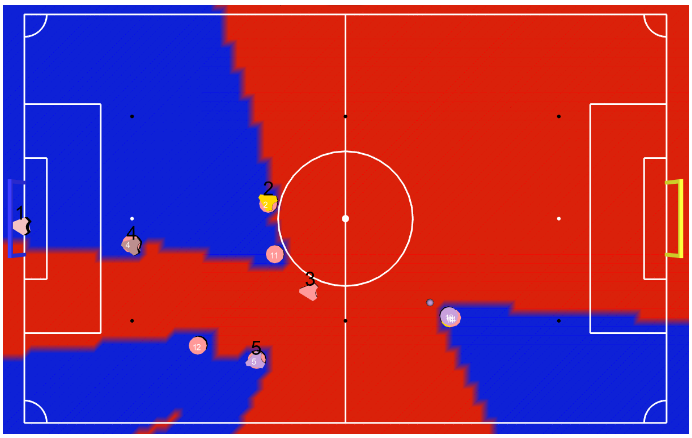
[1]http://doryen.eptalys.net/libtcod



(a) FOV calculated from the ball.



(b) FOV calculated from the robot 3.



(c) FOV calculated from the ball merged with FOV calculated from the robot 3.

Figure 3.2: Examples of FOV.

Chapter 4

Defensive Set Pieces

In defensive set pieces only one role is involved, role Barrier. The robot will stay in this role until the ball is moved in the set piece or the time limit ends (10 seconds), whichever comes first. After this point the robot assume one of two roles, role Striker or role Midfielder, depending on the robot position regarding the ball.

4.1 Role Barrier

Inside role barrier the agent can assume one of the following behaviors based on priorities and conditions, being the first the most priority and so on:

- If a stop signal is sent
  - Behavior - BStopRobotGS
    To comply with the rules, when a stop signal is sent, every robot must stop, so this behavior has the highest priority in every role. The robots cannot move until another signal is sent.
- If it do not know where is the ball and is the robot with the lowest id in this role
  - Behavior - BSearchBallBarrier
    If the ball has not yet been seen since the robot entered this role, the robot go search for it.
- If the gameState is one of the defensive set pieces

– Behavior - BBarrier

  In this behavior, the robot goes to the position indicated by the coach, if it is running, or to the position from the strategy, calculated locally. The position indicated by coach could be a position from the strategy (described below in section 4.2) or a cover position (detailed in section 4.3). In the second case, the objective is to prevent the opponent to have a clear line to pass the ball. While on a cover position, in order to improve responsiveness to the opponent movement, the position indicated by the coach is only used as an orientation for the agent (the coach and agents information is only shared every 100 ms plus network delay). The agent will always try to correct the position in order to stay in the line between the ball and the opponent that he is trying to cover.

4.2 Base Positions

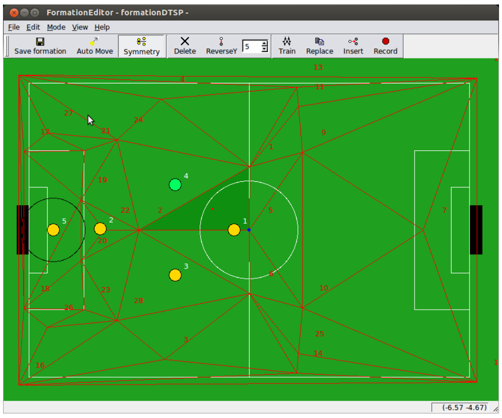
To calculate the base positions for defensive set pieces we use a tool that uses Delaunay Triangulations to interpolate all the possible situations. On top of that are applied the restrictions from the rules (minimum 3m to the ball, except in drop ball that is only 1m; only one player inside our penalty that don't need to respect the previous rule as long it is inside the penalty area).

Delaunay Triangulation is a method that allows to triangulate a plane region based on a given point set. Delaunay triangulation for a set P of points in the plane is a triangulation DT(P) such that no point in P is inside the circuncicle of any triangle in DT(P). If more than 3 points are given, we can get a unique Delaunay Triangulation.[27]

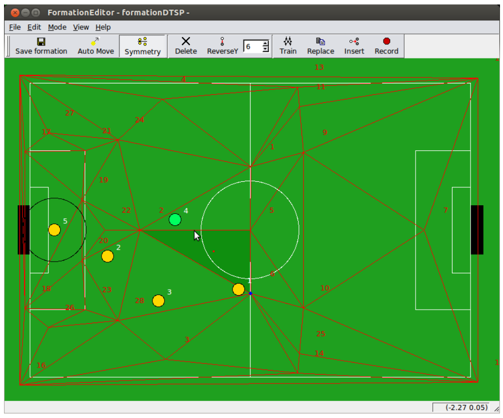
The Figure 4.1 shows the tool used to configure the positions set. Here, the ball positions are used as the vertices of triangles and each vertex means the given training data. Each vertex has the output value as an agent's move position for that vertex (=ball position). When the ball is contained by one triangle, agent's move position is calculated by interpolation algorithm described in [27].

4.3 Cover positions

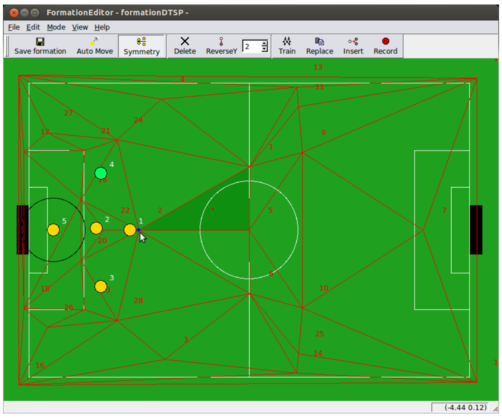
In order to prevent the opponent passes, our robots must be positioned between the ball and the possible Receivers. These cover positions are obtained from a height map.



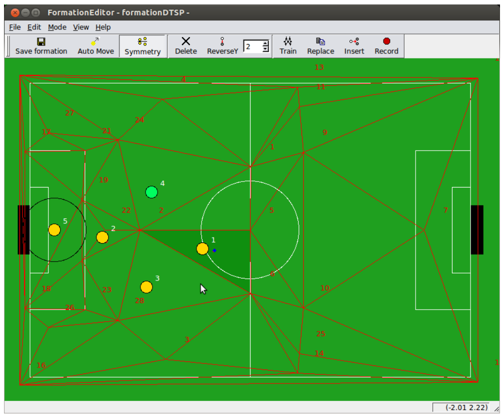
(a) Configured positions, Ball in center.



(b) Configured positions, ball in other vertex.



(c) Configured positions, ball in other vertex.



(d) Obtained positions with the Delaunay Triangulation.

Figure 4.1: Formation Editor, examples of configurations.

That map is calculated on the coach in order to prevent different decisions from the different agents. This way the calculation is done using the information shared by the field agents, and then the decision shared with the agents.

Sometimes exists error in the identification of obstacles. Then it is necessary to filter the information received from the different agents. Obstacles from the different agents close to each other are merged using a clustering algorithm, and obstacles too close to a teammate are ignored unless that teammate sees it. The filtered information is then used to build the map.

From each cluster of obstacles is carved a valley in direction to the ball. After that the calculated map is added with a previous calculated map, that defines the priorities of the positions (Figure 4.2), in order to prioritize the positions. Finally there are added the restrictions (minimum distance to the ball, inside of the field, penalty areas)(Figure 4.3).

The distance between the opponent that we are trying to cover and the robot doing the cover can be easily changed in the configuration file. In the configuration file can be also set the strategic position that are allowed to become cover positions. This means that the human coach can specify how many robots will be used for covering.

It is also possible to specify in the configuration file if a robot will start the set piece in front of the opponent or if it will do it after the start signal. This is useful to allow the opponent team reach the ball to score a set piece.

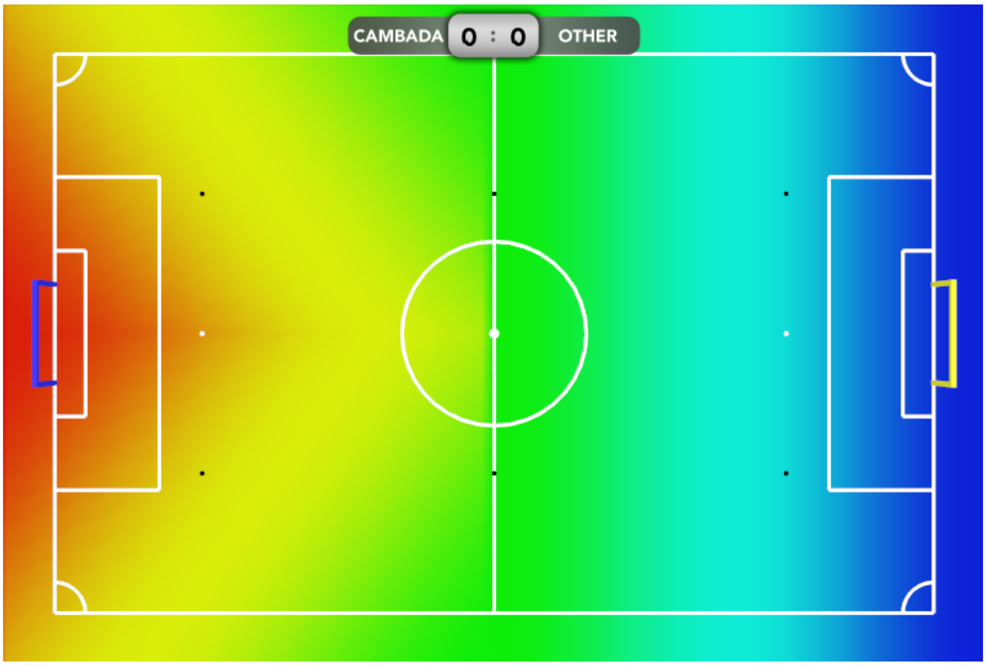


Figure 4.2: Cover priorities map.

4.4 Results

Since this is a new feature it can not be compared to what was there before. So as the results we have only the analysis of some of the Robótica2014 games.

Analyzing the Table 4.1, considering that when the attacking team made a short pass was forced into it by not having other pass options, we have a success of 88.4% in defensive set pieces situations.

Figure 4.3: Example of cover map.

Phase	Opponent team	<3m	Ball possession recovered	Pass not intercepted	% Sucess
Semi-final	Carpe Noctem	21	2	1	95.8
Final	Tech United	10	5	4	78.9
Total		31	7	5	88.4

Table 4.1: Defensive set pieces cover efficiency during second phase of Robotica2014.

If look deeply into the unsuccessful situation, the problem was not the cover but the transition from the barrier role into the Midfielder or Striker role.

Chapter 5

Offensive Set Pieces

In offensive set pieces two roles are involved, role Replacer and role Receiver. The robot closest to the ball will assume the role of Replacer and the others will assume the role of Receivers. After the ball is passed the robot that will receive the ball will became Striker and all the others Midfielders.

5.1 Role Replacer

The robot that assumes this role is the one responsible to perform the pass. Inside this role it can assume one of the following behaviors based on priorities and conditions, being the first the most priority and so on:

- If a stop signal is sent
  - Behavior - BStopRobotGS
    To comply with the rules, when a stop signal is sent, every robot must stop, so this behavior has the highest priority in every role. The robots cannot move until another signal is sent.
- If ball has been passed
  - Behavior BReplacerBallPassedStop
    Go back 1m and stop.
- If ball is not being seen by any of our robots

- Behavior BSearchBall
  Tour a predefined set of points, trying to find the ball.

- If the start signal has not yet been given
  - Behavior BReplacerPos
    Go to Set Play position, close to the ball.

- If the start signal has been given and point to pass has not yet been chosen.
  - Behavior BReplacerAlign
    Choose the point to pass the ball (detailed in Section 5.4)and align with the chosen point and the ball.

- If aligned with ball and point to pass, and ball has not been passed
  - Behavior BReplacerPass
    Go to the ball, engage it, re-align with point to pass. When aligned check if still have line clear to pass. If yes pass the ball, otherwise abort the pass.

- If none of the conditions for the above behaviors has met
  - Behavior BStop
    Makes the robot stop. This behavior is used as a default or fallback behavior.

5.2 Role Receiver

The robots who assume this role are the possible Receivers for the Replacer's pass.

- If a stop signal is sent
  - Behavior - BStopRobotGS
    To comply with the rules, when a stop signal is sent, every robot must stop, so this behavior has the highest priority in every role. The robots cannot move until another signal is sent.

- If the ball was passed to me and none of the teammates have the ball engaged

  - Behavior BMidfielderReceiveBall
    Go to the nearest point of passline to intercept the ball

- If Replacer still has the ball or ball not passed to me and not going in my direction

  - Behavior BReceiverPosition
    If the start signal has not yet been given, go to set play receiver position and calculate an alternative position (explained in Section 5.5) to receive the ball. If the start signal has already been given indicate Replacer which of the points to pass the ball, and if he chooses to pass the ball to me, go to the indicated point.

- If ball not engaged

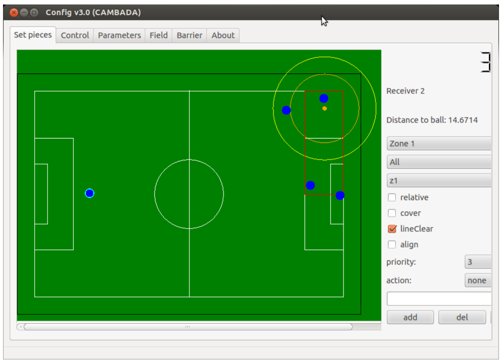
  - Behavior BGoToVisibleBall
    Goes directly to the ball to grab it.

5.3 Base Positions

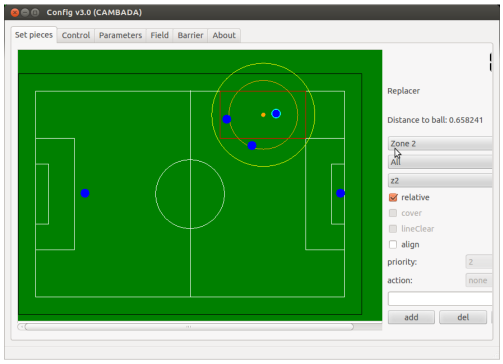
To configure the set pieces we use a tool (Figure 5.1), where the field is divided into 10 zones and each zone defines a set of positions for the Replacer, receivers and position to kick by default. The position of the receivers can be absolute or relative to the ball. We can also define if the receiver needs to have line of sight to be considered an option to pass or should align with the goal. We also indicate what priority the receiver presents and the action to be performed. This action can be a pass, a cross or none. In this last case the receiver will never be considered a pass option.

5.4 Replacer point to pass selection

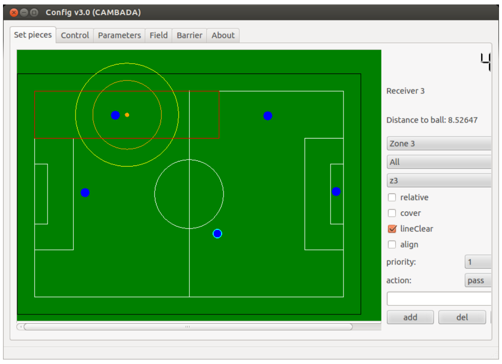
On behavior BReplacerAlign the receivers chooses the point to pass the ball. That decision is made through analyzing the positions shared by the receivers and the action defined in the configuration for that receiver. The positions are analyzed by priority until one of the receivers indicate that has a clear line to receive the ball in the indicated position, or until the set piece time reaches the 8 seconds, forcing the Replacer to shot the ball to the default point defined in the configuration.



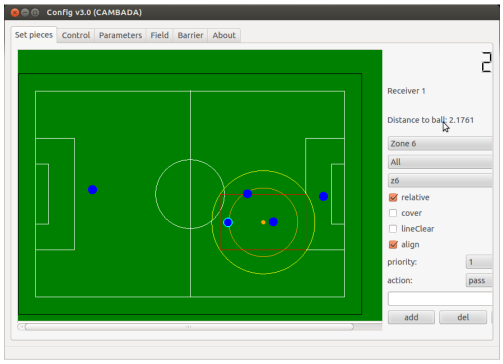
(a) Receiver 2, zone 1.



(b) Replacer, zone 2.



(c) Receiver 3, zone 3.



(d) Receiver 1, zone 6.

Figure 5.1: Set Pieces config tool, examples of configurations.

When a pass is aborted on behavior BReplacerPass the Replacer is forced the the previous behavior, because the point to pass chosen is no longer valid, in order to choose a new point to pass.

5.5 Receiver Alternative Position

When we configure the set pieces the opponent team is not taken in account. So after the opponent team positioning the configured position can be a position where the receiver can not receive a pass. To deal with this situations it is needed to have an alternative reception position calculated dynamically taking in account the opponent team. To calculate the alternative position for the receiver to receive the ball it is used an utility map.

In the construction of this map there are taken into account all the constraints of the field (goal area), minimum distance the ball (2m) and valid positions to score a goal. The field is divided into two zones for the application of different metrics, our side of the field, where the used metric is just the distance to the halfway line, and the opponent's side of the field, where three metrics are used, one is the free space to pass the ball, the second is the weighted average between the distance to the ball, on goal distance, rotation angle for a shot on target and distance from the starting point, and the third the dead angle to goal. The second is applied when the first is greater than one defined value (1m). All weights in the weighted average are easily configurable via the configuration file without having to recompile the code.

These metrics are only applied in a circle of radius also defined in the configuration file, centered on the position from set pieces' strategy, only to points that have FOV from the ball (positions where the ball can be passed) and from the center of the circle (positions where the receiver can move up quickly without having to avoid obstacles).

In Figure 5.2 all the receivers are announcing that have line clear to receive the ball. Robot 4, the Replacer already chosen to pass the ball to the robot 2. The pass it is trying to make is represented by the black line. Robot 2 will move to that point to receive the ball.

Figure 5.2: Example of alternative position map for receiver calculated for robot 2. CAMBADA is attacking to the blue goal. The black line goes from the ball to the alternative position indicated by robot 2 to receive the ball.

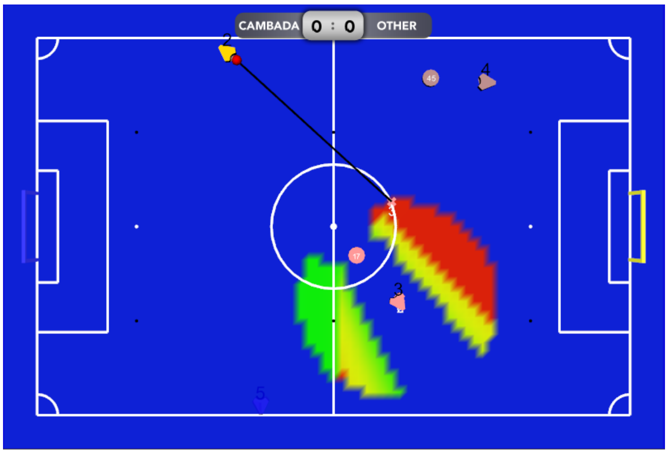
5.6 Results

In the last two games of Robotica2014 against Tech United (final) and Carpe Noctem (semi-final) there was a total of 44 offensive set pieces situation. An average of 22 offensive set pieces per game, in a game of 30 minutes, means a offensive set piece situation every 1 minute and 20 seconds.

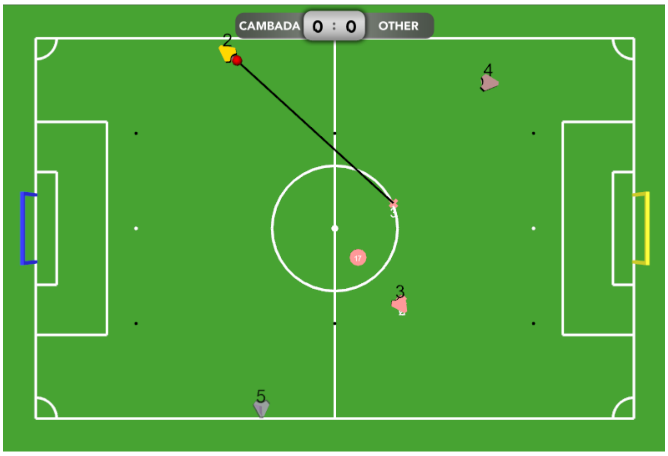
Figure 5.3 represents a typical set piece situation. First the receiver indicates the position where we can receive the ball, then the Replacer chooses a receiver from the available and indicates the intention to pass the ball to that robot, after that the receiver start moving to the indicated point in order to receive the ball and finally when the receiver is close to the indicated point the Replacer passes the ball.

As explained in Section 5.5, the choice of the receiver alternative position to receive the ball is easily configurable. The Figure 5.4 shows 3 possible alternative positions for the same set piece situation obtained just by changing the weights used on the calculation of the weighted average.

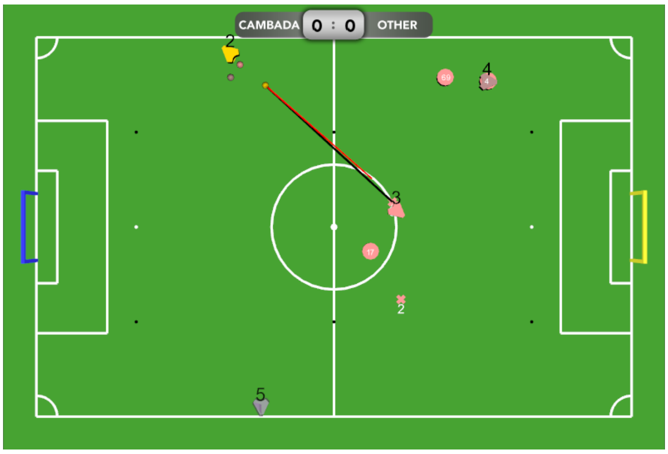
Since the development of this feature had no yet been fully completed by the date of the Robotica2014 competition, it was not possible to test it in real game situations. So set pieces situations were recreated in the lab in order to test the new approach for the offensive set pieces. Three robots played as the attacking team, while other 2 assumed cover positions. 20 situation were recreated and the Replacer never chosen to shot the ball to the default configured point. This means that in all situation at least one of the receivers could find a valid point to receive the ball and the Replacer passed the ball successfully to that point.



(a) Receiver Map with alternative position chosen.

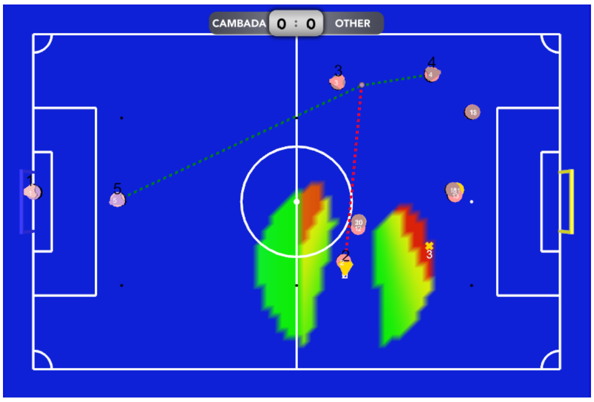


(b) Replacer indicating where is going to pass the ball and waiting fot the receiver to get there.

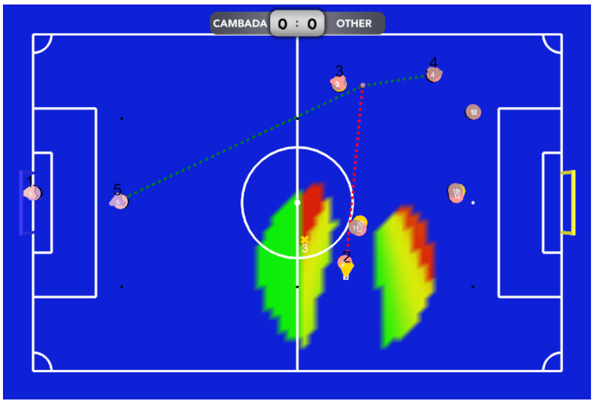


(c) Ball passed to the indicated point.

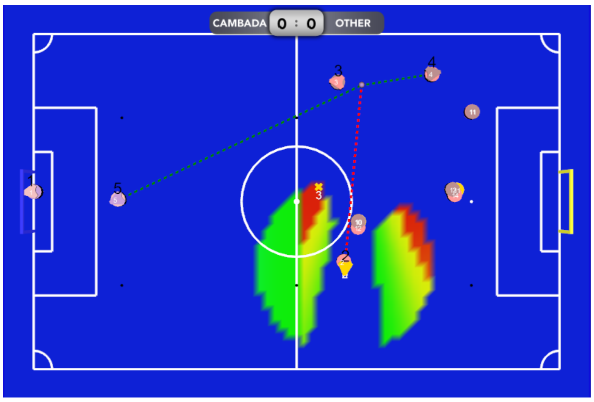
Figure 5.3: Offensive set piece situation.



(a) Receiver alternative position with more weight to the distance to goal.



(b) Receiver alternative position with more weight to the rotation angle to shot on target.



(c) Receiver alternative position with more weight to the distance to the ball.

Figure 5.4: Receiver alternative positions with different weights. The yellow cross with number 3 represents de alternative position.

Chapter 6

General Changes

In this chapter it is presented the changes made to the module responsible for team strategy and the improvements done to coach and basestation applications.

6.1 Strategy

Strategy is the class responsible for calculate the move positions, given the ball position, and assign them to specific robots. It is used in defensive set pieces situations and free play.

- Now we have only one instance of *strategy* for the two situations that uses the *strategy*, instead of two instances of *strategy*, one for each situation.
- Restriction to the positioning are now applied in *strategy* instead of being applied on the behaviors that uses them.
- Positions are only recalculated if the ball moved or the game state changed, instead of being calculated every cycle.
- New exchange algorithms (function responsible for assign the positions to specific robots).
  - Greedy (old) - The robot closest to the most priority position assumes it, the robot (with no attributed position yet) closest to the second most priority position assumes it, and so on. Used when the robots need to move as quickly as possible to the higher priority positions, for example free play.

- Global weighted distance minimization - Positions are attributed so that the global distance that the robots have to move is the minimum. The most priority positions have higher weights. Used when positions priorities only mean the positions that needs to be assigned prioritized, for example when we have less robots in the field than the number of defined positions, the less prioritary are left unassigned.
- Hybrid version of the two above - The n first positions are attributed with greedy algorithm, the rest are assigned with the global weighted distance minimization algorithm. Used for situations where some positions are cover positions and the robots need to get there as soon as possible. The non cover positions are treated in order to minimize the team positioning effort.

6.2 Coach

The coach software agent now is used as a debug tool too. As it shares the RTDB with basestation process, since both run on the base station computer, it can share the height maps to be visualized in the basestation window without congest the network (see Figure 6.1). So the coach is able to calculate any height map using the live information shared by the other agents or the information from the log files.

Some of the maps are calculated on the robots and none of them is shared the network. In order to view them on the basestation, the coach recalculate them using the information shared by the specific robot that we want to see the map.

6.3 BaseStation

BaseStation is the process used to visualize and control robots state remotely. It is also used to analyze the logs and the height maps (see Figure 6.2). The logs contain information from all the agents saved so it can be analyzed later.

Since height maps are now used for some of the decisions, it would be nice to save them too. But they are too large too be stored every 100ms in the log file. As they are calculated from the agents information they could be recalculated using the information saved in the log file.

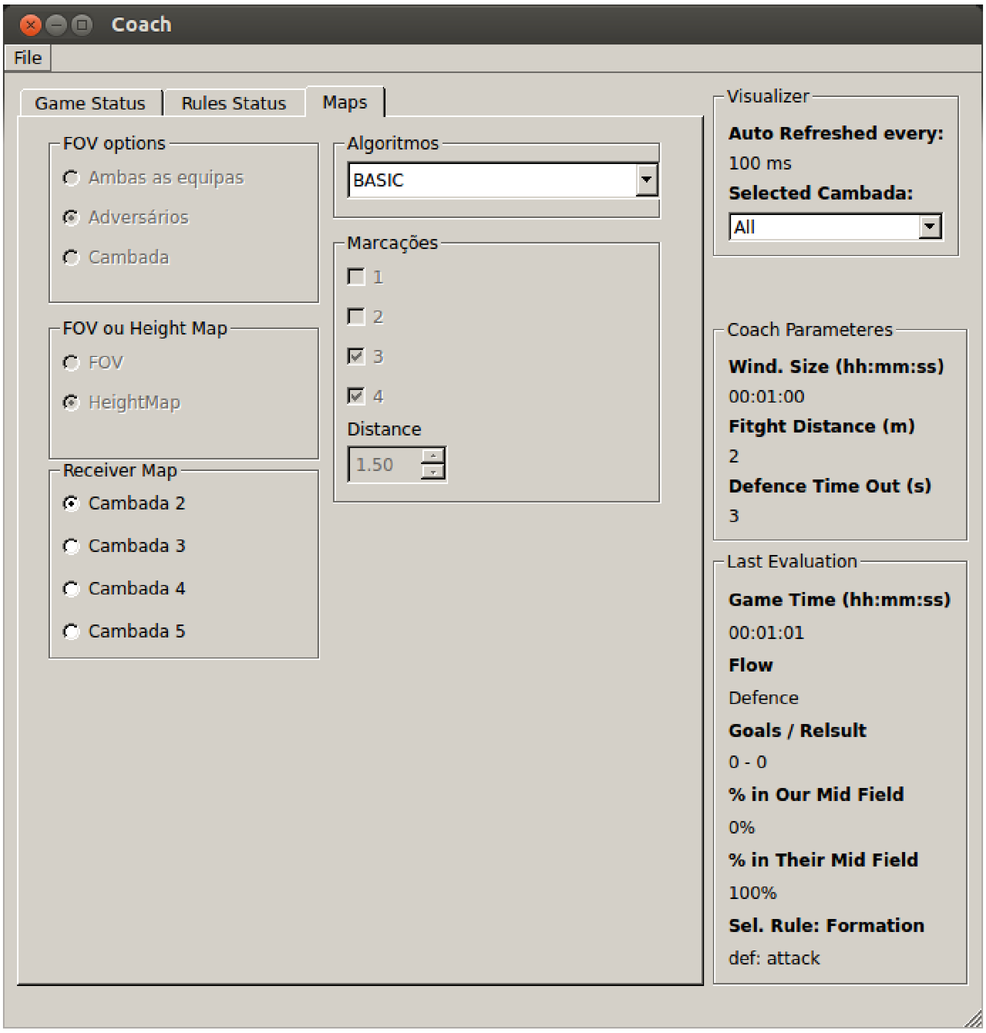


Figure 6.1: Coach interface used for debugging and testing.

So when a log is being analysed instead of only displaying the stored information, it is also shared with coach process so it can recalculate the height maps used for the situation being analyzed.

Thus debugging is much easier and faster than just analyzing the information without the maps used as its base.

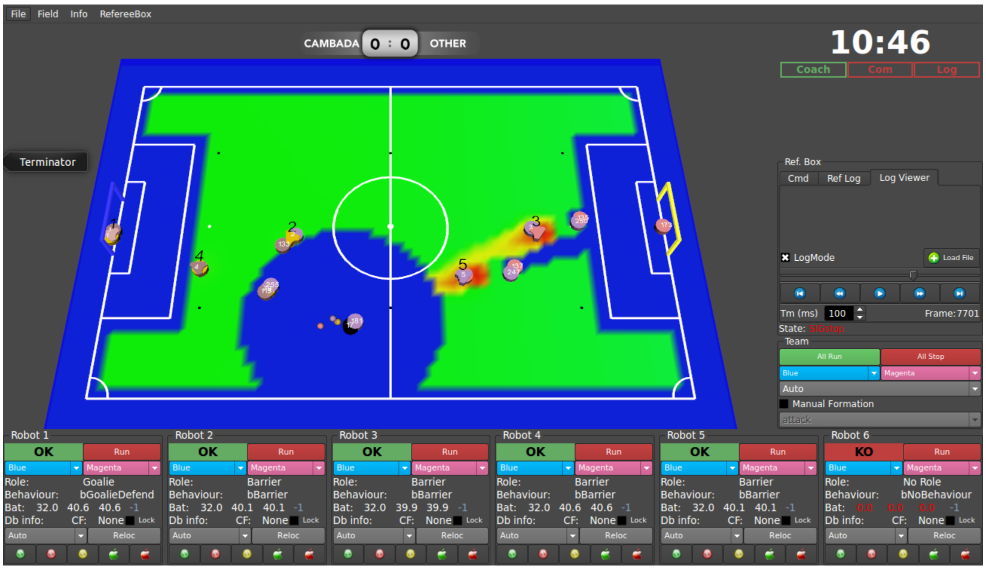


Figure 6.2: Basestation interface showing a map calculated with log data.

Chapter 7

Conclusions

This thesis' main objective was to improve the CAMBADA team performance in set pieces situation.

When this thesis started the set pieces strategy was defined prior to the game and minor adjustments were made concerning the other team attitude. Some dynamism had to be introduced in order to keep up with the other teams.

To achieve the required dynamism a new approach to defensive and offensive set pieces has been presented.

In the defensive set pieces the team have a more active positioning with the goal of preventing the successful passes of the opponent team. Now our robot while trying to block the pass lines of the opponent have the opportunity to recover the ball possession. With the configurable parameters our team attitude can be easily adapted to different opponent teams. This new approach to defensive set pieces led to the opposing teams to have more difficulty in advancing on the field, since in most of the situations they were forced to do a short pass.

In the other hand the offensive set pieces were also tweaked creating the possibility of more successful passes. The reworked choice of the receiver to pass the ball allows the passer (replacer) to have more time to make that decision.

The dynamic calculation of the alternative point to receiver the ball allows the team to perform more successful passes with the possibility of forward passes.

The modification on strategy class were made in order to solve instability problems at the time of assignment of the positions to a particular robot previously identified.

The improvements to coach and basestation were done in order to help in the debugging process.

In conclusion, all the work developed has been tested an included in the team master source-code branch and used in competitions.

Bibliography

[1] Lisa Jean Moya and Andreas Tolk. Towards a taxonomy of agents and multi-agent systems. In *Proceedings of the 2007 Spring Simulation Multiconference - Volume 2*, SpringSim '07, pages 11–18, San Diego, CA, USA, 2007. Society for Computer Simulation International.

[2] Middle size robot league rules and regulations for 2014. `http://wiki.robocup.org/images/d/d1/Msl-rules2014.pdf`. Accessed May 24, 2014.

[3] 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.

[4] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, November 2005.

[5] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2):115–152, 1995.

[6] Pattie Maes. The agent network architecture (ana). *SIGART Bull.*, 2(4):115–120, July 1991.

[7] Ronald C Arkin and Tucker Balch. Cooperative multiagent robotic systems.

[8] Ronald C Arkin and J David Hobbs. Dimensions of communication and social organization in multi-agent robotic systems. In *Proceedings of the second international conference on From animals to animats*, volume 2, pages 486–493, 1993.

[9] Ronald C Arkin, Tucker Balch, and Elizabeth Nitz. Communication of behavorial state in multi-agent retrieval tasks. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 588–594. IEEE, 1993.

[10] Nuno Lau, Luis Seabra Lopes, Gustavo Corrente, Nelson Filipe, and Ricardo Sequeira. Robot team coordination using dynamic role and positioning assignment and role based setplays. *Mechatronics*, 21(2):445 – 454, 2011. Special Issue on Advances in intelligent robot design for the Robocup Middle Size League.

[11] Kian Hsiang Low, Geoffrey J Gordon, John M Dolan, and Pradeep Khosla. Adaptive sampling for multi-robot wide-area exploration. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 755–760. IEEE, 2007.

[12] Brian Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53. ACM, 1998.

[13] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386, 2005.

[14] Stefano Carpin. Fast and accurate map merging for multi-robot systems. *Autonomous Robots*, 25(3):305–316, 2008.

[15] Torbjørn S Dahl, Maja J Matarić, and Gaurav S Sukhatme. Emergent robot differentiation for distributed multi-robot task allocation. In *Distributed Autonomous Robotic Systems 6*, pages 201–210. Springer, 2007.

[16] Lino Figueiredo, Isabel Jesus, JA Tenreiro Machado, J Ferreira, and JL Martins de Carvalho. Towards the development of intelligent transportation systems. In *Intelligent Transportation Systems*, volume 88, pages 1206–1211, 2001.

[17] Girma S Tewolde, Changhua Wu, Yu Wang, and Weihua Sheng. Distributed multi-robot work load partition in manufacturing automation. In *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, pages 504–509. IEEE, 2008.

[18] Adriano Fagiolini, Marco Pellinacci, Gianni Valenti, Gianluca Dini, and Antonio Bicchi. Consensus-based distributed intrusion detection for multi-robot systems. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 120–127. IEEE, 2008.

[19] Noa Agmon, Sarit Kraus, and Gal A Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2339–2345. IEEE, 2008.

[20] Pedro Lima, Luis Custodio, Levent Akin, Adam Jacoff, Gerhard Kraetzschmar, Ng Beng Kiat, Oliver Obst, Thomas Rofer, Yasutake Takahashi, and Changjiu Zhou. Robocup 2004 competitions and symposium: A small kick for robots, a giant score for science. *AI Magazine*, 26(2):36, 2005.

[21] RoboCup. `http://www.robocup.org/about-robocup/objective/`. Accessed May 24, 2014.

[22] JL Azevedo, B Cunha, AJR Neves, J Cunha, R Dias, P Fonseca, N Lau, E Pedrosa, A Pereira, and J Silva. Cambada, hardware description (2014).

[23] Frederico Santos, Luís Almeida, Paulo Pedreiras, L Seabra Lopes, and Tullio Facchinetti. An adaptive tdma protocol for soft real-time wireless communication among mobile autonomous agents. In *Proc. of the Int. Workshop on Architecture for Cooperative Embedded Real-Time Systems, WACERTS*, volume 2004, pages 657–665, 2004.

[24] António JR Neves, Daniel A Martins, and Armando J Pinho. A hybrid vision system for soccer robots using radial search lines. In *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open-ROBOTICA*, pages 51–55, 2008.

[25] Bernardo Cunha, José Azevedo, Nuno Lau, and Luis Almeida. Obtaining the inverse distance map from a non-svp hyperbolic catadioptric robotic vision system. In Ubbo Visser, Fernando Ribeiro, Takeshi Ohashi, and Frank Dellaert, editors, *RoboCup 2007: Robot Soccer World Cup XI*, volume 5001 of *Lecture Notes in Computer Science*, pages 417–424. Springer Berlin Heidelberg, 2008.

[26] João Silva, Nuno Lau, and António J. R. Neves. Cooperative detection and identification of obstacles in a robotic soccer team. In Calin Ciufudean and Lino García, editors, *Advances in Robotics - Modeling, Control and Applications*, pages 219–235. iConcept Press, January 2013.

[27] Hidehisa Akiyama and Itsuki Noda. Multi-agent positioning mechanism in the dynamic environment. In Ubbo Visser, Fernando Ribeiro, Takeshi Ohashi, and Frank Dellaert, editors, *RoboCup 2007: Robot Soccer World Cup XI*, volume 5001 of *Lecture Notes in Computer Science*, pages 377–384. Springer Berlin Heidelberg, 2008.