Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2009

Ricardo Peixinho Sequeira

Estratégia de coordenação da equipa de futebol robótico CAMBADA

Strategic coordination of CAMBADA robotic soccer team

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática, **2009**

Ricardo Peixinho Sequeira

Estratégia de coordenação da equipa de futebol robótico CAMBADA

Strategic coordination of CAMBADA robotic 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 de Professor Doutor Nuno Lau e Professor Doutor Luís Seabra Lopes, professores auxiliares do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

o presidente / president

Doutor Tomás António Mendes Oliveira e Silva
Professor Associado da Universidade de Aveiro

Doutor Agostinho Gil Teixeira Lopes
Investigador Auxiliar da Universidade do Minho

Doutor José Nuno Panelas Nunes Lau
Professor Auxiliar da Universidade de Aveiro

Doutor Luís Filipe de Seabra Lopes
Professor Auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

Agradeço aos meus pais e amigos por todo o apoio dado ao longo desta caminhada de 5 anos.
Agradeço a todos os professores que ao longo destes anos me ensinaram as ferramentas do meu ofício e em especial ao Professor Nuno Lau e ao Professor Luís Seabra Lopes pela a orientação neste último ano.
Agradeço a toda a equipa do CAMBADA, em especial à malta do alto nível.
O meu obrigado a todos.

palavras-chave Sistemas Multi-agente, Coordenação, Cooperação, RoboCup

Resumo

CAMBADA, que significa Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture, é a equipa de futebol robótico da Universidade de Aveiro que participa na Liga dos Robots Médios no RoboCup. Esta equipa foi criada e desenvolvida pelo grupo de investigação Actividade Transversal em Robótica Inteligente (ATRI) que está integrado no Instituto de Engenharia Electrónica e Telemática de Aveiro.
O objectivo deste trabalho é aumentar a coordenação e cooperação dos robots de forma a trabalharem em equipa, em especial nas marcação das bolas paradas devido à recente alteração nas regras desta liga do RoboCup.
Este trabalho melhorou e aumentou a utilização de passes durante o jogo, especialmente nas bolas paradas. Estas melhorias seguem um metodologia baseada em papeis. A estratégia de todas as bolas paradas pode agora ser definida antes de cada jogo e no intervalo sem ser necessária a recompilação do código. Estas melhorias contribuiram para o 3ºlugar alcançado no RoboCup 2009 em Graz, Austria.

keywords Multi-agent systems , Coordination, Cooperation, RoboCup

Abstract

CAMBADA, which means, Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture, is the robotic soccer team from University of Aveiro that participates in Middle Size League of RoboCup. This team was created and developed by the Transverse Activity on Intelligent Robotics (ATRI) research group from IEETA.
The purpose of this work is to increase the coordination and cooperation of the team so that they perform a better teamwork, specially in set pieces due to the recent change in RoboCup MSL rules.
This work improved and increased the usage of passes during the game specially in set pieces. The improvements made use a role based method. The strategy for the set pieces can now be configured before each game and in half-time without the need to recompile the code. This improvements helped the team achieve the 3rd place in RoboCup 2009 in Graz, Austria.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 RoboCup

1.1.1 Introduction

Robocup [17] is an international initiative with the purpose of promoting research in Artificial Intelligence, Robotics and related areas. Soccer was one of the chosen environment because in a game various technologies must be incorporated including: design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, and sensor-fusion. The Robocup ultimate goal is *By the year 2050, develop a team of fully autonomous humanoid robots that can play and win against the human world champion football team*.

There are several leagues in RoboCup in different applications domain:

1. Robocup Soccer
   - Simulation League
   - Small Size League
   - Middle Size League
   - Standard Platform League
   - Humanoid League
2. Robocup Rescue
   - Simulation League

- Robot League

3. Robocup@home

4. Robocup Junior

    - Soccer
    - Dance
    - Rescue

1.1.2 Middle Size League

In this league, each team has up to 5 players in a field with 18 by 12 meters (Figure 1.1). The field is symmetrical. Each robot is completely autonomous. The used rules [20] are an adaption from FIFA rules. Each robot base must fit in a square with 50x50 centimeters with no more than 80 centimeters height and cannot weight more than 40 Kilograms. The field must be green colored, the lines must be white. The ball is orange size 5 by FIFA standards. The robots must be mainly black. Since last year there are two main changes in the rules.

First, in the set pieces the ball must move freely for 50 cm and touch another teammate before a goal can be awarded. This change was made to encourage passes during the game. The other change was the reduction from 6 to 5 of the number of robots in each team. These changes are aimed to enhance cooperation during games.

1.1.3 Rule changes for 2009

The 2009 rules [20] suffered a significant change in the set pieces. Before the change the two main points in set pieces rules were that:

- The opponent robots need to be at least two meters away from the ball.
- The ball has to touch another robot before it enters the goal to be a valid goal;

The new rules are more specific in how to take a set piece:

- Only one robot of the attacking team can be at a distance of less than one meter from ball.
- The robot taking the set piece can only dribble the ball for twenty centimeters;

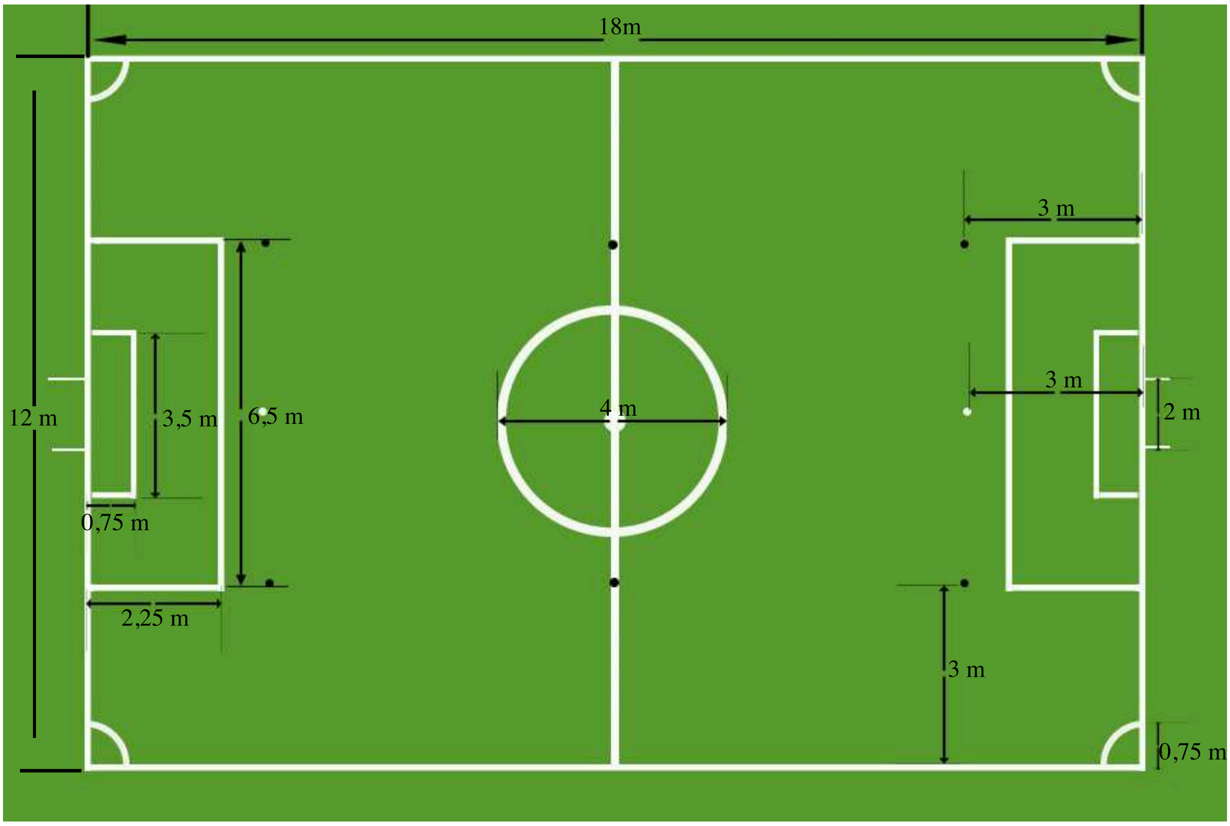


Figure 1.1: Official MSL field

- The ball after being kicked or pushed must roll freely for fifty centimeters before a robot of the attacking team can touch the ball;
- The ball has to touch another robot of the attacking team before it enters the goal to be a valid goal;

The rule of the two meters for the opponent team is maintained. These new rules are an encouragement to passes.

1.2 The problem

CAMBADA[1] is the soccer team from University of Aveiro playing in RoboCup Middle Size League [20] (MSL). In MSL games, and soccer games in general, teamwork is a key component. To achieve a significant teamwork it has to exist cooperation and coordination

[1] Cooperative Autonomous roBots with Advanced Distributed Architecture

between the robots. The rules have evolved to force the cooperation between robots. First the set pieces could be direct, then they were changed to indirect set pieces. Until this year the act of passing the ball was not mandatory, but with the rules, which force to make a pass, it is needed to achieve a certain level of coordination that only a few teams achieved in the past. Also in RoboCup there is a highly competitive side and if you take a set piece always in the same way it is easier for the opponent team to defend it. So, to prevent this, it is needed a method to change the strategy in an easy and fast manner without the need to recompile any code.

1.3 Objectives

The purpose of this work was to improve the coordination and cooperation in the CAMBADA team. The set pieces were changed in order to obey the new rules. To do this some roles were changed to improve the coordination which is necessary to achieve a good result. Some roles need to be parameterized so they can become more dynamic and easy to change. The CambadaConfig tool will be improved and will allow to change the parameters of the new set pieces and other parameters.

1.4 Thesis structure

This thesis has 6 others chapters. The second chapter focus on CAMBADA, describing it in detail. The third chapter focus on cooperation and coordination in the other MSL teams and in other applications. The fourth chapter is about the work done in the new roles and other changes to the code itself while the fifth chapter describes the functionalities and improvements in the CambadaConfig tool. The sixth chapter presents the results of the developed work. The seventh chapter is an analysis of the work done and some thoughts about the future.

Chapter 2

CAMBADA team

2.1 Background

The CAMBADA team (Figure 2.1)[11] was created in October 2003 in IEETA. The project involved the construction of the hardware and the development of the software. In the software layer there are several problems to address like: image processing, sensor filtering, artificial intelligence.

Figure 2.1: CAMBADA team

The team has participated in several competitions along the years:

- Robótica 2004 in Porto, Portugal;
- RoboCup 2004 in Lisbon, Portugal;
- Robótica 2005 in Coimbra, Portugal;

- DutchOpen 2006 in Eindhoven, Holland;
- Robótica 2006 in Guimarães, Portugal;
- RoboCup 2006 in Bremen, Germany;
- Robótica 2007 in Paderne, Portugal;
- RoboCup 2007 in Atlanta, USA;
- Robótica 2008 in Aveiro, Portual;
- RoboCup 2008 in Suzhou, China;
- Robótica 2009 in Castelo Branco, Portugal;
- RoboCup 2009 in Graz, Austria

CAMBADA was the RoboCup MSL world champion, title won in RoboCup 2008 in China [4] and is the national champion for the third time in a row, title won in Robótica 2009 in Castelo Branco [6]. CAMBADA won the third place in RoboCup 2009 in Graz [5]

2.2 Hardware

The CAMBADA robot (Figure 2.2(a)) is of circular shape and has a cut out section to grab the ball. In this section the kicker and the grabber are installed. The kicker is a solenoid actuator and the grabber is a modelism motor with a wheel to maintain the ball under control. The vision system has a hyperbolic mirror and a camera in order to achieve an omnidirectional point of view. For this purpose a Point Grey camera with a 4mm lens is used. To aid localization there is a digital compass in every robot to allow it to tell in which side of the field it is since the field is symmetric and localization is based on the detection of white lines.

The robot is powered by three DC motors with holonomic wheels which allow it to move in every direction (Figure 2.2(b)) while looking at any other direction. At the moment the maximum velocity is around 2 m/s, making CAMBADA robots one of the slowest teams of the RoboCup MSL. In a near future CAMBADA robots will have new engines which will increase the top speed and make them more competitive.

The actuators are controlled by several micro controllers (Figure 2.3). It is used a modified CAN protocol [1] to allow the communication between the different modules.

laptop through a Firewire and an USB link.

2.3 Software

2.3.1 RTDB

The Real Time DataBase [9] is where the information is stored to be accessed by the different processes (Figure 2.3.1). It has one shared section which sends information through wireless every 100 ms to the other robots. It is used an adaptive TDMA protocol [26, 27] to provide a better QoS. The shared section contains information about the robot like self-position, ball position and obstacles. The local section contains informations that is to be used only by the other processes.

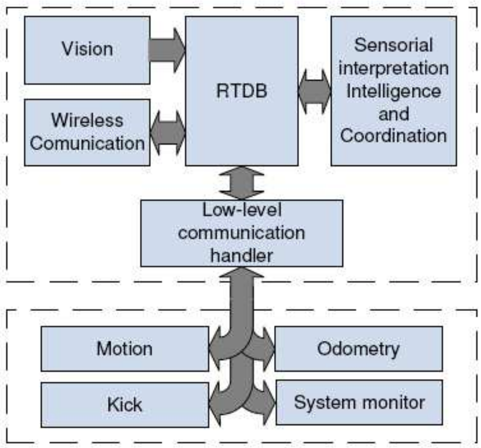


Figure 2.4: Software Architecture. Adapted from [12]

2.3.2 Low-level communication

The information exchange between the laptop and the hardware is done using a gateway that is connected via an USB connection to the laptop and to the micro controllers using the FTT-CAN protocol. The information is passed using the RTDB. In worst case scenario the delay between the orders being issued and performed can be up to 70 ms.

2.3.3 Vision

The vision software processes each frame from the cameras using radial search lines to analyze color and contrast [21]. The vision recognizes the ball, obstacles in the field and the lines. The relative positions are stored in the RTDB to be used by the others processes. The latency is around 15 ms.

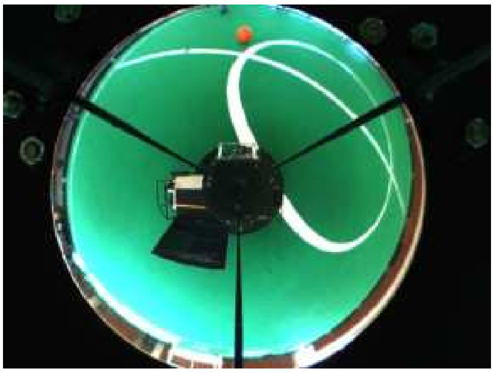


Figure 2.5: Vision frame

2.3.4 Sensorial interpretation, intelligence and coordination

The CAMBADA agent, which is the integration/decision module, is where all the decisions are taken based on the worldstate. The worldstate is filled with the information that comes from the RTDB by the `Integrator` class. The high-level functions are based on the concepts of role and behavior [30]. The `Decision` class will assign the role based on the environment. Each role will run an internal state machine and use behaviors. The behaviors are responsible to send orders to the low-level layer.

The CAMBADA coordination is based in SBSP [23, 19] strategies used in RoboCup Simulation League with some changes in order to adapt to the MSL specifications [18]. For role assignment it is used a DPRE algorithm which will assign each role/robot to the strategic positions according to priorities and number of active robots [23, 19].

In order to filter errors and achieve a more accurate position of the robot and ball it is used a Kalman filter and a Linear Regression [28]. In CAMBADA it is used closed-loop control and PID. Each behavior has its own PID tuned for the desired performance.

In the set pieces two roles are used, *Role Toucher* and *Role Replacer*. These roles use the behaviors `MoveToAbs`, `Kick` and `StopRobot`. In the set pieces both robots, the one with

Role Replacer and the one with *Role Toucher*, are aligned with the opponent goal and the toucher pushes the ball until it is engaged in the *Replacer* moving away immediately. The *Replacer* kicks the ball as soon as possible. in the corner kick the strategy is different. The toucher makes a small pass to the *Replacer* which is in a hard-coded position [16].

Chapter 3

Existing Coordination and Cooperation methodologies

3.1 Introduction

Coordination means to make people or in this case robots, function efficiently and in an organized way. Cooperation means working together for a common purpose. So with coordination and cooperation the result is robots functioning together efficiently and in an organized way towards a common purpose.

Coordination is divided in explicit and implicit. In implicit coordination each agent is independent and works towards his own goals. By doing this it also contributes to the team goal. In explicit coordination two or more agents work together for the same goal, exchanging information between them.

When we have several robots working in the same environment towards the same goal their performance will only be better, comparing to a single robot doing the same task, if they cooperate and coordinate their actions. This problem appears not only in MSL games but generally in coordinated multi-robot system.

In MSL games, cooperation is very important not only for strategy purposes but also for sharing information [15]. One of the near goals to be achieved by most teams is to make a pass. To do so the robot that will pass and the one which will receive the ball need to coordinate [31] their actions. Coordination will play a bigger part in MSL games while the game style will approach an human soccer game.

3.2 Role-based architecture

In many team sports there are several tasks that have to be fullfiled simultaneously. For example, in real soccer there are eleven players in the field and they all have a different role in the team. Only this way it is possible to achieve the different goals at the same time, like defending while attacking.

Another league of the Robocup that make use of multi-agent cooperation and had contributed to the use of it in MSL is the Simulation League. Many of the concepts and ideas before they started being used in MSL, were first applied in Simulation League. CAMBADA team specially benefited of a previous participation with success in the Simulation League by the FC Portugal team [23, 22].

In MSL all robots also can not do the same thing at the same time. For instance, if one team is attacking it needs to continue defending its goal. To do so there must be at least one robot with a defensive role despite its team is attacking. The role assignment can be static or dynamic [30]. In a dynamic assignment the roles change between robots in response to the environment.

Roles are used to facilitate the coordination, they do this by reducing the possible actions a robot has. For example, only the robot with the Striker role can try to score a goal during free play.

3.3 Coordination and Cooperation in CAMBADA

The CAMBADA team during the entire game uses a formation (Figure 3.1) to position the robots in the field. A formation has a base position for each robot and an attraction value to the ball. The final position is the result of the ball position multiplied by the attraction which is then summed to the base position. The robots may change positioning during the game since it is used a dynamic role assignment. The role of goal-keeper is assigned to the robot with shirt number 1. CAMBADA uses three roles during free play and three more are only used in set pieces [16, 14].

- Free Play
    1. Striker - pursues the ball and try to score goals
    2. Midfielder - maintains the robot at a strategic position
    3. Goalie - static role of the goal-keeper

- Set Pieces

    1. Replacer - role to kick the ball into play
    2. Toucher - pushes the ball until it is engaged in the Replacer
    3. Barrier - Defensive role for opponent set pieces

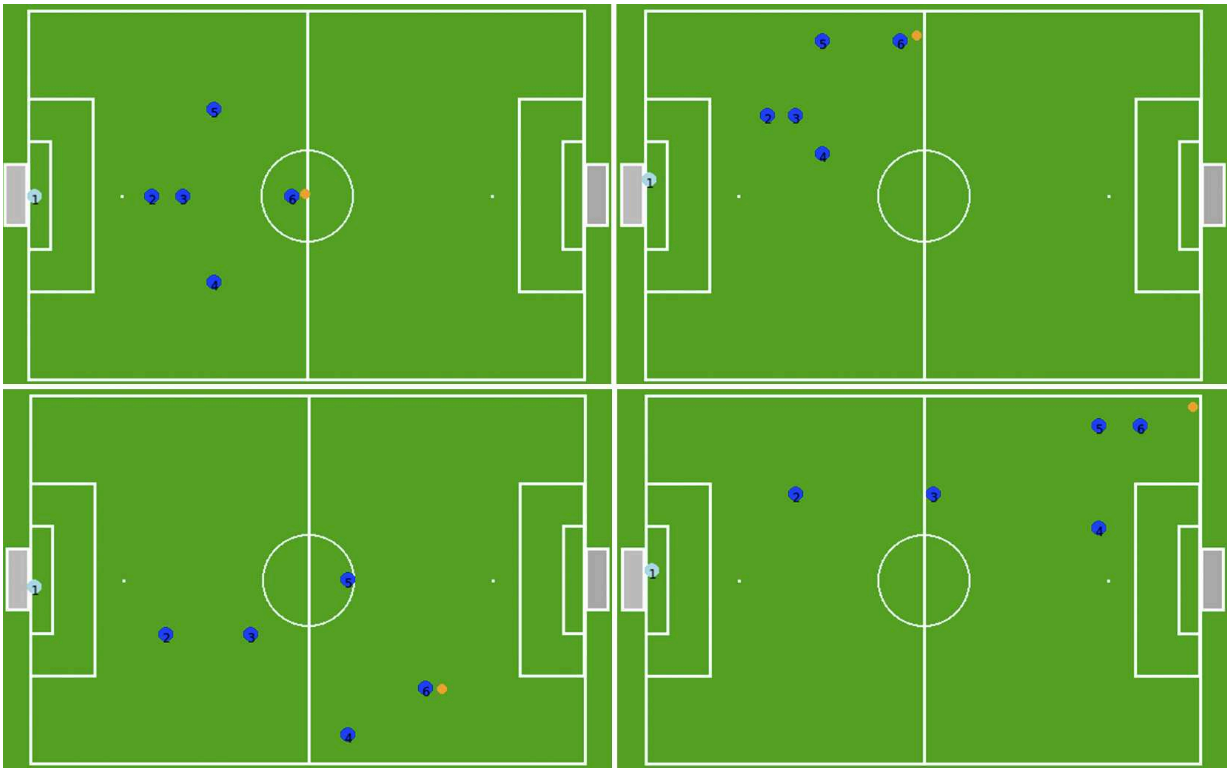
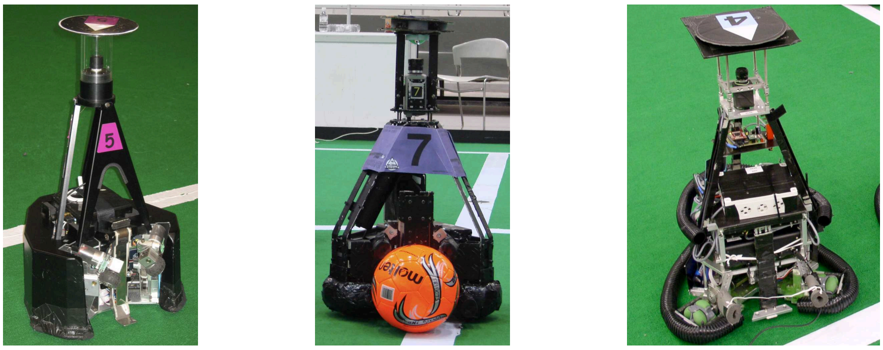


Figure 3.1: CAMBADA formation in 2008. Adapted from [14]

In offensive set pieces two distinct roles are used, Replacer and Toucher. The remaining robots will be midfielders. The Toucher will push the ball to the Replacer and then move away when the Replacer flags the ball engaged.

3.4 Coordination and Cooperation in MSL

In MSL are used different approaches to deal with the cooperation and coordination problem. Only the teams that performed better or have other relevant work will be analyzed.



(a) Tech United robot

(b) Brainstormers Tribots robot

(c) RFC Stuttgart old robot (CoPS)

Figure 3.2: MSL different robots

3.4.1 Tech United

The Tech United (Figure 3.2(a)) team used to have static roles assigned during the game, but in 2009 they will use a dynamic role assigner [7](Figure 3.3). They have one block called Strategy Decider that will pick a formation (e.g. 1 keeper, 2 defenders, 2 attackers) and the roles based on the information of the world state. Then the Role Assigner will assign each role to the best suitable robot. This choice is based on positions, of the entire team and of the ball, and if the robot is fully operational or not. After the role is assigned another block takes charge, the Behavior Executer. This block will use the information of the world state in order to act. Its actions are controlled by a behavior tree. The selection of strategy and role assignment is weight based. These weights are defined before a game. Future work will introduce learning so that the weights are adjusted during a game. This way they intend to improve their team cooperation which they lacked due to the static roles.

3.4.2 Brainstormers Tribots

The Tribots (Figure 3.2(b)) roles are initially assigned by the Teamcontroll which also decides what formation to use depending on the number of robots in the field and robots preferences [25]. Their coordination is based on centralized decision. One robot is the master and decides which set play to use. The master is usually the robot closer to the ball. In defensive plays they use a defensive rotation (Figure 3.4). It is the master that

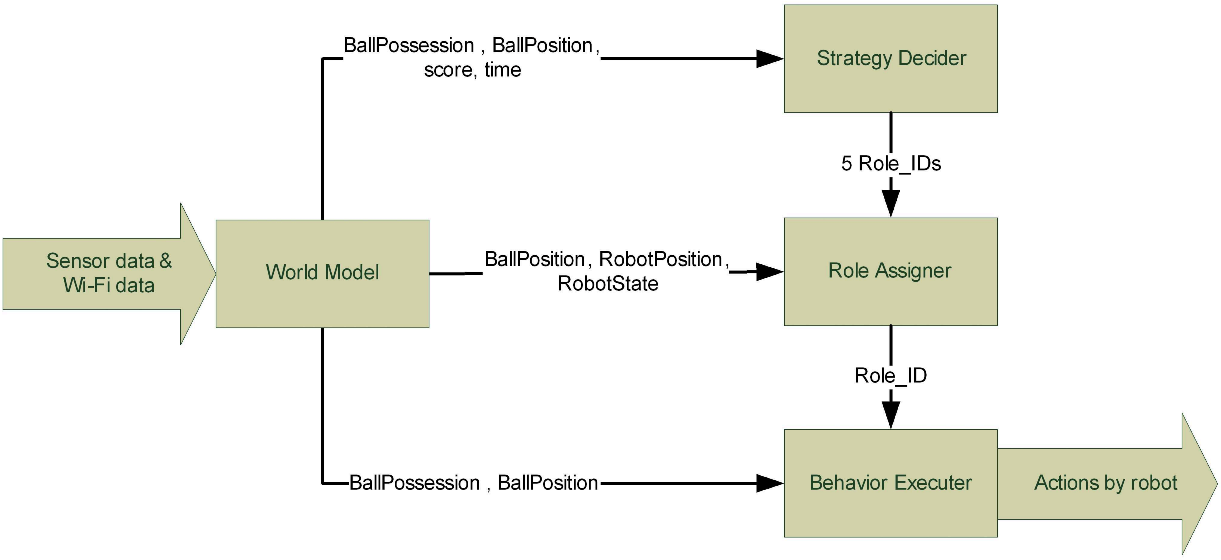


Figure 3.3: Tech United role structure. Adapted from [7].

evaluates the environment and demands a rotation once they have been overplayed through the sideline. This means that the first line of defense is no longer between the ball and the goal. This rotation makes the robots change their roles.

When they have the ball the master robot is the one with the ball. It is the only robot that can request a role exchange for itself and only the master can start a set play like a pass. These decisions are transmitted using keywords. For that to work all robots must know all possible actions and associate them to keywords. When deciding which robot to be master there is a possibility to have an ambiguity. The first robot to request it to Teamcontroll gets it and if they request at the same time the one with lower shirt number gets it.

3.4.3 RFC Stuttgart

This team being the continuation of the CoPS (Figure 3.2(c)) team supposedly use the same software that was used in CoPS. They used dynamic role assignment during the game with subroles so it is possible to have two robots with the same role (Defender, Forward) but with different subroles (Left Defender, Right Forward, etc) [13]. The list of roles is defined before each game. Cooperation is achieved through a modified Interaction net (Figure 3.5) to be used in RoboCup. The mapping of the behaviors uses information of the world state to each agent.

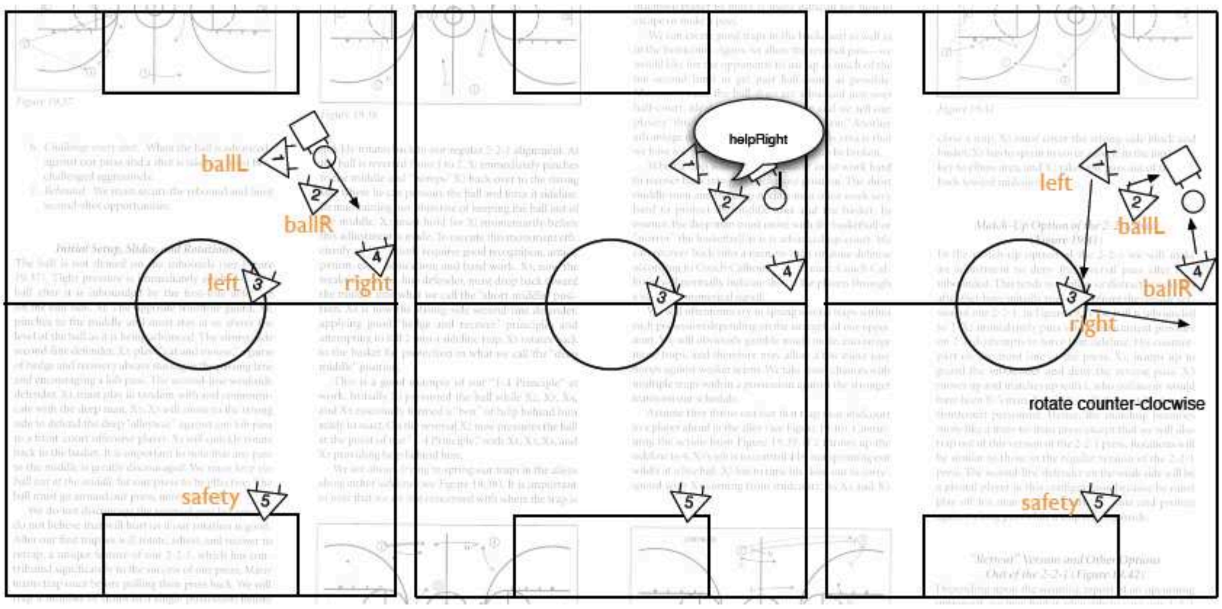


Figure 3.4: Tribots defensive rotation. Adapted from [25].

3.4.4 Carpe Noctem

The Carpe Noctem have developed a new specification language, ALICA[1][29]. Behaviors are modeled using this language. Each robot will estimate the decisions of the teammates and will act based on those [10]. The communication will correct any wrong assumption. The language supports a degree of commitment for strategies with several robots. A pass requires a higher degree of commitment than deciding who attacks and who defends.

3.4.5 CS Freiburg

Each robot of the CS Freiburg will calculate an utility field to all the roles in order to chose the best to fit it[32]. Then the utility fields are shared and it will change to the role chosen if there is no other robot to want it or have it and wants to change it. There is also a hysteresis factor to prevent oscillations between two robots with similar utility fields to the same role.

[1] A Language for Interactive Cooperative Agents

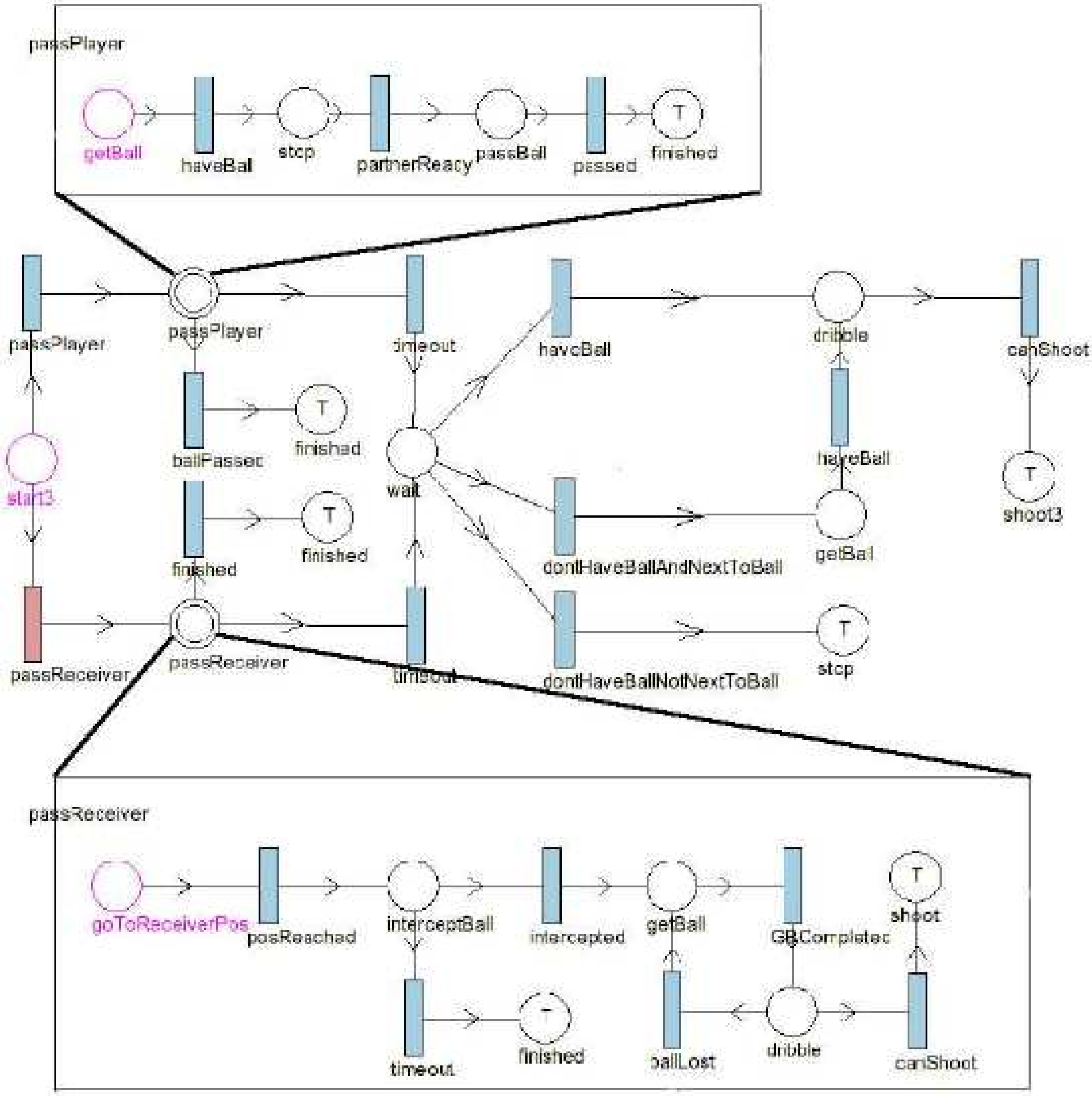


Figure 3.5: Interaction Net for a pass. Adapted from [13].

3.4.6 Summary

In this chapter several methods used by some teams were presented. Every team is working towards more dynamic methods and learning methods so that the coordination and strategy can adapt during the game without human intervention.

Chapter 4

New Coordination and Cooperation methods in CAMBADA

4.1 Introduction

In every sports team there is the need of cooperation and coordination between the players. In MSL the teams that can achieve it gain a significant advantage. In this chapter I will explain the new CAMBADA approach to set pieces and explicit cooperation between two robots to make a pass in free play.

4.2 Set pieces

4.2.1 Introduction

Set pieces are a very important part of CAMBADA game. For example, in last Robocup final, four of the seven goals scored resulted directly from a set piece. So it is important to keep this rating and at the same time adapt to the new rules. In a typical set piece scenario there are two robots directly involved and the rest are in strategic positions.

Set pieces have two points of view, offensive and defensive. In offensive set pieces CAMBADA new approach will have in the same cases three robots involved to have two robots ready to receive a pass. All set pieces are based in two roles, Role Replacer, the robot that makes the pass, and Role Receiver, the robot that will receive the ball. So in each set piece there will be one Replacer and two Receivers, if all the robots are running.

In defensive set pieces usually roles are just based in strategic positioning. Because of that they tend to be static. In this chapter it will also be described a new, more dynamic, Role Barrier.

In this new approach all the strategy for set pieces can be parameterized by the configuration file. In that file there is information about the positioning of the Receivers for every set piece, to whom the Replacer should pass first.

These roles will only be active during ten seconds immediately after the *start* order is given in a set piece.

4.2.2 Role Replacer

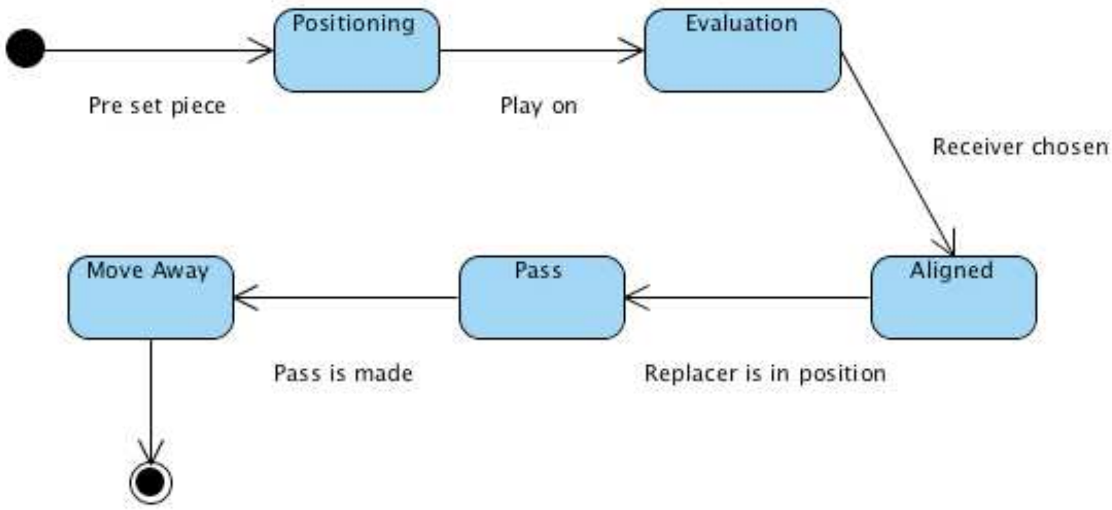


Figure 4.1: Role Replacer state machine

The Replacer role is the one responsible for giving the first touch to the ball in every set piece. This role is the one in charge of the set piece. It will decide to where to pass or kick. After passing it will also try to block incoming opponents.

The Replacer state machine (Figure 4.1) is executed every cycle. All the states and transitions are described in the following sub-sections.

Positioning state

In this state the robot will simply go to the position best suited for the set piece. That position is given by the method *setPlayPosition()*. This method will calculate the position and orientation of the Replacer. In all set pieces the Replacer will place itself thirty centimeters away from the ball. The direction is variable according to the set piece. The role will leave this state as soon as the referee gives the *play on* order.

Evaluation state

This is the most important state in the Role Replacer. It is here that the point to pass or kick will be decided. That decision is made by the *passEvaluation()* method. It will take into account the number of the running robots and the default Receiver chosen in the configuration file.

First the Replacer will check if it is possible to pass to the default Receiver 1 with the aid of the worldstate function *isLineClear()*[28]. This function will return a bool value indicating if there are any obstacles between the robot and a given point. The Receiver will also do this check using the same function and place the result in the *coordinationFlag* which is a worldstate variable that is transmitted to all robots in the team. The obstacles detection is only reliable under five meters so this second check is required for a pass above that distance. The flag has the *LineClear* value if there is no obstacle between the Receiver and the Replacer and the value *notClear* if there is. The ball will only be passed to the Receiver if both evaluations are clear. If not the Replacer will try to pass to the other Receiver. If the two Receivers are blocked the Replacer will pass to a previously chosen position that it will get from the configuration file. If the Replacer is the only running field robot it will kick the ball to a chosen position that is in the configuration file.

Typically the position to kick the ball when both Receivers are blocked should be the opponent goal to reduce the risk of suffering a counter attack. This state is only active for one cycle, the first cycle after the *play on* order has been received.

In order to communicate its decision to the other robots the Replacer will put its *coordinationFlag* with the value *TryingToPass*(shirt number) where the shirt number is the number of the Receiver chosen. When the two Receivers are blocked it will simply put the flag like *TryingToPass*.

Aligned state

This state is not visible in the free kick and throw in set piece because in those the Replacer is already aligned. In this state the Replacer is going to align with the ball and the point chosen to make the pass, it can be another robot or the opposite goal. It is projected a five centimeter circle in the position where the robot must be to be aligned. When the position of the robot is inside the circle it will change state.

Pass state

This is a simple state, the robot will approach the ball and use the behavior kick with the pass option [16]. This option will make the kick behavior use a different kick power and signal the low-level layer that is a pass. The low-level layer will then use a special sequence of two discharges with a time difference of some milliseconds. With this the ball will not gain so much elevation. The pass is considered completed and the state will change when the ball is forty centimeters from its initial position.

Move away state

This is a state where the Replacer will try to give some cover to the Receiver to have the best position possible to kick to goal. This state is not used in the corner kick set piece. The Replacer will be placed two meters from the ball initial position in the direction of the goal in order to block incoming robots.

4.2.3 Role Receiver

The role Receiver is responsible for the positioning of two robots in the set pieces. One of those robots will receive a pass from the Replacer and kick the ball into the opponent goal or will try and cover the other Receiver. There are two Receivers only when all robots are running. The two Receivers distinguish between them by the number of the shirt, the lower number is Receiver 1 and the other is Receiver 2.

Positioning state

In this state the Receiver will read from the configuration file the position it has assigned according to its number and set piece. Usually this position is an absolute one but in some set pieces it is a position relative to the ball, like in free kick and throw in, these ones the positions are not read from the file but calculated. The second Receiver has a cover option in some set pieces. This option will position the second Receiver relative to the ball. This position is also parameterized by the configuration file. The Receiver will leave this state when the play on signal arrives.

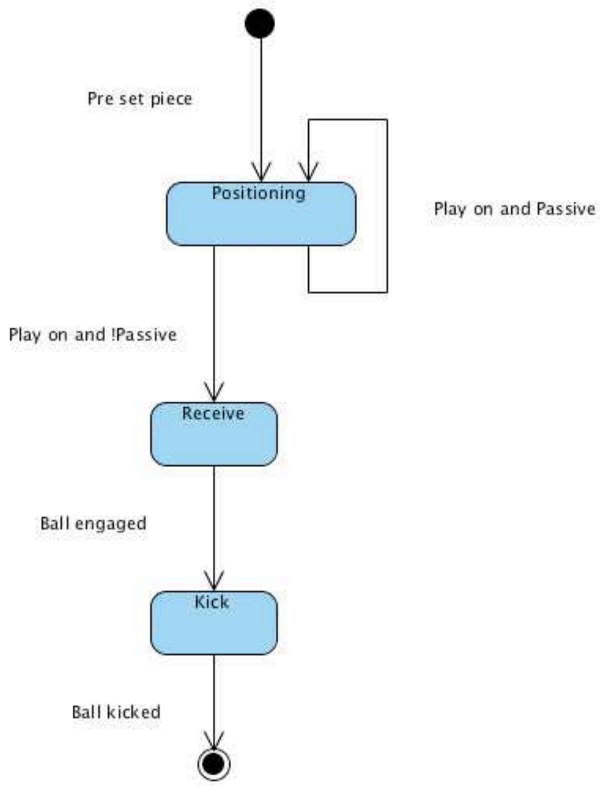


Figure 4.2: Role Receiver state machine

Receive state

In this state the Receiver will read the Replacer *coordinationflag*. If this flag has the value *TryingToPass*(shirt number), where the shirt number is the number of the Receiver, it will use the behavior *CatchBall* [28] to try to receive it. If the shirt number is a different one the robot will simply not move. If the flag has the value *TryingToPass* both Receivers will try to get the ball. The Receiver will leave this state when the ball is engaged or with timeout.

Kick state

In this state the Receiver with the ball will simply try to kick the ball into the opponent goal, independently of the distance to it. It is possible that in some cases the Receiver will not reach this state due to the fact the ten seconds window has expired.

4.2.4 Role Barrier

The initial idea of this new version of the Role Barrier was to have a barrier that could easily be changed in order to adapt to the opponent and the specific set play. This role is only a strategic positioning of the robots in order to protect the goal (Figure 4.3).

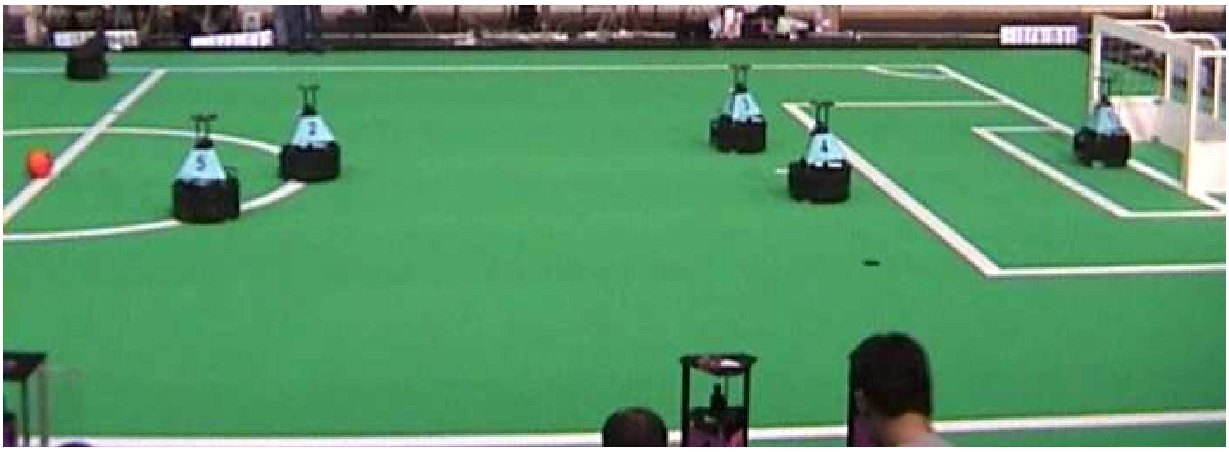


Figure 4.3: Barrier positioning at a Kick Off

The concept is similar to the one used to a free play formation [14]. Basically there is a player base position to the ball position multiplied by an attraction factor is added. All the data is stored in the configuration file and the robots will read the information for the specific set piece and for the position each one has in the barrier which is decided based on the distance to the center of the field and on how many robots are running.

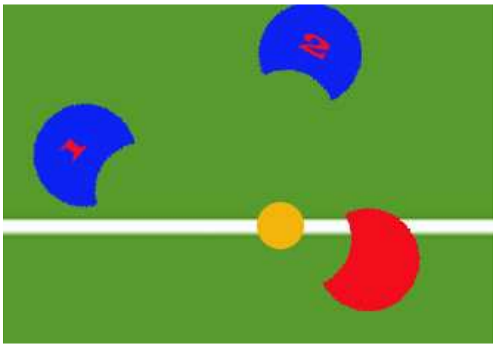
The biggest advantage of this Role Barrier to the previous is that there is a different barrier for each set piece. Each barrier is very customizable and can be adapted to every team strategy.

4.2.5 Examples

In this subsection every set piece will be explained and some examples will be given. I am going to explain how the coordination is done and what can be changed in the set pieces using the configuration file.

Throw in

In the example of the figure 4.4(a) the Receiver 2 has a passive role. With the configuration file the position of the Receiver 2 can be changed, also the angle that Receiver 1



(a) Theoretical positioning - Red robot is the Replacer and blue robots are the Receiver 1 and 2

(b) Real game positioning - Robot 4 is the Replacer and robots 2 and 3 are the Receivers

(c) After play on signal, Replacer covering the Receiver with ball(Robot 2)

(d) Ball kicked and goal scored

Figure 4.4: Throw in

has with the side line can be changed but it always stays one meter from the ball. This 1 meter distance is forced because the short pass is the most efficient form of taking the set piece with the current setup. The Replacer will always be aiming at our goal except when there is no Receiver then it will aim to the opponent goal and shoot directly. Figure 4.4(b) shows a real positioning for a throw in. Immediately after the pass the Replacer covers the Receiver (Figure 4.4(c)) which kicks the ball as soon as possible (Figure 4.4(d)).

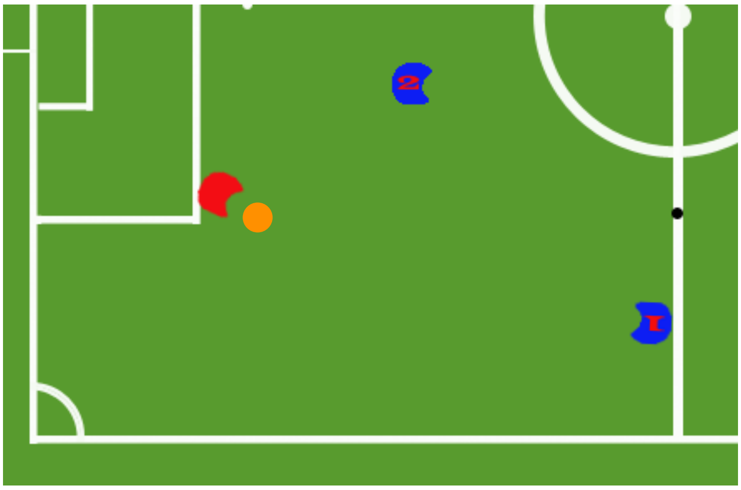


Figure 4.5: Goal kick

Goal kick

In the goal kick the Replacer will be aiming to outside of the field and only after it will align to the target. The Receiver 2 is passive in this set piece and typically it will be placed in front of the goal to provide extra cover in case the ball is lost. The Receiver 1, to whom the pass will be targeted is usually near the side line of the goal kick side so that the ball during the pass does not cross in front of our goal (Figure 4.5).

This is just one possible way to take a goal kick. By changing the parameters in the configuration file the pass can preferentially go to the Receiver 2, and the position of both Receivers can be changed.

Kick off

For the Kick off set piece currently it is being used a long pass to the side (Figure 4.6). The Replacer will check if the line is clear and choose one of the Receivers. The Replacer initially is aiming to the opponent goal and then will align with the target. Both Receivers position can be changed through the configuration file as well as which is the preferential Receiver.

Free kick

The free kick set piece is a more static solution and similar to the one used last year. The Replacer and the Receiver 1 will be near the ball already aligned with the opposite goal (Figure 4.7). The Replacer will pass the ball to the Receiver 1 which will kick it

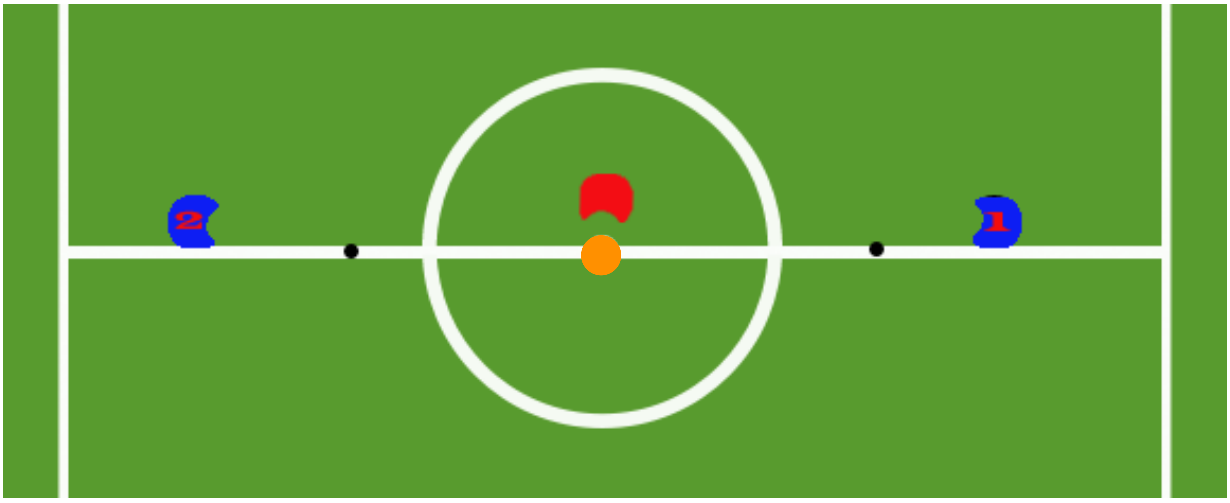


Figure 4.6: Kick off

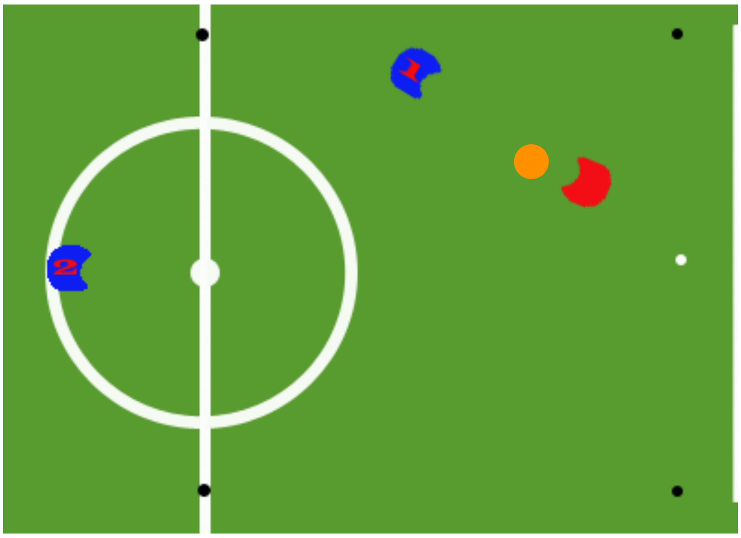


Figure 4.7: Free kick

immediately to the goal. The Receiver 2 is passive and will be placed around the midfield zone in order to cover any counter attacks. The Receiver 2 positioning can be changed with the configuration file. If desired the pass can also be made to the Receiver 2 just by changing the configuration file.

Corner kick

The corner set piece is similar to the kick off one. The Replacer has two possible Receivers and will choose to which to pass based on the default Receiver (chosen on the

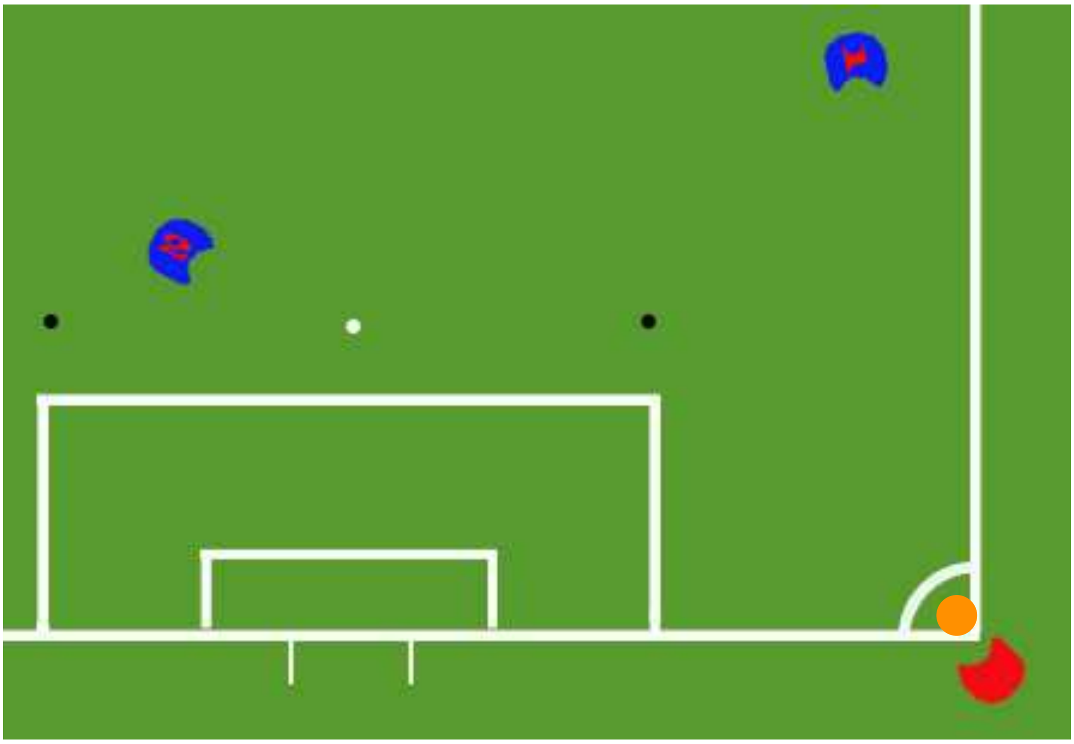


Figure 4.8: Corner kick

configuration file) and on if there are opponent players that can intercept the ball. The positioning chosen is with one Receiver by the side line and another near the penalty mark (Figure 4.8). Both Receivers positioning can be changed in the configuration file.

4.2.6 Exceptions

In some situations the used approach does not support all the possible set pieces. In free kicks when both Replacer and Receiver are already aligned with the opponent goal, if the ball is to close to the side line there is a high possibility that the ball goes out of the field before the Receiver can catch it. In this situation it is better to use the throw in strategy. If the ball is under one meter away from the side line (Figure 4.9) then it is best to use the throw in strategy instead of the free kick strategy. The Replacer will decide which strategy to use, the decision is shared using the *coordinationFlag*. If the flag has the value *freeISthrow* both Receivers will use the throw in strategy.

If a free kick is too close to the corner the angle to score is very narrow and almost impossible to make the ball enter the goal. To take advantage of this free kick and not waste it the best solution is to use the corner kick strategy in these situations. The Replacer

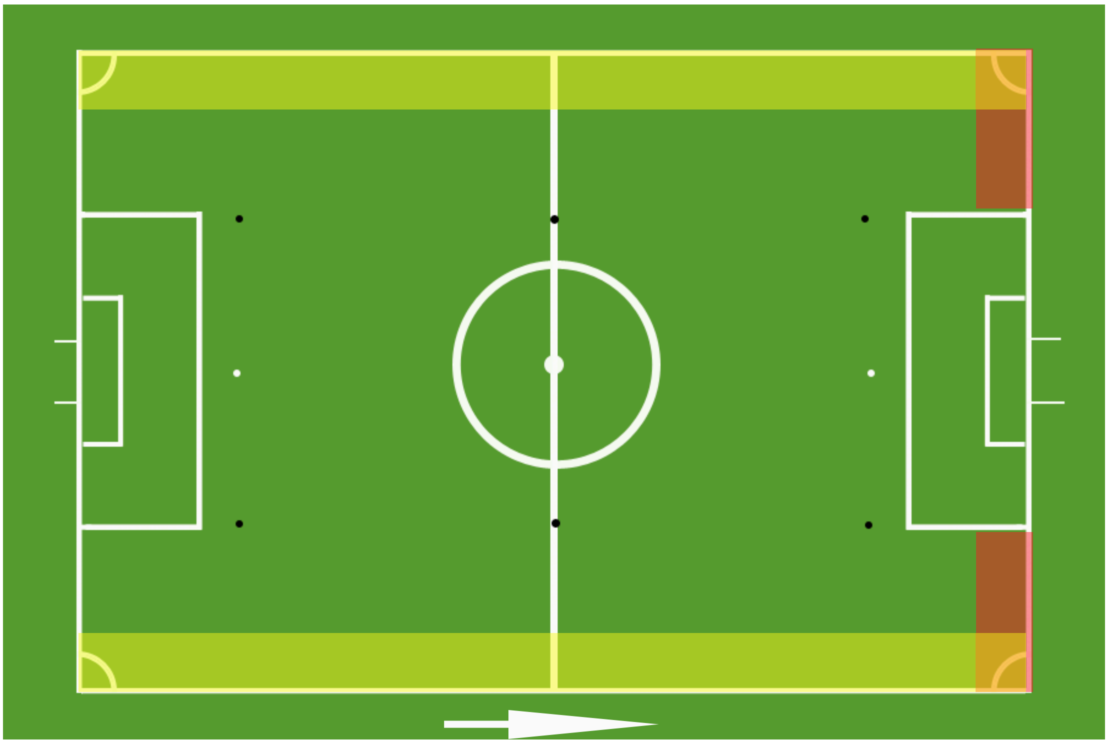


Figure 4.9: Exceptions areas - Yellow area a free kick uses throw in strategy, Red area a free kick uses corner kick strategy

will decide which strategy to use. The decision is shared using the *coordinationFlag*. If the flag has the value *freeIScorner* both Receivers will use the corner kick strategy.

It is easy to change which strategy to use since actual code is common to all set pieces. The only difference is the initial positions of the robots.

4.2.7 Summary

The work described in this section is responsible for the offensive and defensive set pieces of the CAMBADA team. The configuration file has a very important role in the set pieces and is constructed by the CambadaConfig tool which will be described in the following chapter.

4.3 Making a pass in free play

4.3.1 Introduction

In MSL games a pass being made during the game without being in a set piece strategy was never seen except once or twice [24, 33]. Teams avoid doing a pass because it involves many risks and there is a high chance of loosing the ball. In consequence of this, attacks are based on dribbling. Because of that the defensive strategy of most teams is putting all robots around the ball. Of course this type of game does not have any similarity with the real game.

So the CAMBADA team motivated with the rules change for 2009 considered that it would be a good idea to have the possibility of making a pass in free play. This pass would be from the robot that has the ball which under special circumstances would pass it to the other side of the field, changing the attacking flank. This set play would be very useful against teams that concentrate all of the players around the ball.

4.3.2 Role Striker

The Role Striker is responsible to decide that a pass is needed so it has to be changed in order to check the conditions to make the pass(Figure 4.10). The conditions needed for a pass to happen are:

- The Striker must be in opponent field: this is considered a condition because it is a risk to make a pass too close to our goal;
- There are opponent blocking the Striker path towards the opponent goal: while the striker is dribbling it is usual that opponent robots block the way; to overcome this situation a pass is a good solution or the striker is stuck. This usually happens when opponent robots come head to head with our Striker, after recovering the ball the pass to the other flank will leave the other robot free of opponents;
- All field robots running: because making a pass is a risk we believe that it should only be done when all the field robots are running.

These conditions are checked while Role Striker is in the *Score* state dribbling [14]. If the conditions are fullfiled the Striker goes into a new state, *Pass*. In this new state the Striker will only pass when the robot with the Role Receiver 2 is ready and flags it using the *coordinationFlag* of the worldstate. Of course making a pass in free play is very risky

Figure 4.10: Conditions in which a pass should be made in free play. The red robot is the Striker, the blue robot is the Receiver and the black ones are the opponents.

and there is a high chance that the pass is intercepted. So there are some conditions to be checked to verify the success or failure of the pass. First the estimated ball velocity needs to have its direction towards the Receiver. There should be two timeouts, one for the Striker to pass the ball and another for the ball to reach the Receiver. The first will reduce the probability of the Striker being stuck in trying to pass to Receiver and being constantly blocked by the opponents. If that happens the Striker should try and insist with the dribbling. The other timeout is to make sure the Receiver will not be expecting the ball for too much time. A pass that lasts more than three to five seconds is a signal that something went wrong and so the Receiver changes its Role back to Midfielder and goes to its strategic position. Assuming the ball is passed with success, when it is near to the Receiver robot, the Receiver will change its role to Striker.

4.3.3 Role Receiver 2

The Role Receiver 2 has this name because it is a simpler version of the Role Receiver used in the set pieces. It only has two states, POS and RECEIVE. When a robot enters the Role Receiver 2 it will position itself in the symmetric position of the Striker one meter forward (Figure 4.10). When it is at that position it will change to RECEIVER state and will put its flag with *Ready* value where it stays until it changes to Role Striker. The RECEIVE state will align the robot to the ball to make the best possible reception.

The Role Receiver 2 will also evaluate the conditions around it and using the *coordi-*

nationFlag will inform the Striker about them. It will check if there are any obstacles in line of pass, using the *isLineClear* worldstate function and it will also check if there is any opponent near it. If there is a line of pass but there are opponents near it will use the *notFree* flag. If there is not a line of pass it will use *notClear*. If both conditions are good, there is a line of pass and there are not any obstacles near the Receiver, it will use *LineClear* flag.

4.3.4 Summary

When we start thinking in making a pass during free play one of the requirements was that the ball velocity needed to be big enough so the opponent robots could not intercept it. To achieve this it was required a kicking system that could kick the ball close to ground with that speed. Unfortunately that kicking system was not developed so this approach was not tested but it is partially implemented so that next year, hopefully, it can be used.

Chapter 5

CambadaConfig tool

5.1 Introduction

In the previous chapter a configuration file was mentioned where all the options of the set pieces can be stored. The configuration file is written in XML and contains many informations, such as field dimensions, PID parameters and other parameters. The advantage of using this file is that there is no need of recompiling the whole code. To better use this file and edit it safely the CambadaConfig can be used. This tool is a small program that allows to edit in an easy and straight forward way all of the parameters.

5.2 Description

When creating a new formation instead of having to design it, code it and then test it in the robots, CambadaConfig allows to try it out in a simulated field where the ball can be placed in any part of the field and the strategic position of the players can be observed. This feature already existed, but with the creation of dynamic set pieces the positioning of the players needed to be coded. So, some new features were added to this tool in order to make the pure soccer strategy easier to implement.

This tool will also allow to make some changes in half time since it is very fast to use and does not require the code to be recompiled. The tool also provides a more user friendly environment since it is all built using Qt libraries v.3 [3]. The tool also provides a button to check the connectivity of the robots.

All these tasks were previously performed using a text editor and console commands which took time and a lot of attention to details. With CambadaConfig the user can focus on the tactics and let the tool deal with the rest.

There are some data structures also used in the agent to store information (Figure 5.1). To represent the field it is used a Canvas and to store the information of the configuration file it is used the class ConfigXML as well as the class PID to store PID values.

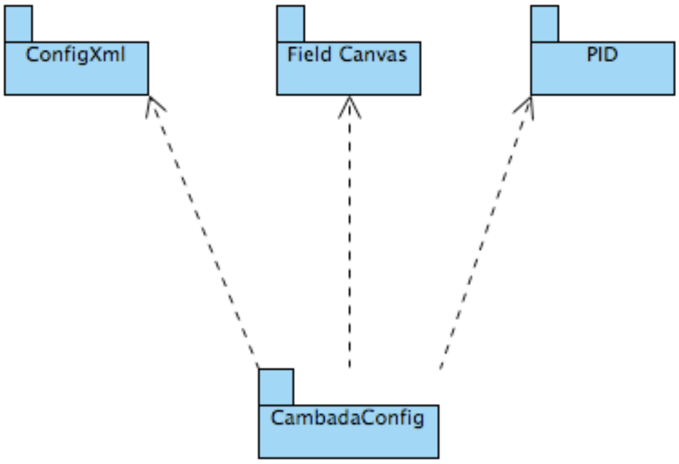


Figure 5.1: CambadaConfig dependencies

5.3 Requirements

This section lists the requirements of this tool. The support of multiple formations is required because we intend to have the formation changed during the game taking in consideration the time and the score. The Control, Field and Parameters tabs are all very similar, simple tables that show the values. The Control and Parameters tabs have an option to add and to delete lines. The Field tab does not need this option but needs a simple way to reset the values to our laboratory field and configure to a normal field. The set pieces and barrier a simple straight forward way of changing the positions and the parameters. The communications tab has to check connectivy with the robots and a simple way of sending the new information to the robots.

5.4 Features

In this section the main features of the CambadaConfig will be explained in detail, namely the formation tab which allows the user to create and edit formation, the set pieces tab, where all the set pieces configuration can be changed, the parameters tab, where the various parameters can be changed and the Control tab, where the interaction between the user and the robot is made.

5.4.1 Formation

The Formation tab (Figure 5.2) is where formations are parameterized and tested. This tab allows the testing and creation of multiple formations. Each player position is related to the ball position, using attractions and a containing box [14]. Initially the positions are set using the center of the field as referential. When *Run Simulation* button is pressed, a ball appears that can be moved using the mouse, allowing to watch the positioning of the robots during the game.

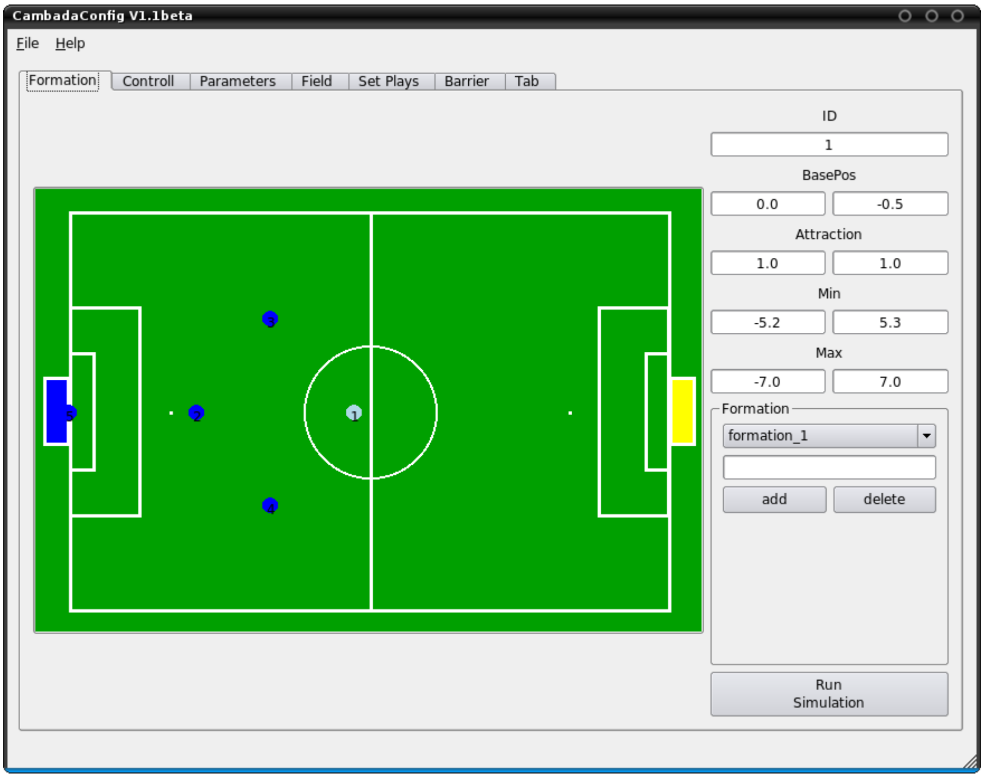


Figure 5.2: Formation tab

The attractions are set in the form at the right. The strategic position is calculated using the base positioning adding the relative ball position multiplied by the attraction factor.

5.4.2 Set pieces

The *Set Pieces tab* (Figure 5.3) is where all the strategy for the set pieces is defined. Using the combo box it can be chosen which set piece it is shown. The ball is placed in a possible place to that set piece to occur. The ball has marked around it a 1 meter circle and the 2 meters circle to aid the positioning of the robots. In the bottom are some instructions which are specific to each set piece. In free kicks and throw ins the Receiver 1 positioning can not be changed. The *Receiver to Pass* combo box allows the user to choose to which Receiver the Replacer will first try to pass. If the option *0* is chosen then it will pass to point set in Figure 5.3 by robot 4.

Figure 5.3: Set Pieces tab

The cover option is only available in free kicks and throw ins. It allows to change the positioning of the Receiver 2 from absolute to relative to the ball if active. In the case presented in figure 5.3 the Receiver 2 is with cover option on. If the ball was on the left side of the field the Receiver 2 would be to the right of the ball. This happens because the field is divided horizontally with a symmetric axis. So if a set piece is defined in the right side of field when it happens on the left side it will look symmetric. This is very important to ensure that the covering robot does not end outside the field.

5.4.3 Other parameters

There are other tabs which are simpler, like *Field tab* (Figure 5.4(a)) that is only a table. There are two buttons in the bottom. One restores the dimensions of our field and the other set the ideal dimensions for a proper field (18 m x 12 m). The *Parameters tab* and the *Control tab* are similar to this one, but instead of those two buttons they have an add and a remove button. The *Control tab* is where all PID values are stored. The *Parameters tab* as the name suggests is where some parameters that are likely to be changed often are stored, this way there is no need to recompile the entire code when changes occur.

The *Communication tab* (Figure 5.4(b)) is a tab where the user can ping the robots, send the code to the chosen robots by selecting them in the specific checkbox and use the *Reconfigure* button to force the agent to read the new configuration files. This button is used to force the agent to read the new configuration file. The feedback of this operation appears in the textarea which can be cleaned using the *Clean* button.

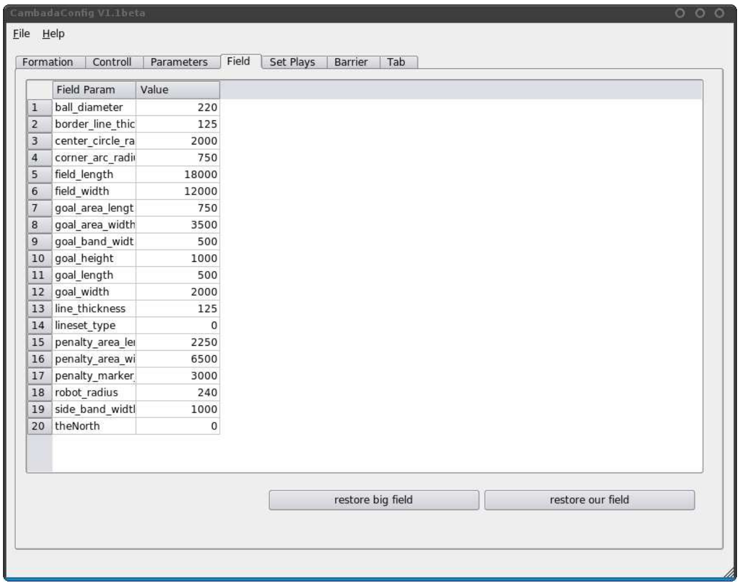
The *Barrier tab* is very similar to the *Formation tab* but sets up a formation to each set piece and it can be tested in the same way formations are. During the simulation the ball will have the one meter and the two meters line around it to aid the positioning of the robots. The number of the robots determines the priority of the positions.

5.5 Integration with agent

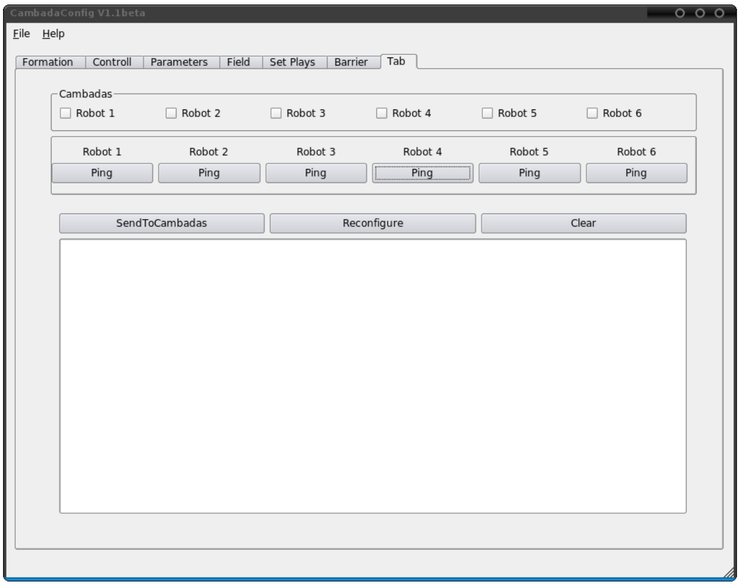
The information contained in the configuration file is read by the class *ConfigXML* which makes use of libxml [2]. The information is stored in a map[1]. An instance of *ConfigXML* is inside worldstate so every other class in the agent can access it.

Regarding the set pieces, before accessing the data each role will analyze in which set piece it is and only then it will get the data of that set piece. The Receivers also have to determine if it is Receiver 1 or 2. If it is Receiver 2 it needs to check cover option to see if the values stored are an absolute position or an offset to the ball position. In the Role Barrier the robots need to be ordered and then the positions are assigned to each one according to its priority which is defined in the tool. The robot with highest priority will be the number 1 robot and the one with the lowest priority will be robot 4.

[1] http://www.cplusplus.com/reference/stl/map/



(a) Parameters tab



(b) Communication tab

5.6 Summary

The CambadaConfig tool is responsible for the configuration of all the set pieces which were described in the previous chapter. This tool is starting to have a key role in CAM-

BADA team and in the future with new features being added it can be turned into the control center of the team.

Chapter 6

Results

6.1 Introduction

In this section the results of the work described in the two previous chapters will be presented and discussed.

The CAMBADA team participated and won the Infaimon Cup in Robótica 2009 at Castelo Branco. The results that will be presented here come from that event and five tests performed in a weekend spent on ISEP[1] testing and tuning the robots for RoboCup 2009 in a field with the official size (18m x 12m) since the field used for development at University of Aveiro only has 7m x 5m. The semi-final game of RoboCup 2009 against Tech United will also be analyzed.

6.2 Results

For these tests several parts of the set pieces are took in account. First, the positioning of the robots and if the wrong positioning affects the outcome of the set piece or not. Then, there is the pass, in the pass it is evaluated if the target is correctly chosen, the power and the if the receiver catches the ball successfully. Finally it is took in consideration the kick, if it was immediately or not and if it resulted in goal or not.

[1]Instituto Superior de Engenharia do Porto

6.2.1 Results of Robótica 2009

In Robótica 2009 out of 8 scheduled games CAMBADA only played 5 because of absence of the opponent team in the other 3. In this section only the set pieces will be analyzed, since the others features such as Role Barrier and passing during a game were not used.

The results was gathered by direct observation from the videos of those games (Table 6.1). The CAMBADA team always records, with a video camera, all games in official competions in which it participates. This is very useful for latter analysis. In this analysis a set piece is considered successful when the Receiver has the ball engaged and can kick the ball, whether it is goal or not it is not considered since the kick is done by the `kick` behavior which is not a part of this work.

	Set pieces taken	Set pieces successful	Success rate
Free kick	3	3	100%
Goal kick	4	4	100%
Kick off	5	5	100%
Corner kick	10	8	80%
Throw In	16	14	87,5%
Totals	38	34	89,5%

Table 6.1: Results from set pieces in Robótica 2009

6.2.2 Results at ISEP

In the weekend at ISEP 5 tries of each set piece were performed with four running robots, without the goal keeper and without an opponent team. These tests had a success rate of 100%. This success rate reflects the set piece sequence and not if it was goal or not. The robots were placed in the defensive side of the field, the ball was placed in the set piece position, this position was different in all tries, before the set piece order was sent from the referee box and the *play on* order was only sent when the situation was stable, meaning there were no robots moving.

6.2.3 Results at RoboCup 2009

In RoboCup 2009 the CAMBADA team played 14 games, winning 12 of them and scoring 71 goals. CAMBADA team achieved an average of 5.1 goals per game, the highest in the tournament. Analyzing the semi final game against Tech United (Table 6.2) which CAMBADA lost by two goals (2 - 0).

	Set pieces taken	Set pieces successful	Success rate
Free kick	6	4	66%
Goal kick	2	1	50%
Kick off	3	3	100%
Corner kick	1	1	100%
Throw In	13	11	84,6%
Totals	25	20	80%

Table 6.2: Results from set pieces in semi-final against Tech United in RoboCup 2009

Outcomes	Number of occurrences	
Player was blocked by opponent	12	60%
Ball went off the field through the goal line	3	15%
Ball hit goal framework	2	10%
Ball was defend by the goalkeeper	2	10%
Player continued by the ball	1	5%
Total	20	100%

Table 6.3: Outcome of successful set pieces in semi-final against Tech United in RoboCup 2009

In this game many of the set pieces did not resulted in a kick to the opponent goal (Figure 6.3) because Tech United robots were in many situations 2 meters away from the ball measured from their geometric center, when they should be 2 meters away from the exterior of the robot, and also because of their speed. In these cases the set piece considered successful when all sequence worked correctly.

6.3 Discussion

In Robótica 2009 there 10,5% of the set pieces that did not work. For these case there are some possible explanations:

- the referee give the *play on* command before the robots were in the right place;

- the wireless environment was too congested which increased the communication delay;
- the coordination flags were not resetted, this usually happens when a set piece command is given, then a play on command, then the same set piece command and only then the play on command.

In the ISEP tryouts there were no failures, this is explained since the probable sources of problems were not present in those trials. We waited until the robots were stable, wireless environment was not congested since there were only four robots running and the order from the referee box were given in the correct and normal order.

The RoboCup 2009 results were not so good as the Robótica 2009 and RoboCup 2008. When compared to the Robótica 2009 results it is expected to be worst since RoboCup has more competitive games. Comparing to the RoboCup 2008 results are worst in part because of the rules changes regarding the set pieces, which give more time to the opponent team to block the offensive players. This happens because a pass needs to be made and the ball must roll freely for 50 centimeters, and this gives the opponent time to reach the offensive robot, if they are fast enough like Tech United robots.

In RoboCup 2009 some of the missed set pieces were because the some of the processes terminated during the set piece during a set piece and the monitor application was not fast enough to start them before the *play on* order. Also some robots were not placed in the right place when the start command was given, this happens because when they were moving towards the correct position there were obstacles in the way and the robot to avoid them did not reached the corrected position in time.

The CambadaConfig tool was used during the Robótica 2009 and it has proved its value, but at the time some user friendly utilities were not implemented, like the 2 lines in the set piece tab which indicates the 1 meter and 2 meters distances. In the weekend at ISEP the CambadaConfig used there was close to final one here described and the improvements were welcome and aided the user to take decisions. In RoboCup 2009 the CambadaConfig was very useful because it allowed the set piece strategy to be changed before every game to surprise the opponent team.

Chapter 7

Conclusion

7.1 Conclusion

This year one of the main goals of the CAMBADA project was the adaptation to new set pieces rules while keeping the same success rate last year set pieces achieved a success rate of 85,7% [16] measured during the RoboCup 2008 final wile this year the success rate of the set pieces was 80% which was measured during the RoboCup 2009 semi-final. So we can see a slightly decrease in the success rate, but I believe the biggest improvement was that the set pieces are no longer static. In fact, the robots decide in some cases how the set piece is going to occur. This allied with the CambadaConfig features it is now possible to have different set pieces for each game adapting them to the opposite team without even have to recompile the code.

One of the things that would improve this performance would be a kicking system that allowed to kick the ball close to ground. With it we could make passes with twice the speed we do and more precise to ease the reception.

With these improvements the CAMBADA team is the one of only two teams that have this kind of control over set pieces and one of few teams that makes an actual pass rather than pushing the ball.

After the RoboCup 2009 where CAMBADA won the 3rd place and the 1st place in the Technical Challenge, these are the results of a positive participation. The CAMBADA team was the only one to achieve a victory against the present World Champions, 1.RFC Stuttgart and it achieved a similar performance as last year regarding goals scored and games won and being the team with most goals scored for the 3rd year in a row. The Tech United had new grabber and kicking systems and 1.RFC Stuttgart, former CoPS, had new robots, even so CAMBADA finished the tournament with the best statistic regarding goals per game.

Even if the CAMBADA set piece success rate was not as good as last year, it was the only team that used a pass in every set piece. The 80% success rate was worst than last year (85,7%). There could be more goals in direct result of the set pieces if the pass was made with more speed and if our robots had more speed to escape the opponent robots.

7.2 Future work

The main concern for the next year should be making a kicking system which allows to kick close to ground and over the opposite goal keeper. With the new kicking system the *CatchBall* behavior needs to be improved so it is possible to catch the new pass without problems.

One major improvement that should be made is in the *Role Receiver*, instead of only evaluating the line of pass and signal the result through the coordination flag the Role should make the robot move if the line is blocked in order to be able to receive the pass. The behavior *kick* should also be improved to take into consideration the position of the goal keeper, some tests have been made with this approach that were not very conclusive probably due to the lack of accuracy in the directionality of the kicking system.

The formation should be improved so that not only the *Striker*, robot that chases the ball, as an active role. The defensive strategy should change in cases when a robot is overplayed.

In the sequence of this work more parameters should be added to the configuration file to allow a faster customization of the team. The CambadaConfig tool should be re-done using Qt libraries v4.0 and it could be merged into the Basestation.

The CAMBADA hardware structure needs to be continuously updated so it can be competitive, the kicking system is one of the essential parts. Another important part is the motors, actually CAMBADA robots are one of the slowest robots in the competition, with a top speed of 2 meters per second, contrasting against the 4 meters per second of the opposite teams.

One of the biggest challenges of the next year will be the change from a orange ball to regular ball, it is important to not only surpass this problem but at the same time try to make the improvements above described. Another improvement in the vision should be the use of a frontal camera to enable stereoscopic vision. This is specially important for the goal keeper so it can track the balls that come high, like CAMBADA kicks.

Bibliography

[1] *Bosch CAN home page*. http://www.can.bosch.com. Last visited: 21 of July 2009.

[2] Libxml home page. http://xmlsoft.org/. Last visited: 21 of July 2009.

[3] Qt software home page. http://www.qtsoftware.com. Last visited: 21 of July 2009.

[4] RoboCup 2008 home page. http://robocup-cn.org/. Last visited: 21 of July 2009.

[5] RoboCup 2009 home page. http://www.robocup2009.org/. Last visited: 21 of July 2009.

[6] Robótica 2009 home page. http://www.est.ipcb.pt/robotica2009/. Last visited: 21 of July 2009.

[7] W.H.T.M. Aangenent, J.J.T.H. de Best, B.H.M. Bukkems, F.M.W. Kanters, K.J. Meessen, J.J.P.A Willems, R.J.E. Merry, and M.J.G. v.d. Molengraft. TechUnited Eindhoven Team Description 2009. *CD proceedings of RoboCup 2009 Symposium*, 2009.

[8] L. Almeida, P. Pedreiras, and J. A. Fonseca. The FTT-CAN Protocol: Why and How. *IEEE Transactions on Industrial Electronics*, 49(2), December 2002.

[9] L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, and L.S. Lopes. Coordinating distributed autonomous agents with a real-time database: The CAMBADA project. In Tugrul; Korpeoglu Ibrahim Aykanat, Cevdet; Dayar, editor, *Proc. of the ISCIS*, pages 876–886. Springer, 2004. Lecture Notes in Computer Science, Vol. 3280.

[10] T. Amma, P. Baer, K. Baumgart, P. Burghardt, K. Geihs, J. Henze, S. Opfer, S. Niemczyk, R. Reichle, D. Saur, A. Scharf, J. Schreiber, M. Segatz, S. Seute, H. Skubch, S. Triller, M. Wagner, and A. Witsch. Carpe Noctem 2009. 2009.

[11] J. L. Azevedo, N. Lau, G. Corrente, A. Neves, M. B. Cunha, F. Santos, A. Pereira, L. Almeida, L. S. Lopes, P. Pedreiras, J. Vieira, D. Martins, N. Figueiredo, J. Silva, N. Filipe, and I. Pinheiro. CAMBADA2009: Team Description Paper. *CD proceedings of RoboCup 2009 Symposium*, 2009.

[12] José Luís Azevedo, Manuel Bernardo Cunha, and Luís Almeida. Hierarchical Distributed Architectures for Autonomous Mobile Robots: a Case Study. In *ETFA2007-12th IEEE Conference on Emerging Technologies and Factory Automation*, pages 973–980, 2007.

[13] T. Buchheim, U.-P. K ppeler, R. Lafrenz, M. Oubbati, H. Rajaie, M. Schanz, F. Schreiber, O. Zweigle, , and P. Levi. Team Description Paper 2005 CoPS Sttutgart. 2005.

[14] Gustavo Corrente. Architecture of control/coordination of robotic soccer team. Master's thesis, Universidade de Aveiro, 2008.

[15] Antonio DAngelo Enrico Pagello and Emanuele Menegatti. Cooperation issues and distributed sensing for multi-robotystems. 2006.

[16] Nelson Filipe. Individual and Coordinated Decision Functionalities for the CAMBADA team. Master's thesis, Universidade de Aveiro, 2008.

[17] Y. Kuniyoshi I. Noda H. Kitano, M. Asada and E. Osawa. RoboCup: The Robot World Cup Initiative. In *Proceedings of the first international conference on Autonomous agents*, pages 340–347. ACM New York, NY, USA, 1997.

[18] N. Lau, L. Seabra Lopes, G. Corrente, and N. Filipe. Multi-Robot Team Coordination through Roles, Positionings and Coordinated Procedures. In *Proc. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2009)*, 2009. to appear.

[19] N. Lau, L.S. Lopes, and G.A. Corrente. CAMBADA: Information Sharing and Team Coordination. *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008*, pages 27–32, April 2008.

[20] MSL Technical Committee 1997-2009. Middle Size Robot League Rules and Regulations for 2009, 2008.

[21] A.J.R. Neves, D.A. Martins, and A.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'2008*, pages 51–55, Aveiro, Portugal, April 2008.

[22] Luís Paulo Reis and Nuno Lau. FC Portugal Team Description: RoboCup 2000 Simulation League Champion. In Tucker Balch Peter Stone and Gerhard Kraetzschmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, Berlin, 2001. Springer Verlag Lecture Notes in Artificial Intelligence.

[23] Luís Paulo Reis, Nuno Lau, and Eugénio C. Oliveira. Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents. In Jan Wendler Markus Hannebauer and Enrico Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent System: From RoboCup to Real-World Applications*, Berlin, 2001. Springer Lecture Notes in Artificial Intelligence.

[24] Fernando Ribeiro, Ivo Moutinho, Pedro Silva, Carlos Fraga, and Nino Pereira. Vision, Kinematics and Game strategy in Multi-Robot Systems like MSL RoboCup. 2004.

[25] RoboCup Workshop Kassel 2008, http://www.ni.uos.de/fileadmin/user_upload/tribots/Research/Kooperation.pdf. *Tribots: Soccer Strategy*, 2008. Last visited: 21 of July 2009.

[26] F. Santos, L. Almeida, P. Pedreiras, L.S. Lopes, and T. 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 2004*, Lisboa, Portugal, 2004.

[27] F. Santos, G. Corrente, L. Almeida, N. Lau, and L.S. Lopes. Selfconfiguration of an Adaptive TDMA wireless communication protocol for teams of mobile robots. In *Proc. of the 13th Portuguese Conference on Artificial Intelligence, EPIA 2007*, Guimarães, Portugal, 2007.

[28] J. Silva. Sensor fusion and behaviours for the CAMBADA Robotic Soccer Team. Master's thesis, Universidade de Aveiro, 2008.

[29] H. Skubch, M. Wagner, and R. Reichle. A Language for Interactive Cooperative Agents. *http://carpenoctem.das-lab.net/dl/doc/ALICA.pdf*, 2009. Last visited on: 21 of July 2009.

[30] Peter Stone and Manuela M. Veloso. Task decomposition and dynamic role assignment for real-time strategic teamwork. In *ATAL 98: Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, pages 293–308, 1999.

[31] Bob van der Vecht and Pedro Lima. Formulation and Implementation of Relational Behaviours for Multi-Robot Cooperative Systems. 2004.

[32] Thilo Weigel, Jens steffen Gutmann, Markus Dietl, Er Kleiner, and Bernhard Nebel. Cs-freiburg: Coordinating robots for successful soccer playing. 2002.

[33] O. Zweigle, U.-P. K ppeler, T. R hr, K. H ussermann, R. Lafrenz, F. Schreiber, A. Tamke, H. Rajaie, A. Burla, M. Schanz, and P. Levi. CoPS Stuttgart Team Description 2007. 2007.