FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# High level coordination and decision making of a simulated robotic soccer team

**Miguel Augusto Pereira de Oliveira**

Master in Informatics and Computing Engineering

Supervisor: Doutor Luis Paulo Gonçalves dos Reis

Co-Supervisor: Doutor José Nuno Panelas Nunes Lau

December 2010

# High level coordination and decision making of a simulated robotic soccer team

**Miguel Augusto Pereira de Oliveira**

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Doutor António Augusto de Sousa

External Examiner: Doutor Artur José Carneiro Pereira

Supervisor: Doutor Luis Paulo Gonçalves dos Reis

9th December, 2010

# Resumo

O RoboCup é uma iniciativa internacional que proporciona diferentes competições com vista a promover a investigação em robótica e inteligência artificial.

O trabalho descrito nesta tese centra-se numa dessas competições – a liga de simulação 3D do RoboCup. Desenvolver uma equipa de robots humanóides capazes de jogar um jogo de futebol apresenta diversos desafios como a locomoção bípede, chutar a bola e a coordenação da equipa. Esta tese centra-se na coordenação da equipa de robots.

De modo a obter uma equipa de robots cooperantes, estes necessitam de ter o mesmo conhecimento do ambiente que os rodeia (estado do mundo). Este facto é particularmente importante em jogos de futebol devido às constantes mudanças no estado do mundo, quer dos jogadores quer da bola.

Primeiro foi desenvolvido um mecanismo de comunicação para atacar este problema. Através da comunicação, os agentes obtêm um conhecimento idêntico do estado do mundo e torna-se possível trabalhar em métodos não-triviais de coordenação de equipa.

De seguida, são apresentadas algumas melhorias no processo de decisão dos agentes. Estas são baseadas no uso de fluxos que valorizam poções no campo de acordo com a proximidade de marcar golo ou manter a bola afastada da baliza da equipa.

Após este trabalho, os agentes são capazes de seleccionar diferentes decisões em vez de irem sempre em direcção à baliza do adversário como acontecia anteriormente.

# Abstract

The RoboCup initiative provides several interesting competitions that foster the interest in the robotics and artificial intelligence research.

The work described in this thesis focus on one of those competitions – the RoboCup 3D simulation league. Developing a team of humanoid robots capable of playing soccer matches presents several challenges, from biped locomotion, kicking the ball to the coordination of the team. This thesis focused on latter.

In order to have a team of cooperating robots, they need to share the same knowledge about the environment (world state). This fact is particularly important in soccer matches due to the dynamic environment of the match where both the players and the ball are constantly moving.

First, a communication mechanism was implemented to tackle this problem. With communication, the agents have identical knowledge about the world state and it's possible to work in non-trivial methods for team coordination.

Then, some improvements in the agent's decision making process are presented. These are based on the use of fluxes that value positions on the field according to the proximity to scoring a goal or keeping the ball away from the team's goal.

After this work, the agents are able to select different decisions other than going directly towards the opponent goal (old approach).

# Aknowlegments

It is always difficult to point out all the people that helped me to produce the work presented in this thesis. Having said that, I would like to thank my supervisor Luis Paulo Reis for his ever-lasting patience and support. I also want thank my co-supervisor Nuno Lau for sharing his knowledge about this field and priceless help in technical issues. A special thanks to my family and friends that have been supporting me throughout life and this work in particular. On a final note, I want to thank my master's director, professor Augusto Sousa for his kind words and encouragement in the final step of this work.

Miguel Augusto Pereira de Oliveira

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| ADVCOM | Advanced Communications |
| DOF | Degree Of Freedom |
| HARL | Heuristic Accelerated Reinforcement Learning |
| IEETA | Institute of Electronics and Telematics Engineering of Aveiro (Instituto de Engenharia Electrónica e Telemática de Aveiro) |
| KBKR | Knowledge-Based Kernel Regression |
| LIACC | Artificial Intelligence and Computer Science Laboratory (Laboratório de Inteligência Artificial e Ciência de Computadores) |
| LRA | Locker Room Agreement |
| MCAP | Multi-Criteria Assignment Problem |
| ODE | Open Dynamics Engine |
| PTS | Periodic Team Synchronization |
| SBSP | Situation Based Strategic Positioning |
| SPADES | System for Parallel Agent Discrete Event Simulation |
| TROT-RL | Team-Partitioned Opaque-Transition Reinforcement Learning |

# Chapter 1

# Introduction

This thesis focuses on the improvement of a humanoid robotic soccer team's coordination and decision making. This work was applied in the FC Portugal 3D humanoid agent, which competed in the RoboCup 2010 3D simulation league.

## 1.1 Motivation

Intelligent robots have been used numerous times in science fiction series or films. In order to help to make the transition from fiction to reality, an international initiative called RoboCup was created to foster artificial intelligence and robotics research [1].

Since 2000, the FC Portugal project dedicates its research to the development of coordination methodologies applied to the RoboCup Simulation Leagues [2], achieving remarkable results throughout the past few years. These include several European and World championships.

In 2004, the 3D simulation league was created, extending the existing 2D simulation league and modelling the robots as spheres. Recently, the robots evolved to legged robots using humanoid models. This presents several research opportunities both in robotic locomotion and high level coordination of the teams.

This work focuses on the problem of coordinating a robotic soccer team in order to allow the team to play soccer matches.

However, it is important to note that although the focus of this work is a robotic soccer competition such as the RoboCup 3D simulation league, it is possible to extend it to other fields of multi-agent coordination, since most faced problems are general in robotics research.

## 1.2 Objectives

The objective of this work is to improve the coordination and decision making process of a robotic soccer team, namely, the 3D humanoid FC Portugal team. In order to achieve this major goal, first, it is necessary to obtain a uniform world state (knowledge about the environment in which the robot is operating) among the robots. Then, using this common world state, this work pretends to improve the decision making process achieving an effective teamwork.

## 1.3 Thesis outline

The remainder of this thesis is organized as follows.

Chapter 2 shows an overview of the current state of the art in multi-agent coordination and communication.

Chapter 3 presents the RoboCup initiative and RoboCup competitions related to the work described in this document.

Chapter 4 describes the simulation environment used in the RoboCup 3D simulation league in which this work was tested. The used agent's robot model and the 3D simulation league are also detailed.

Chapter 5 introduces the FC Portugal project and gives a brief description of the 3D humanoid agent code structure and decision making process.

Chapter 6 first analyses the communication system in the simulation environment and describes the improvements developed in this work. Then, it describes the coordination and decision making process used in the robotic soccer team and the problems faced during the development process.

Chapter 7 presents the conclusions about this work and proposes a few topics for future work.

# Chapter 2

# Coordination methodologies

Almeida, Lau and Reis [3] identified several coordenation methodologies for simulated robotic soccer teams.

## 2.1 Coordination issues

During a game a player can receive a lot of information at once, but trying to use it all is unfeasible due to the magnitude of the state space. The following main issues can difficult action coordination:

- Environment's unpredictability (e.g. unknown opponent's behaviour) makes it difficult to predict the world's next state;
- Conflicting behaviours/goals (e.g. due to different perceptions gathered from a subjective view of the world) between teammates [4];
- Inherent high complexity of most tasks (e.g. positioning) complicates implementation and modelling;
- Due to possible of communication failures, agents can't assume that all messages are exchanged and therefore can't rely on the other party;
- Low-bandwidth makes it difficult to convey significant knowledge in messages;
- Difficulty in building a global and reliable view of the environment;
- Uncertainty in perceived world information may lead to invalid state knowledge representations;
- Uncertainty in actuators may induce errors on action coordination, leading to them being executed different than expected;
- Agent heterogeneity introduces uncertainty about the player's modelling as their characteristics vary;

## 2.2 Coordination by communication

By sharing information about the world state, it's possible to achieve and maintain the coordination of the team. Nowadays, the competition constraints forbid long messages, so the agents must select carefully which information is vital to broadcast to the team. With the world state knowledge, the team can make better decisions. FCPortugal developed an Advanced Communications (ADVCOM) framework that created a separate communicated world state using only information from teammates, without any prediction or perception information from the player. The comparison between the player's communicated and perceived worlds allows him to assess the interest of each item of his perceived world state to his teammates and select the most useful information (e.g. players and ball positions) to transmit. The utility metrics for calculating the interest of the balls used domain-specific heuristics and was later on extended to accommodate the current situation and estimated teammate's knowledge [3, 5]. Recently, techniques based on player's beliefs of the world state were used (discarding team communication) to take into account player's intentions in the decision and control phases [6,7,8,9].

## 2.3 Intelligent perception

The information received by a robot from its sensors is limited and so it must be managed wisely. This allows gathering the most valuable information at each instant that can be used to enhance the accuracy of the player's world state and consequently enable strategic decisions. FCPortugal suggested an approach in which players can assume three types of visualizations during a soccer match [3, 10]:

- Active: look at the target location of a desired action (e.g. a pass to perform);
- Ball-centered: look at the ball to react rapidly to its sudden velocity changes (e.g. kick by a player);
- Strategic: look at a strategic location to improve the world's state accuracy and maximize the chance for success of cooperative actions with teammates using a Strategic Looking Mechanism (SLM).

## 2.4 Coordination for action selection

Deciding at a given moment what action the player should perform is very important in a soccer game. A player's individual decision typically depends on the actions performed (or expected) of other players and balances its possible risks and rewards. However, in this kind of dynamic environment these dependencies can change rapidly as a result of the continuously changing state and so efficient and scalable methods must be developed to solve this issue [3].

Several action selection mechanisms have been proposed throughout the years.

One of the first used player roles and a measurement level of how opponents could interfere in the current situation using a multi-layer perception [8] for this purpose.

Later on, Coordination Graphs (CGs) [9] was proposed on the assumption that in most situations only a small number of players need to coordinate their actions while the remaining are capable of acting individually (e.g. the ball owner coordinates his actions with nearby players). This mechanism has been widely adopted and several methods were applied to improve its efficiency (e.g. variable elimination [11], max-plus algorithm [12] and simulated annealing [13]).

Afterwards, an approach consisting of neuro-fuzzy systems and bidirectional neural networks was proposed to determine the probabilities and a priority based system which maps human knowledge to the action selection method.

More recently, a Case-Based Reasoning (CBR) [14] approach was proposed that explicitly distinguishes between controllable and uncontrollable indexing features, corresponding to the positions of the team members and opponent robots.

## 2.5 Coordination for behaviour acquisition

A soccer team often uses flexible predefined strategies (to some point), typically set on the LRA. However these strategies can prove fruitless, when playing against an opponent that exhibits an unknown and incompatible behaviour so as to defeat opponents. For this matter, modelling the opponent's behaviour becomes a necessity in order to allow a convenient adaptation. However, as most players' are unseen for some time and the received visual information may be incomplete or inaccurate this task becomes very challenging.

With adequate models of players' (teammates and opponents) behaviour, a player can improve his world model's accuracy and consequently improve decision-making (e.g. by anticipating collaborative needs of their teammates by positioning themselves in a useful way so that the ball carrier has several useful passing options) [3].

Machine learning techniques have been proposed to address the issue of player adaptation to unforeseen situations [15,16].

Layered learning [17] has been proposed to enable learning low-level skills and ultimately use them to train higher-level skills that can involve coordination. The highest layer of the previous approach uses a Team-Partitioned Opaque-Transition Reinforcement Learning (TPOT-RL) technique to allow a team of players to learn effective policies and thus cooperate to achieve a specific goal. This technique eases the learning task by dividing it among teammates, using coarse action-dependent features and gathering rewards directly from environmental observations. It is particularly suitable in this kind of domain that presents huge state spaces (lots of them hidden) and limited training opportunities.

Two other important subtasks of a soccer game, Keepaway and breakaways, have been used to study specific behavioural coordination issues [3]. Keepaway can be described as a game situation where one team (the keepers), tries to maintain possession of the ball within a limited region, while the opposing team (the takers) attempt to gain possession. BreakAway is a game played at one end of the soccer field with the purpose of the attackers trying to score goals against defenders. Reinforcement learning techniques have proven its usefulness to improve decision-making in these tasks [18,19].

Moreover, free-kick decision making for coordinating a kicker and a receiver was coped using policy gradient reinforcement learning [20,21].

The recognition of the potential for RL techniques, lead to the proposal of several methods to accelerate them:

- Giving advice about preferred actions using Knowledge-Based Kernel Regression (KBKR) and Preference-KBKR [18];
- Heuristic Accelerated Reinforcement Learning (HRAL): using heuristic information to accelerate (e.g. using Minimax-Q [22]) and Q-Learning [23];
- Case Based-HARL (CB-HARL): heuristics are derived from a case base (e.g. using Q-Learning [46].

## 2.6 Coordination for strategic actions

In a real soccer game, team strategies are usually rehearsed during mundane training of team players and applied during the game. Normally, a team follows the same strategies during every game, but for certain opponents they must be changed because they might not be suitable to their behaviour.

Strategies usually consist on a set of tactics composed by formations that map a strategic position and a distinguished role to each player which guides his behaviour [3].

To deal with the challenges of PTS domains a Locker Room Agreement [17] (LRA) was proposed, in which players consent on globally accessible environmental cues as triggers for changes in team strategy. This mechanism is useful in domains with reduced communication and was based in the definition of a flexible team structure based on roles, formations and set-plays. A timestamp was also included in the communication of team strategies so that players could recognize changes and always keep the most recent one to disseminate to others. The team's formation can be static or can change dynamically during the match on team synchronization opportunities (e.g. throw-in) or via triggered-communication where one teammate (e.g. captain) decides and broadcasts the decision to teammates.

Set-play s can be described as predefined plans for structuring a team's behaviour depending on the situation. A high-level generic and flexible framework that defines a language for set-play definition, management and execution was proposed for FCPortugal in 2007 [24]. A set-play in this framework involves participants (players or role references) and steps (states of execution) that can have conditions for execution. Each step is lead by a player (the current ball carrier who makes the most important decisions) and can have several transitions (possibly with conditions) that point to the next step. The main of a step transition defines a list of directives that includes the actions that should be executed (or not). The execution of a set-play requires a tight synchronization between all involved players to enable a successful cooperation and thus to cope with the simulator communication restrictions only the lead player is allowed to send messages [3].

## 2.7 Defensive coordination

The main goal of a defending team is to stop the opponent's team attack and to create conditions to launch their own. In general, defensive behaviours can be divided in two major skills marking and blocking (e.g. positioning to intercept the ball). Defensive positioning is a key aspect of the game, as players without the ball will spend most of their time moving somewhere rather than trying to intercept it.

Collaborative defensive positioning has been described as a Multi-Criteria Assignment Problem (MCAP) where n defenders are to be assigned to m attackers, each defender must mark at most one attacker and each attacker must be marked by no more than one defender [25]. The Pareto Optimality principle was applied to improve usefulness of the assignments made by simultaneously minimizing the required time to execute an action and the threat prevented by taking care of an attacker [26]. Threats are considered pre-emptive over time and their prevention is done using a heuristic-criterion that considers:

- The angular size of own goal from the predicted opponent's location;
- Distance from the opponent's location to own goal;
- Distance between the ball and opponent's location.

Marking is another defensive action, that can be described as the action of guarding a player to prevent him from advancing the ball towards the goal, making a pass or getting the ball from a teammate. The goal of this action is to intercept the ball and start an attack.

The opponent to mark can be chosen by the player (e.g. the closest opponent), by the team captain which can ensure that all opponents are marked, following a preset algorithm as part of the LRA [17] or by using matching algorithms [27].

In 2009, Bahia 2D [28] used a Neural Network trained with a back-propagation algorithm that uses a linear transfer function to decide the type of marking to perform based on the distance from the player to ball, number of opponents and teammates within the agent's field of view and the distance from the player to his own goal. Moreover, Fuzzy Controllers were used to decide if a player should mark an opponent.

## 2.8 Summary

This chapter presented an overview of the current state of the art in multi-agent coordination and communication with some examples of approaches used in RoboCup competitions.

# Chapter 3

# RoboCup

RoboCup [29] is an international project which aims to be a vehicle to promote robotics and Artificial Intelligence research, by offering a publicly appealing, but formidable challenge. Among the several RoboCup competitions, this thesis focuses on the RoboCup Soccer 3D Simulation League. The aim of the RoboCup soccer competitions is stated as follows:

*"By 2050, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official FIFA rules, against the winner of the most recent World Cup of human soccer"*

## 3.1 RoboCup soccer

Despite of being a soccer competition, in order to develop a team of robots able to play a soccer match, there are several general robotic research topics that must be explored and technologies that must be improved. Namely: design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real time-reasoning and sensor fusion [29].

Moreover, since soccer is one of the most popular sports in the world, it becomes another motivation to do research in this area and participate in these competitions.

### 3.1.1 Small size league

The Small Size League focuses on the problem of intelligent multi-agent cooperation and control in a highly dynamic environment with a hybrid centralized/distributed system [30].

The games are played on a green field that is 6.05m long by 4.05m wide between teams of five robots each, using an orange golf ball as the soccer ball. Each robot must fit within an 18cm diameter circle and must not exceed 15cm tall.

**Figure 1: Small size league kickoff**

### 3.1.2 Middle size league

The rules of the Middle Size League [31] are similar to the official FIFA rules [32], adapted when necessary to the robot players characteristics. The matches are played between two teams of up to five robots. The soccer ball complies with the official FIFA standard and there is a human referee.

The middle size robots' base must fit in a square of size 52x52 cm. Their height is at most 80cm and can not exceed 40kg. The robots are autonomous. Their sensors are onboard and they can use wireless connections to communicate. There is a coach that receives data from the robots and can send instructions to the team. However, no human interaction is allowed (except for substituting malfunctioning robots).

**Figure 2: Teams testing their robots**

### 3.1.3 Humanoid league

The Humanoid League is one of the most dynamically progressing leagues and is said to be the one closest to the 2050 goal [33].

In the Humanoid league, the robots are autonomous, have a human-like body plan and human-like sensors. Thus, these robots face the same challenges as humans in terms of world perception and modelling.

This league has three subcategories: KidSize (30-60cm height), TeenSize (100-120cm) and AdultSize (130cm and taller). In the KidSize class, teams are composed of three robots competing with each other. The TeenSize competition is similar to the previous but the teams are composed by two robots. In the AdultSize class, a striker robot plays against a goal keeper robot first and afterwards they play with exchanged roles against each other.

**Figure 3: Humanoid robot preparing a kick**

The main challenges in this league are: dynamic walking; running and kicking the ball while maintaining balance; visual perception of the ball, other players and the field; self-localization and team play.

### 3.1.4 Standard platform league

In the Standard Platform League, all teams compete with identical robots [34]. Since 2008, the robots used are the humanoid Aldebaran Nao [35].

**Figure 4: The Nao robot**

The games are played between two teams composed by three autonomous robots: one goalkeeper and two field players. Communication is only allowed among robots, using wireless connections [36].

Since all teams use the same robots, they can only focus on developing the best software they can.

### 3.1.5 Simulation league

In this league, the matches are simulated by software, using a client-server architecture. Each player is a computer process which communicates with the game server. The server is responsible to simulate the soccer match between two teams, using the official robot models. It receives the players' commands and sends them feedback.

**Figure 5: Overview of the 3D simulated soccer match**

There are three subcategories in this league: 2D, 3D and mixed-reality. As in the Standard Platform League, all teams use the same robots (robot models in this case). However, in simulation leagues, the robots can't be damaged (which can consume a considerable amount of the teams' budget) and the researchers can use methods like parallelization to reduce the optimization and testing time.

## 3.2 Summary

This chapter presented the RoboCup initiative, its objectives and some of its major soccer leagues. This work was based on the 3D simulation league and so this league will be more detailed in the next chapters.

# Chapter 4

# Simulation environment

The construction and maintenance costs of real robots can easily become major budget issues. This is the main advantage for researchers to use simulation instead of real robots. The idea of a simulation league is to develop a virtual agent capable of thinking and acting so that the acquired knowledge can be transferred to the real robots. To make this possible, it is necessary to construct accurate and reliable models of the real robots [37].

The RoboCup simulation league started with a 2D simulator, where the matches were played 11 versus 11 players and these were represented by circles. The competition evolved during the years and, in 2004, the 2D was extended to 3D (players were represented by spheres instead of circles) and the 3D simulation league was created. This evolution continues every year and nowadays, the 3D simulation league's players are humanoid robots and each team has up to 6 players.

## 4.1 Simspark

Simspark is a generic simulator for physical multi-agent simulations in three-dimensional environments. The goal of its authors was to create a flexible framework which facilitates exchanging components and extending the server [38].

Simspark is currently used as the simulator of the RoboCup 3D simulation league. The simulation system is composed by three distinct modules: the server, the monitor and the agents [39].

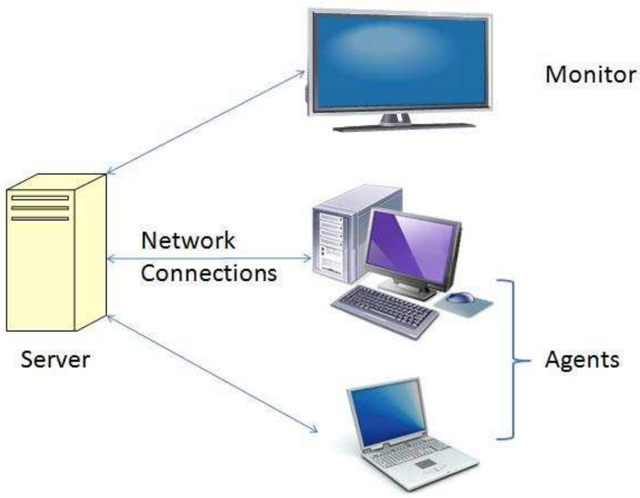


**Figure 6: Distributed simulation architecture**

### 4.1.1 Server

The Simspark server hosts the simulation process and is responsible for advancing the simulation. The simulation state is constantly evolving. Objects in the scene change their state, i.e. one of their properties like position, speed or angular velocity changes due to several influences. They are under the control of a rigid body physical simulation that handles every physical contact and forces applied in the objects. The agents also modify objects according the actions they perform using their effectors (actuators of the robots) [40].

As one can observe in figure 7, the server is composed by three main modules [37, 38]:

- **Physics Engine –** The physics engine is based on the Open Dynamics Engine (ODE) [41]. This module supports the simulation of the system dynamics and all physical interactions between objects.

- **Object / Memory Management –** The central component of the system is the Zeitgeist framework and provides access to the simulator's services. This framework takes care of object and memory management and is implemented using the C++ programming language and following the object-oriented paradigm.

- **Simulation Engine** – The simulation control is performed by the Simulation Engine, such as the main control loop and the communication with the agents. For scientific reasons, it is important to have a robust simulation system that ensures the simulation's reproducibility. However, in distributed systems, factors like computational power, network traffic, latencies and machine load are volatile and may influence the simulation's outcomes. Thus, to solve this problem, this module uses the System for Parallel Agent Discrete Event Simulation (SPADES) [42] middleware layer. SPADES takes care of event handling between the server and the agents and ensures the reproducibility of distributed simulations.

In the RoboCup 3D simulation league, the game time is discrete and divided in simulation cycles. On each cycle, the server sends information about the world state to each agent. Since this is a soccer match, this information includes the current result, game mode and game time. Information about the agents' perceptors (input sensors) is also provided. The agents are responsible to send their actions to the server in terms of their effectors' changes.

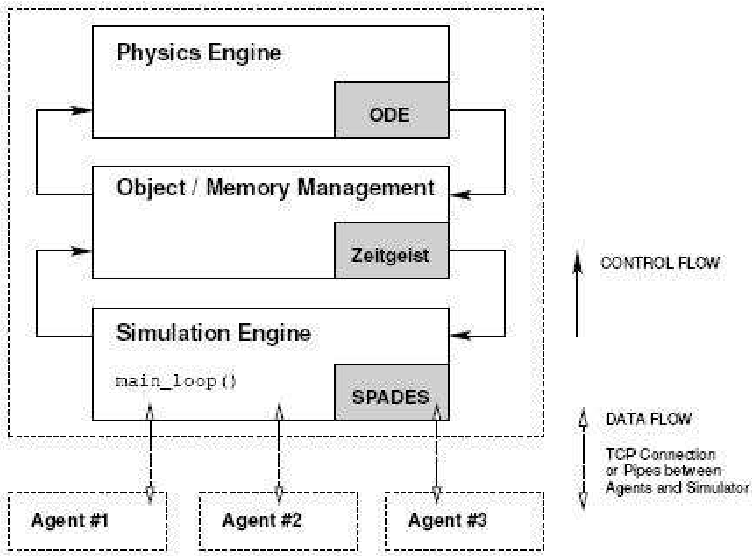


**Figure 7: Simspark server architecture. Source [38]**

### 4.1.2 Monitor

The Simspark monitor is a graphical interface that connects to the Simspark server and renders the current simulation [40]. On another hand, it is also possible to use the server logs and replay a past simulation.

In the RoboCup 3D simulation league, the monitor acts as a television broadcast of a soccer match. It shows the team names, game result and game time. It also provides several shortcut keys to change camera views, drop the ball and other commands (as long as they are supported by the server) [37, 40].


**Figure 8: A print screen of a kick-off**

## 4.2 Agents

Simspark provides a few sample robot models. These models are defined in a language similar to LISP [43]. This way, it's not necessary to recompile the simulator every time one needs to change an existing robot model or add a new one [37].

The RoboCup 3D simulation league uses a model of the Aldebaran Nao robot [35].

### 4.2.1 The simulated Nao robot

The simulated Nao is intended to be as similar as possible with the real Nao robot (section 3.1.4). Its height is 57cm, it has about 4.5kg and 22 Degrees Of Freedom (DOF): 2 on the neck, 4 on the arms and 6 on the legs. These 22 DOF correspond to 22 hinge joints (a hinge joint is a simple joint with one DOF). With the goal of the RoboCup initiative in mind, unnatural movements are inadvisable, so the joints' movements are limited.


**Figure 9: The real Nao and the simulated version. Source [40]**

The Nao model has several effectors and perceptors [40]:

- **Gyroscope and accelerometer –** both located at the torso. They provide information about radial and axial movement in the three dimensional space
- **Force resistance perceptor –** located in each foot, indicates the actual pressure on the foot. It can be used to determine if there's contact with any obstacle (e.g. ground or other robots).
- **Restricted vision perceptor –** provides visual information about the world.
- **Say effector and hear perceptor –** used to communicate with or receive messages from other robots, respectively.
- **Joint effectors and perceptors –** each joint is handled with the corresponding hinge joint effector and its state is perceived with the corresponding hinge joint perceptor.
- **Game state perceptor –** provides information about the game state and game mode.

Figure 10 and Table 1 present more detailed information about the simulated Nao joint configuration and movement limits.

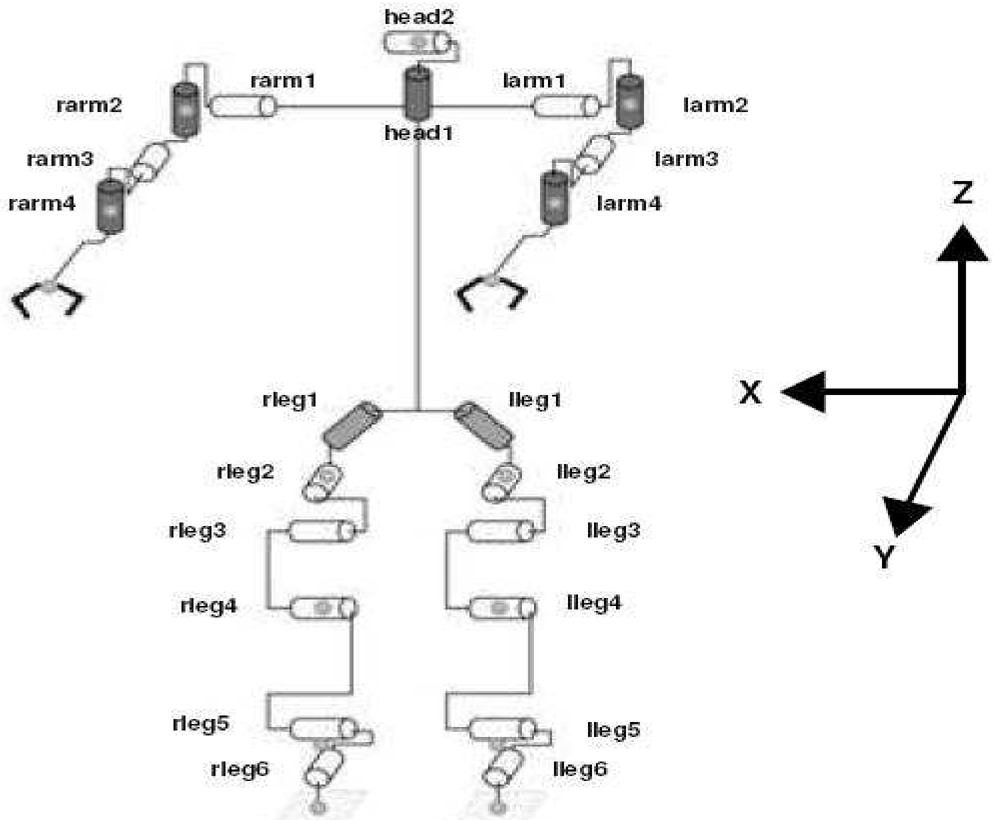


**Figure 10: Simulated Nao joint configuration. Source [37]**

**Table 1: Simulated Nao joint configuration. The l and r prefixes of leg and arm joints represent the left and right side, respectively, and were omitted for readability. Source [37].**

| Joint name | Joint type | Parent | Rotation axis (X,Y,Z) |
|---|---|---|---|
| head1 | Hinge | Neck | (0,0,1) |
| head2 | Hinge | Neck | (1,0,0) |
| arm1 | Hinge | Shoulder | (1,0,0) |
| arm2 | Hinge | Shoulder | (0,1,0) |
| arm3 | Hinge | Shoulder | (0,0,1) |
| arm4 | Hinge | Elbow | (1,0,0) |
| leg1 | Hinge | Hip | $(-\sin(\frac{\pi}{4}), 0, \sin(\frac{\pi}{4}))$ |
| leg2 | Hinge | Thigh | (0,1,0) |
| leg3 | Hinge | Thigh | (1,0,0) |
| leg4 | Hinge | Knee | (1,0,0) |
| leg5 | Hinge | Foot | (1,0,0) |
| leg6 | Hinge | Foot | (0,1,0) |

## 4.3 RoboCup 3D simulation league details

In 2010, the 3D simulation league matches were played between two teams composed by up to six players (at most one goalkeeper and any number of field players) on a field 18m long and 12m wide. The games were divided in two halves of five minutes each.

Regarding the work described in this thesis, it is important to note that the perceptors and effectors are limited, in order for the simulation to be as close to reality as possible. For example, the vision perceptor is restricted to a 120 degree range and it also has latency.

## 4.4 Summary

This chapter presented the simulation environment of the RoboCup 3D simulation league. The architecture of simulation environment, Simspark, was presented and was followed by a description of the simulated Nao, the robot model used in the competition. Finally, a few details and constraints present in the 3D simulation league were discussed.

# Chapter 5

# FC Portugal

In this chapter, the FC Portugal project will be presented, the team, the research areas and achievements. Then, the FC Portugal agent that competed in RoboCup 2010 will be analysed.

## 5.1 FC Portugal project

FC Portugal is a joint Project between IEETA/University of Aveiro and LIACC/University of Porto [44]. The team started to compete in RoboCup 2D simulation league in 2000. Afterwards, FC Portugal also competed in other competitions like the 3D simulation league, coach competition, mixed reality league, rescue simulation, among others. In the future, the team will probably extend their efforts to the standard platform league.

Some of the main research interests are [2,44, 45]:

- Communication in PTS Domains for Coordinating Teams of Autonomous Agents
- Creating accurate world states for intelligent agents
- Intelligent perception and sensor-fusion
- Multi-agent collaboration and communication
- Soccer, Game Analysis, Strategic Reasoning and Tactical Modeling

The project research work has achieved very good results, including several european and world championship awards. A few examples are:

- European RoboCup 2010, Bremen, Simulation 3D League, 3rd place
- European RoboCup 2010, Bremen, Simulation 2D League, 3rd place
- European RoboCup 2007, Hannover, Simulation 3D League, Champions
- RoboCup 2006, Bremen, Simulation 3D League, Champions
- European RoboCup 2006, Eindhoven, Rescue Sim. League, Champions

- RoboCup 2002, Fukuoka, Coach Competition League, Champions
- German Open 2001, Paderborn, Simulation League, Champions
- RoboCup 2000, Melbourne, Simulation League, Champions
- European RoboCup 2000, Amsterdam, Simulation League, Champions

## 5.2 FC Portugal agent

In this section, the structure of the FC Portugal agent (fcpagent) will be analysed. This section is based on [45]. The agent is implemented using the C++ programming language and the object-oriented paradigm.

### 5.2.1 Agent structure

The agent structure is divided in seven main parts: six packages and a central module (FCPAgent) that runs the agent main loop and makes decisions based on the information provided by the other modules.

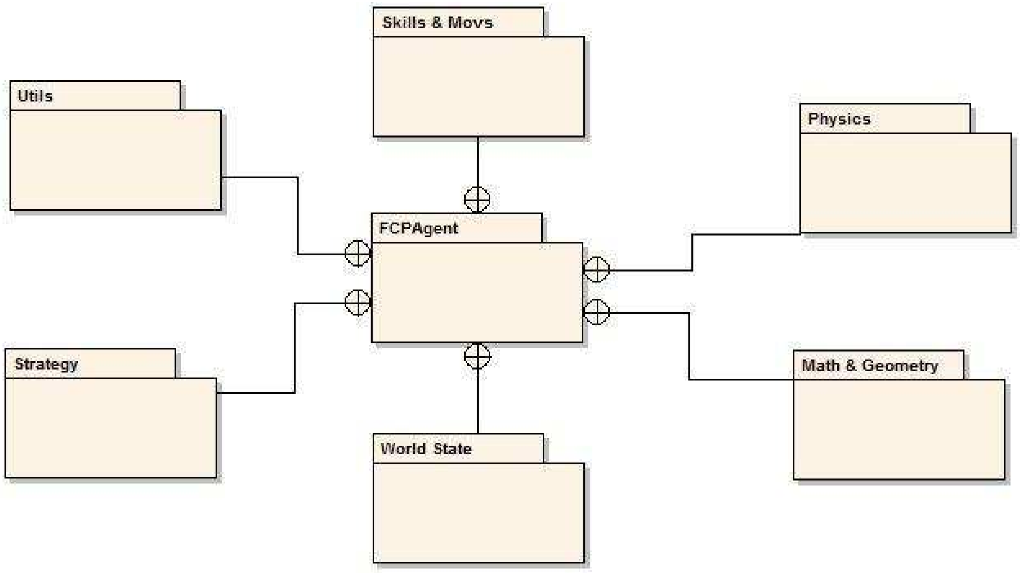


**Figure 11: FC Portugal agent structure**

The six packages are:

- **Physics –** contains information about the robot model currently in use.
- **Math & Geometry** – contains classes related to math and geometry which support non-trivial geometric and mathematical operations (e.g. 2D and 3D vector operations, line-point and segment-point distance, line-circle intersection, etc).
- **Skills & Movs –** supports the agent's low level skills such as walking, kicking, getting up, etc.
- **Strategy** – contains the high level Decision Making Unit (DMU) and is responsible to analyse what are the best actions for the agent, depending on its goals.
- **Utils –** the utilities package contains useful classes and the module responsible for the communication with the server (both build and send messages and parse the received server messages).
- **World State –** the world state is the most complex package. It manages all information about the environment and its information is crucial to the other packages. It holds information about the game (e.g. field dimensions, goal position), game state (e.g. play mode, current time), and other relevant objects (e.g. own location, ball, teammates and opponents positions).

### 5.2.2 Basic control flow

The fcpagent follows a very simple control flow as shown in figure 12. When the process is started, it initializes the agent and communicates with the server (1). Then, the agent enters the main loop. It waits the reception of a simulator message, parses it (2) and then process the information contained in the message and decides what action the agent should perform (3). Finally, the action is sent to the server and the agent goes back to step 2 (4).

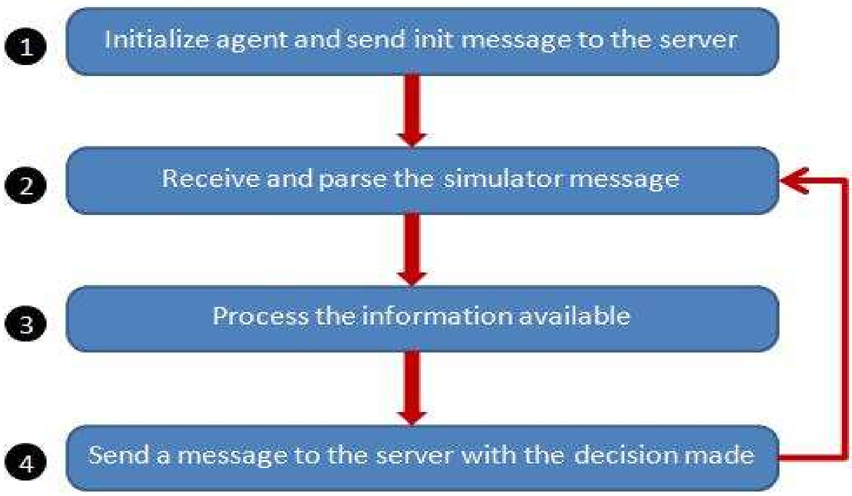


**Figure 12: Agent's basic behaviour**

### 5.2.3 Team strategy

One of the major research interests of the FC Portugal team is the strategic reasoning of the players and the research lead to some breakthroughs in this subject [46]. The Situation Based Strategic Positioning (SBSP) method uses the current situation to determine the best strategic position for the player. This positioning depends on the team's tactic, formation and their positioning in the formation. Sometimes, the agents can be far from their ideal strategic position due to several reasons. Thus, the Dynamic Positioning and Role Exchange (DPRE) method provides a mechanism for the agents to exchange roles if this action improves the team global utility (in terms of strategic positioning).

SBSP and DPRE are important as they keep the team formation organized and enables the strategy module to change formations easily in order to be more aggressive in the attack or more conservative in defensive formations as the match result requires.

## 5.3 Summary

This chapter presented the FC Portugal project and the structure of the FC Portugal humanoid agent, competing in the RoboCup 3D simulation league. The work developed in the scope of this thesis focus on some parts of the agent structure and these parts will be explored in more detail in the next chapters.

# Chapter 6

# Communication and coordination

## 6.1 Communication methodologies

Communication between the players is intended to share information about their environment and lead to a better world state. An effective communication system leads to a common world state among the agents. As the agents have basically the same world state, the team' coordination is improved indirectly because an agent can predict the teammates' decisions and act accordingly. It can also improve directly the team coordination if the agents broadcast their decisions to their teammates.

### 6.1.1 Old approach - no communication

Previously, no communication was used among the robots of the FC Portugal 3D simulation team. Thus, the agents' world state was based in the vision sensor and the SBSP system [46]. Every information about the objects contained in the world state (e.g. ball position, ball velocity, player orientation) has an associated confidence value. In each simulation cycle, if the object is seen, the information is updated accordingly, otherwise its confidence value is decreased. When the confidence on the information about other players is too low, the agent expects that the robot (either teammate or opponent if the team knows in advance the opponent formation in advance) is positioned around its strategic position (given by the SBSP system), so it updates this position to the strategic position with a low confidence value.

### 6.1.2 Implemented approach

The first step of this work was to implement an effective communication system that would improve the agent's world state information. The simulator has a few restrictions regarding communication:

- In each simulation cycle, only one agent is allowed to speak (send a message to the teammates);
- This message can have at most 20 bytes, using only printable characters, excluding a few special symbols (this leads to about 90 available characters).

This approach complements the old method described in the previous section. If a player does not receive information about other players, the old approach is used to obtain an estimate of their current positions.

### Message information

The messages sent by the agents must have relevant information to contribute to a better world state. This information includes data about the robot that is sending the message (usually the robot has a very high confidence on its position and orientation), ball position and velocity and information about other players such as position, team (either teammates or opponents).

Every value has an associated confidence value as explained in the previous section. When an agent receives a message, it checks if the communicated values have higher confidence values than its own.

Figure 13 shows the class definition of the AgentMessage class which models a message sent by and agent.

```
struct PlayerInfo {
    int         id;
    int         team;
    int         confidence;
    Vector3f    position;
};

class AgentMessage
{
    private:
            int                  msgType;
            int                  selfId;
            Vector3f             selfPosition;
            float                selfOrientation;
            Vector3f             ballPosition;
            Vector3f             ballVelocity;
            int                  ballConfidence;
            vector<PlayerInfo>   playersInfo;

            int                  passPlayer;
            Vector*              passPosition;

    public:
            // constructor and desctructor

            // set and get methods
};
```

**Figure 13: The AgentMessage class definition**

### Agent communication class

The agent communication class (AgentComm) is responsible for encoding and decoding agent messages.

Since the messages can only have 20 bytes and there are about 90 available chars, it was decided to use 64 chars (6 bits of each byte, $64 = 2^6$), in order to simplify the encoding and decoding process. This implies that the message can have at most 120 bits (6 bits * 20 characters) with information.

As in real soccer matches, the agents hear messages sent by the opponents. Since the message format is specific to each team, it's better to ignore these messages and only analyse messages from teammates. This lead to the implementation of a simple message authentication mechanism: the actual messages are preceded by an initial stamp symbol and followed by a hash value. The authentication reduces the number of available bits but ensures that no messages from other teams are decoded which would most likely compromise the quality of the agents' world state.

Due to the communication constraints, every floating point value must have a precision known by all agents (e.g. the player position is truncated with 1 decimal place, so if a position of a player is [5.56; 6.23], the position sent in the message would be [5.5; 6.2]). It is also necessary to normalize values with negative values. So, if a value has the range [-offset, offset], it's transformed into the range [0, 2*offset] during the message encoding and restored into the original value in the decoding method.

```
class AgentComm
{
    public:
            string encode(const AgentMessage & msg);

            AgentMessage* decode(const string & msg);

    private:
            bool checkAuthentication(const string & msg);

            void insertVal(int val, int & start, int nbits );

            int getVal(int & start, int nbits);

            int normalizeValue(float value, float offset, int precision);

            char            msgBuffer[MAX_MSG_BITS];
            unsigned int    nAllowed;
            unsigned short  decodechar[256];

            float           xPosOffset;             // offsets in communications
            float           yPosOffset;
            float           xVelOffset;
            float           yVelOffset;

            static const string allowed;
            static const string otherAvailableChars; // available chars other than letters and digits
};
```

**Figure 14: The AgentComm (Agent Communication) definition**

**Example of a message**

Based on the previous description of the possible agent messages, a possible message type could be:

- Initial stamp = 6 bits
- Message type = 3 bits
- Ball information
  - Position x, y (8+7) = 15 bits
  - Confidence = 3 bits
- Speaker information
  - Position x, y, z (8+7+1) = 16 bits
  - Orientation = 6 bits
  - Confidence = 3 bits
- Number of other players = 2 bits
- Other players information blocks
  - Position x, y, z (8+7+1) = 16 bits
  - Team = 1 bit
  - Confidence = 3 bits
- Hash value = 6 bits

The initial stamp and the hash value are related to the message authentication.

The message type identifies the following message structure. Each information is associated with a confidence value. The 3 bits assume that the confidence is given as a multiple of 10 in the range from 30 to 100% as there is no use of including information with close to zero confidence.

In this case, the positions (x, y) of the ball and the players are given with one decimal place. The z coordinate just informs if the player is standing up or fell on the floor. Since the field size is 18x12 meters, 8 and 7 bits are used for the x and y coordinates, respectively. 8 bits give 256 different values, providing the range [-12.8, 12.7] which is larger than the [-9, 9] range needed for the x coordinate. Similarly, 7 bits give 128 different values, providing the range [-6.4, 6.3] which is larger than the [-6, 6] range needed by the y coordinate.

Using 6 bits for the player orientation, its resolution would be about 6 degrees (360 degrees divided by 64 possible values).

The "number of other players" mentions the number of blocks with information about other players, whether they are teammates or opponents (team field).

### World state update and communication mechanism

At the beginning of each simulation cycle, the agents' world state is updated with the information received by the server. This process is implemented in an "update" function. This function starts by updating the positioning of the agent and the ball with the vision sensor information. Then, it updates the information about the other agents (teammates and opponents).

Whenever the agent receives a message from other agent, the world state is updated with the information decoded by the AgentComm.

```
class WorldState
{
    public:
        void update();

        // related to communication
        void updateWorldStateWithComm(AgentMessage * message);
        bool canCommunicate();
        void initCommunication();
        void updateNextToComm();

        // ...
```

**Figure 15: World state update functions**

Since there can only be one agent sending a message each cycle, the agents wait on a queue until they're allowed to send a message. The first agent sends a message, then the second, third and so on. When the last agent sends a message, the first will send a message again and the process continues in this cycle (see figure 16).

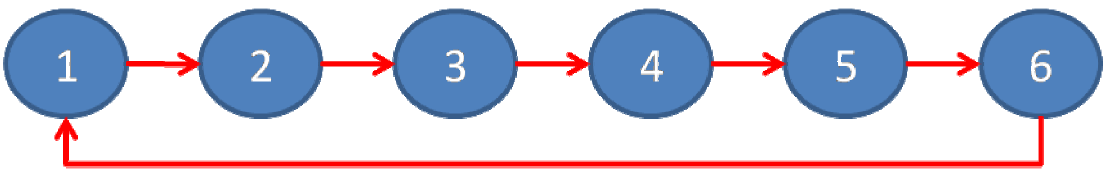


**Figure 16 : Communication cycle**

In order to check if an agent can communicate, one could be tempted to check which agent sent the previous message and communicate if the current agent is the next in the queue. However, network messages may be lost and this could disrupt the communication system (if a message was lost, no agent would communicate again). So, it was chosen a method based on the game time (sent by the server). The game time is not influenced by this kind of network problems and guarantees that the right agent will try to communicate throughout the simulation.

## 6.2 Strategy definition

The team strategy is configured in a text file. The header of the file contains the strategy type (in other words, the league in which this strategy will be used) and the number of team tactics, players, player types, formations, flux matrices and set plays. The usage of simple text files is particularly useful for testing since there's no need to recompile the code every time one wants to change something in the strategy.

.

### Tactics section

Special team tactics are used depending on the result and the current game time. For example, if the team is losing by a few goals, it should change into a more offensive formation. On the contrary, if the team is winning near the end of the game, it may choose a more defensive formation in order to secure the victory. A sample of the tactics section is presented in figure 17.

```
2 2 # Number of Time Tactics and opponents - tactic depends on result, time and opponent
000 399   0  1 1 1 1 1     1  1 1 1 1 1  # Opp Numb - Losing Bad, Losing, Drawing, Winning, Winning Bad
400 600   0  1 1 1 1 1     1  1 1 1 1 1
# Tactics Definition ---------------------------------------------------------------------------
1 # Tactic 1 - Tactic Description - Exp1
4 1 1   0.5 0.3 0.2    0.5 0.5 # Formation, Flux, SetPlans, WFlux, WSafe, WEasy, WPass, WDrib
4 4   4 4   4 4   6 5   4 4  4 4 4 4 4 4 4 4 # Form used in each situation
                                             # (Att/Def, KickOff(O/T), CornKickIn, FKick, GFKick, Pen )
2 # Tactic 2 - Tactic Description - Exp2
4 1 1   0.4 0.4 0.2    0.5 0.5 # Formation, Flux, SetPlans, WFlux, WSafe, WEasy
0 # No Form used in each situation (Att/Def, KickOff(O/T), CornKickIn, FKick, GFKick, Pen
```

**Figure 17: strategy file – tactics section**

### Player types section

In this section, different player types are defined. A player type is defined by its attraction to the ball in the x and y coordinates whether the player should be always behind the ball (usually defensive players), the square of the field that the player should take and the decision algorithm used. A sample of player type's definition is shown in figure 18.

```
# Player Types Definition ------------------------------------------
  1    2    3    4    5    6   # PT Number
 0.0  0.7  0.75 0.6  0.6  0.6  # AttractionX
 0.3  0.7  0.7  0.4  0.4  0.4  # AttractionY
 0.0  0.2  0.2  0.2  0.2  0.2  # AreaAttrAttack (last 1/4)
 0.0  0.2  0.2  0.2  0.2  0.2  # AreaAttrDefense (last 1/4 of field)
  1    1    0    0    0    0   # BehindBall
-8.7 -9.0 -9.0 -9.0 -9.0 -9.0  # MinX
-5.0  8.5  8.5  8.5  8.5  8.0  # MaxX
-2.0 -6.0 -6.0 -6.0 -6.0 -6.0  # MinY
 2.0  6.0  6.0  6.0  6.0  6.0  # MaxY
  1    2    2    2    2    2   # Decision Algorithm 1-Goa, 2-Def, 3-Att
```

**Figure 18: strategy file – player types section**

### Formations section

A formation definition starts with the formation number and the type of the formation (based on SBSP or Delaunay triangulation).

Then, it's defined for each player its base position on the field and its player type (see figure 19).

```
# Formations Definition ----------------------------------------
1 1 # 2-1-2 Formation 1 - Type 1 - SBSP
-8.7 -4.0 -4.0 -1.0  2.0  2.0   #Posx
 0.0 -0.8  0.8  0.0 -1.5  1.5   #Posy
  1    2    2    3    3    3    #Type PT_GOA PT_MID PT_DEF...
2 1 # 2-1-2-1 test Formation 2 - Type 1 - SBSP
 2.5 -3.0 -3.0 -1.0  1.0  1.0   #Posx
 0.5 -1.0  1.0 -0.5 -2.0  2.0   #Posy
  5    5    5    5    5    5    #Type PT_GOA PT_MID PT_DEF...
3 2 # Formation 3 - Type 2 - Delaunay - Formation description 3
formations/normal_formation_3D_3.conf # Name
  1    2    2    3    3    3    #Type PT_GOA PT_MID PT_DEF...
4 2 # Formation 4 - Type 2 - Delaunay - Formation description 4
formations/normal_formation_3D_3.conf # Name
  1    2    2    3    3    3    #Type PT_GOA PT_MID PT_DEF...
```

**Figure 19: strategy file – formations section**

### Flux matrices section

The flux matrices contain the value of each portion of the field. This value is influenced by the easiness to score a goal from that portion of the field, expected player density in that portion or danger to concede a goal (close to the team's goal). A sample of the flux matrices section is presented in figure 20.

```
# Flux Matrices Definition --------------------------------------------------
1 1 # Flux 1,Type 1 (old) - Types:Norm 1, Del 2 - Flux Description - Normal
 00 20 30 40 50 50 50 50 30
 00 20 30 40 50 60 60 60 50
 00 10 20 30 50 60 70 80 75
 00 00 20 30 50 60 70 90 100
 00 10 20 30 50 60 70 80 75
 00 20 30 40 50 60 60 60 50
 00 20 30 40 50 50 50 50 30
2 1 # Flux 2,Type 1 - Flux Description - Test go back with ball
100 90 80 70 50 40 30 20 0
 90 80 70 60 40 30 20 10 0
 80 60 50 40 30 20 10 00 0
 70 60 40 30 20 10 10 00 0
 60 40 20 20 10 10 00 00 0
 20 20 10 10 10 00 00 00 0
 10 00 00 00 00 00 00 00 0
3 2 # Flux 3,Type 2 - Delaunay - Flux Description
flux_normal_3D.conf # Name
4 2 # Flux 4,Type 2 - Delaunay - Flux Description
flux_normal_3D.conf # Name
```

**Figure 20: strategy file – flux matrices section**

### Set plays section

This section contains the definition of set plays that may be used by the team (see figure 21).

```
# SetPlays Definition ("setplay.conf")-----------------------------------------
1 2 # Set Number and Type (1 - Old Setplays, 2 - New Setplays) - Set Description
1 2 3 4 5 6 7 8
2 2 # Set Number and Type 2 - New Setplays - Set Description
1 2 3 4
3 2 # Set Number and Type 2 - New Setplays - Set Description
1 2 8
```

**Figure 21: strategy file – set plays section**

### General domain parameters section

In the final section, there are a few general parameters of the game such as the game time, extra time, field size and agent skills. A sample of the tactics section is presented in figure 22.

```
# General Domain Parameters
600 300 # Game Time, Extra Time
18 12 # Field Size
1.8 3.9 2.1 # Penalty Area Size and goal width
3.0 0.6 0.3 # max kick distance, running speed/sec, agentsize
```

**Figure 22: strategy file – general domain parameters section**

## 6.3 Decision making process

### 6.3.1 Old approach

Previously, the players just tried to take the possession of the ball and then head to the opponents' goal and score. The robot's kicking ability wasn't stable and reliable, so passing the ball was not an option.

So, basically, the agent tried to intersect the ball by blocking the ball's current trajectory or reach the ball before any other agent if it's not in any robot's possession.

### 6.3.2 Using flux

First of all, it is important to note that developing high level skills such as the team's coordination and decision making can only be possible when the low level skills are stable, reliable and provide the required behaviours. For example, previously, it didn't make sense to consider passing or shooting to the goal since the agent couldn't kick the ball properly. Other movements as dribbling or approaching the ball (go to ball) are also very important.

The implemented approach uses the strategy definitions of the strategy file. When the agent has the ball, it considers the actions of dribbling, passing the ball to a teammate and shoot on goal.

In order to evaluate which is the best option, the decision making process uses three concepts:

- **Flux** – these are the values defined in the flux matrices contained in the strategy file;
- **Easiness** – the easiness value evaluates, given the current circumstances, how hard is the action for the agent. In other words, it approximates the probability of success of dribbling (being able to control the ball) or kicking towards the goal or a passing position.
- **Safety -** the safety value analyses the positions of the players on the field and checks if the opponent may intersect the ball.

For each possible action, the agent evaluates its utility and selects the action with the highest utility.

## 6.4 Conclusions

This chapter presented the work done on the communication and coordination of the FC Portugal humanoid agent. Through debugging, it was seen that the implementation of an effective communication mechanism lead to a significant improvement of the agents' world state quality. This allowed to work on high level coordination and decision making. Using the concepts of Flux, Easiness and Safety, the agents are capable of selecting the best action even if it involves moving away from the opponents' goal in order to escape the opponent defenders or conceding a corner to move the ball away from the team's goal.

# Chapter 7

# Conclusions and future work

## 7.1 Conclusion

The main focus of this thesis was to improve a humanoid robotic soccer team's coordination and decision making process, applying the work in the FC Portugal 3D humanoid agent which competed in the RoboCup 2010 3D simulation league.

As seen in chapter 3, the RoboCup initiative organizes interesting competitions in order to foster robotics and AI research. The robotic soccer leagues provide several interesting research problems that are general to the robotics field. Soccer provides an extra motivation since it's one of the most popular sports in the world.

The simulation platforms provide an opportunity to develop and test high level methods as the teams don't have to worry about constructing the robots or fix them when damaged. The robots used in the simulation leagues are always approximations of real robots, but these approximations are already close to reality. So the work developed in the simulation leagues can be later applied in the real robot leagues.

It is essential to have solid and reliable low level skills such as different types of walking and kicking in order to be able to work on high level skills like team coordination. One of the most frequent problems faced throughout this work was the frequent change of the low level skills which usually had unexpected effects in other low level skills and making it difficult to develop more complex high level approaches.

The use of communication among the multi-agent team significantly improved the quality of the agents' knowledge about their environment and allowed the agents to have an almost identical world state. This improvement made it possible to work on high level coordination and decision making skills. The concepts of Flux, Easiness and Safety showed to cover the main issues involved in the decision making of a robotic soccer player.

This work contributed to the FC Portugal team that competed in the RoboCup German Open 2010 and the RoboCup 2010 3D simulation leagues. In the European competition, the team achieved 3rd place and in the World championship the team obtained an honourable 5th place.

## 7.2 Future work

Although this work proved to achieve good results in the RoboCup competition, experiments should be done to validate these approaches.

The formal definition of a common world state, based on the information sent by communication could allow the development of more intelligent communication methods (instead of having predefined queue where all agents communicate in a cycle).

The use of fluxes still presents many challenges. The evaluation of the flux, easiness and safety can be further improved and it is suggested to run experiments with parameters, against different types of teams.

# Bibliography


[1] I. Noda, et al., "RoboCup as a Strategic Initiative to Advance Technologies," in Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC), 1999, pp. 692-697.

[2] Luis Paulo Reis, Nuno Lau, Hugo Picado, and Nuno Almeida. Paper Contributions, Results and Work for the Simulation Community of FC Portugal, March 2008

[3] Fernando Almeida, Nuno Lau and Luis Paulo Reis, "A Survey on Coordination Methodologies for Simulated Robotic Soccer Teams", RoboCup Symposium 2010, Singapore

[4] J. Penders, "Conict-based Behaviour Emergence in Robot Teams," in Conicting Agents: Conict Management in Multi-Agent Systems, ed Norwell: Kluwer Academic Publishers, 2001, pp. 169-202

[5] Luis Paulo Reis, Rui André Ferreira, Nuno Lau, "Situation Based Communication for Coordination of Agents," in FEUP Edições, 2004, pp. 39-44.

[6] J. R. Kok, et al., "Non-Communicative Multi-Robot Coordination in Dynamic Environments," Robotics and Autonomous Systems, vol. 50, pp. 99-114, Feb 2005

[7] F. Stulp, et al., "Implicit Coordination in Robotic Teams using Learned Prediction Models," in International Conference on Robotics and Automation, ed New York: IEEE, 2006, pp. 1330-1335.

[8] E. Pagello, et al., "Cooperative behaviours in multi-robot systems through implicit communication, "Robotics and Autonomous Systems, vol. 29, pp. 65-77, Oct 1999.

[9] R. Rico, et al., "Team Implicit Coordination Processes: A Team Knowledge-Based Approach," Academy of Management Review, vol. 33, pp. 163-184, Jan 2008.

[10] L. P. Reis, "Coordination in Multi-Agent Systems: Applications in University Management and Robotic Soccer," PhD, Electrical and Computers Engineering, FEUP, Porto, 2003.

[11] H.-S. Kim, et al., "Action Selection Mechanism for Soccer Robot," presented at the Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997

[12] J. R. Kok, et al., "Multi-Robot Decision Making Using Coordination Graphs," in In Proceedings of the 11th International Conference on Advanced Robotics, Coimbra, Portugal, 2003, pp. 1124-1129.

[13] W. Jin, et al., "Multi-robot decision making based on coordination graphs," in Mechatronics and Automation, 2009. ICMA 2009. International Conference on, 2009, pp. 2393-2398.

[14] J. Dawei and W. Shiyuan, "Using the Simulated Annealing Algorithm for Multiagent Decision Making," in RoboCup 2006: Robot Soccer World Cup X, ed: Springer-Verlag, 2007, pp. 110-121

[15] R. Zafarani and M. R. Yazdchi, "A Novel Action Selection Architecture in Soccer Simulation Environment Using Neuro-Fuzzy and Bidirectional Neural Networks," International Journal of Advanced Robotic Systems, vol. 4, pp. 93-101, March 2007.

[16] R. Ros, et al., "A Case-based Approach for Coordinated Action Selection in Robot Soccer," Artificial Intelligence, vol. 173, pp. 1014-1039, 2009.

[17] T. Andou, "Refinement of Soccer Agents' Positions Using Reinforcement Learning," presented at the RoboCup-97: Robot Soccer World Cup I, 1998.

[18] A. Agah and K. Tanie, "Robots Playing to Win: Evolutionary Soccer Strategies," in International Conference on Robotics and Automation - Proceedings. vol. 1-4, ed New York: IEEE, 1997, pp. 632-637.

[19] P. Stone, "Layered Learning in Multi-Agent Systems," PhD, Computer Science, Carnegie Mellon University, Pittsburgh, 1998.

[20] R. Maclin, et al., "Giving Advice about Preferred Actions to Reinforcement Learners via Knowledge-Based Kernel Regression," presented at the Proceedings of the 20th National Conference on Artificial intelligence, Pittsburgh, Pennsylvania, 2005.

[21] P. Stone, et al., "Reinforcement Learning for RoboCup Soccer Keepaway," Adaptive Behaviour, vol. 13, pp. 165-188, 2005.

[22] K. Nakamura and H. Igarashi, "Learning of Decision Making at Free Kicks using Policy Gradient methods," Proceedings of Robotics and Mechatronics Conference (ROBOMEC), June 2005 2005.

[23] H. Igarashi, et al., "Learning of Soccer Player Agents using a Policy Gradient Method: Coordination between Kicker and Receiver during Free Kicks," in IEEE International Joint Conference on Neural Networks (IJCNN), 2008, pp. 46-52.

[24] L. A. Celiberto, et al., "Heuristic Reinforcement Learning applied to RoboCup Simulation agents," in Robocup 2007: Robot Soccer World Cup Xi. vol. 5001, U. Visser, et al., Eds., ed Berlin: Springer-Verlag Berlin, 2008, pp. 220-227.

[25] R. A. Bianchi, et al., "Improving Reinforcement Learning by Using Case Based Heuristics," presented at the Proceedings of the 8th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development, Seattle, WA, 2009

[26] L. Mota and L. P. Reis, "Setplays: Achieving Coordination by the Appropriate use of Arbitrary Pre-De_ned Flexible Plans and Inter-Robot Communication," presented at the Proceedings of the 1st international conference on Robot communication and coordination, Athens, Greece, 2007

[27] V. Kyrylov and E. Hou, "While the Ball in the Digital Soccer is Rolling, Where the Non-Player Characters should go in a Defensive Situation?," presented at the Proceedings of the Conference on Future Play, Toronto, Canada, 2007

[28] V. Kyrylov and E. Hou, "Pareto-Optimal Collaborative Defensive Player Positioning in Simulated Soccer," presented at the RoboCup 2009: Robot Soccer World Cup XIII, 2010

[29] H. Kitano et al. RoboCup – The RobotWorld Cup Initiative.

[30] The RoboCup Small Size League website, http://small-size.informatik.uni-bremen.de/, December 2010

[31] Middle Size Robot League Rules and Regulations for 2010, http://groups.google.com/group/robocupmsl/web/msl-rules-2010-05-12.pdf, December 2010

[32] Official FIFA Rules, http://www.fifa.com/worldfootball/lawsofthegame.html, December 2010

[33] The Rocobup Humanoid League website, http://www.tzi.de/humanoid/bin/view/Website/WebHome, December 2010

[34] The Standard Platform League website, http://www.tzi.de/spl/bin/view/Website/WebHome, December 2010

[35] Aldebaran Robotics | Humanoid Robot - NAO's World - RoboCup website, http://www.aldebaran-robotics.com/eng/NaoRobocup.php, December 2010

[36] RoboCup Standard Platform League (Nao) Rule Book, http://www.tzi.de/spl/pub/Website/Downloads/Rules2010.pdf, December 2010

[37] H. Picado, "Development of Behaviors for a Simulated Humanoid Robot", Master Thesis,Universidade de Aveiro, Aveiro, 2008.

[38] Oliver Obst and Markus Rollmann. Spark - a generic simulator for physical multi-agent simulations. In Gabriela Lindemann, Jörg Denzinger, Ingo J. Timm, and Rainer Unland, editors, MATES, volume 3187 of Lecture Notes in Computer Science, pages 243–257. Springer, 2004.

[39] Joschka Boedecker, Klaus Dorer, Markus Rollmann, and Yuan Xu. SimSpark User's Manual, March 2008.

[40] Simspark community wiki, http://simspark.sourceforge.net/wiki/index.php/ , December 2010

[41] Russell Smith. Open Dynamics Engine (ODE) User Guide, May 2004.

[42] Patrick Riley. SPADES: A system for parallel-agent, discrete-event simulation. AI Magazine, 24(2):41–42, 2003.

[43] G. Steele et al, "The evolution of Lisp". The second ACM SIGPLAN conference on History of programming languages, 1993

[44] FC Portugal website, http://www.ieeta.pt/robocup , December 2010

[45] N. Almeida, "Control Agent Architecture of a Simulated Humanoid Robot", Master Thesis,Universidade de Aveiro, Aveiro, 2008.

[46] Luis Paulo Reis, Nuno Lau and Eugénio Costa Oliveira, "Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents". In Markus Hannebauer, Jan Wendler, and Enrico Pagello, editors, Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Lecture Notes in Computer Science, pages 175–197. Springer Berlin / Heidelberg, 2001.