FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

COORDINATED MULTI-ROBOT FORMATION CONTROL

Tiago Pereira do Nascimento

Doctoral Program in Electrical and Computers Engineering

Supervisor: António Paulo G. M. Moreira (PhD.)

Co-Supervisor: André Gustavo Scolari Conceição (PhD.)

October, 2012

COORDINATED MULTI-ROBOT FORMATION CONTROL

Tiago Pereira do Nascimento

Doctoral Program in Electrical and Computers Engineering

Resumo

Um sistema multi-robôs é formado quando um grupo de robôs interage com o ambiente como um único sistema. Este sistema pode também ser formado para realizar tarefas outrora difíceis ou até impossíveis de serem realizadas por um único robô.

Este trabalho apresenta os resultados da pesquisa desta tese de doutorado objetivando o controle de formação de um grupo de robôs móveis no problema de rastreamento ativo de alvo. Todos os robôs trocam informações sobre sua posição (localização) no mundo, a posição e velocidade do alvo formando um sistema totalmente distribuído. Dessa forma, o principal objetivo deste trabalho é procurar e rastrear um alvo minimizando a incerteza total sobre sua observação, desviando de colegas e obstáculos, obedecendo as características impostas à formação.

Um controlador preditivo não linear foi proposto para solucionar o problema de rastreamento ativo de alvo através da formação de um sistema multi-robôs. O controlador de formação criado como a principal contribuição desta tese é um controlador de formação preditivo não-linear (NMPFC). Este trabalho também apresenta outras contribuições em controle de formação no que tange o desvio de obstáculos, a ausência temporária do alvo e a formação de um grupo heterogêneo (robôs não-holonômicos e holonômicos). Estas contribuições melhoraram a formação tornando-a versátil, considerando incertezas dinâmicas, bem como a busca e perseguição de um alvo inicialmente ausente. Simulações e experimentos com robôs reais foram realizados, apresentados e discutidos.

Este trabalho é parte do projeto FCT: PTDC/EEA-CRO/100692/2008 - Perception-Driven Coordinated Multi-Robot Motion Control

Abstract

A multi-robot system is formed when a group of robots interact with the environment as a single system. This system can also be in formation in order to accomplish tasks rather difficult or impossible to achieve with a single robot.

This work presents results for a doctoral thesis research aiming the formation control of a mobile robot group in the active target tracking problem. All robots exchange information about their position (localization) in the world frame, and the target position and velocity forming a distributed system. Therefore, the main objective of this work is to search and track a target minimizing the total amount of uncertainty in the target's observation, avoiding mates and obstacles, and obeying the characteristics imposed to the formation.

A nonlinear model predictive controller was proposed in order to deal with the active target tracking problem through the formation of a multi-robot system. The formation controller created as the major contribution of this thesis is a nonlinear model predictive formation controller (NMPFC). This work also presents some contributions in the formation control concerning the obstacle avoidance, the temporary absence of the target and the formation control of a heterogeneous group (holonomic and nonholonomic robots). These contributions improved the formation making it versatile with dynamic uncertainties as well as the search and track of an initially absent target. Simulations and experiments with real robots were made, presented and discussed.

Finally, this work is inserted in the FCT project PTDC/EEA-CRO/100692/2008 - Perception-Driven Coordinated Multi-Robot Motion Control.

Support

With the support through the PhD Scholarship SFRH/BD/70698/2010 and

Acknowledgment

The author would like to thank:

At first to Jehovah, the only truthful God that has illuminated me in the most difficult times in my life and during this journey in three years as a PhD student;

My daughter Beatriz, my reason to live, for giving me the strength during all the difficulties by smiling at me, talking to me through phone and Internet;

My parents that supported me;

The Prof. Dr. António Paulo Moreira for accepting me as a PhD student in FEUP and for his orientation;

The Prof. Dr. André Gustavo Scolari Conceição for helping me be a PhD Student and for his orientation;

The Prof. Dr. Andrea Bonarini for accepting me in Politecnico di Milano as an internship student;

The Prof. Dr. Matteo Matteucci for helping me in Politecnico di Milano with his comments about my work;

The Prof. Pedro Lima for helping with his advices and accepting me in the FCT project that inspired my thesis;

My colleagues Luis Oliveira, Heber Sobreira, Marcos Ferreira, Filipe Neves for helping me through my PhD;

My good friends Hugo Alves, David Nossa, Helena Leal, Diego Almeida, Gustavo Bacelar and Daniele Bacelar, and all the others that I met during my life in Europe that easy my burden as friends.

And last but not least to all absent, present, colleagues and friends that directly or indirectly influenced in my path and choices made in this journey.

The author would like to thank also the INESC TEC for the financial support in the first year of the doctoral program and in the costs that this research presented, and the FCT for the financial support in the following years with the scholarship grant given (SFRH/BD/70698/2010).

Finally, the author thanks also to the FCT project PTDC/EEA-CRO/100692/2008 - "Perception-Driven Coordinated Multi-Robot Motion Control".

Tiago Pereira do Nascimento

"This journey really was Legen...
Wait for it...
...dary!"

Tiago Nascimento

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Objective . . . 3
- 1.3 Contributions . . . 4
- 1.4 Thesis Structure . . . 5

2 State of the Art **7**
- 2.1 Introduction . . . 7
- 2.2 Approaches in the Formation Control Problem . . . 8
  - 2.2.1 Virtual Structure Approach . . . 8
  - 2.2.2 Behavior-Based Approach . . . 11
  - 2.2.3 Leader-Following Approach . . . 13
  - 2.2.4 Other Used Approaches . . . 16
- 2.3 Model Predictive Control . . . 17
  - 2.3.1 Nonlinear Model Predictive Control . . . 21
- 2.4 Active Target Tracking Problem . . . 22
- 2.5 Discussion . . . 23

3 Robots and Experimental Setup Description **25**
- 3.1 Introduction . . . 25
- 3.2 Mobile Robots . . . 25
- 3.3 The Multi-Robot System . . . 26
  - 3.3.1 Mini-AGV Robot . . . 27
  - 3.3.2 The Middle-Size League 5dpo Robot . . . 27
- 3.4 The SimTwo Simulation Environment . . . 31
  - 3.4.1 The ODE Library . . . 33
  - 3.4.2 The 5dpo Robot Model . . . 35
- 3.5 Model Validation . . . 38
- 3.6 Conclusion . . . 39

4 Formation Control in Active Target Tracking **41**
- 4.1 Introduction . . . 41
- 4.2 Control Architecture . . . 43
- 4.3 Nonlinear Model Predictive Formation Control . . . 46
  - 4.3.1 The Model . . . 47
  - 4.3.2 The Cost Function During the Presence of a Target . . . 52
  - 4.3.3 The Cost Function During Target Absence . . . 57

4.4 The Optimization Algorithm . . . 58
4.5 Weight Tuning Methodology . . . 60
4.5.1 Initial Analysis of the Terms and Weights of the Cost Function . . 61
4.5.2 The Tuning . . . 62
4.5.3 The Influence of the Number of Terms . . . 63
4.6 Results . . . 66
4.6.1 Simulations . . . 67
4.6.2 Results of the Experiments with Real Robots . . . 76
4.7 Conclusion . . . 83

5 Obstacle Avoidance in Formation Control 85
5.1 Introduction . . . 85
5.2 Problem Formulation . . . 87
5.3 Potential Field Approach in NMPFC . . . 87
5.3.1 Mate Avoidance Function . . . 88
5.3.2 Obstacle Avoidance Function . . . 90
5.4 The Modified A* Path Planning Algorithm . . . 92
5.4.1 The Modifications . . . 94
5.5 Results . . . 95
5.5.1 Simulations . . . 95
5.5.2 Results of the Experiments with Real Robots . . . 101
5.6 Conclusion . . . 105

6 Intelligent State Changing Applied to Multi-Robot Systems 107
6.1 Introduction . . . 107
6.2 The State Machine . . . 111
6.3 The Takagi-Sugeno Type Fuzzy Automaton . . . 113
6.4 Problem Formulation . . . 115
6.4.1 Membership Functions . . . 117
6.4.2 The Problem Example . . . 117
6.5 Results . . . 119
6.5.1 Simulations . . . 119
6.5.2 Results of the Experiments with Real Robots . . . 123
6.6 Conclusion . . . 125

7 Conclusions and Future Work 127
7.1 Conclusion . . . 127
7.2 Future Work . . . 129
7.2.1 Changeable Leader Robot . . . 129
7.2.2 Model Considerations . . . 129
7.2.3 Parameters Tuning . . . 129
7.2.4 Heterogeneity Considerations . . . 130
7.2.5 Target Observation . . . 130
7.3 Final Considerations . . . 130

A The Covariance Model 131
A.1 Transformation in the Canonical Form . . . 131

References **135**

List of Figures

1.1 Examples of Mobile Robots . . . 1
1.2 Examples of Leader Following Formations . . . 2

2.1 Flocks of Migratory Birds in a V Formation . . . 8
2.2 Virtual Structure Formation. . . . 9
2.3 Leader-Following Formation without Target. . . . 13
2.4 Leader-Following Formation with Target. . . . 13
2.5 Basic Principle of the MPC [89] . . . 18
2.6 Generic MPC Structure. . . . 18

3.1 Omni-directional Wheel . . . 26
3.2 The 5DPO Mobile Soccer Robots . . . 27
3.3 The Mini-AGV Robot . . . 27
3.4 The General Architecture . . . 28
3.5 The HAL Software . . . 29
3.6 Coach Application . . . 30
3.7 The Dec Software . . . 30
3.8 The SimTwo Main Windows . . . 31
3.9 SimTwo Configuration Window . . . 32
3.10 SimTwo Code Editor . . . 32
3.11 SimTwo Plot Window . . . 32
3.12 SimTwo XML Editor . . . 33
3.13 Geometric Representation of the Robot's Body . . . 36
3.14 Trajectory for Model Validation . . . 38
3.15 Test with $1m/s$ and $\theta = 0^\circ$ . . . 38
3.16 Test with $1m/s$ and $\theta = Variable$ . . . 39

4.1 Structure of the Nonlinear Model Predictive Formation Controller Applied to robot 1 . . . 44
4.2 Controller Diagram . . . 45
4.3 The 5dpo robot graph representation. . . . 47
4.4 Model of Observation . . . 51
4.5 Formation Following a Leader . . . 58
4.6 Tuning Test Setup . . . 63
4.7 Testbed results: using the initial weights . . . 65
4.8 Simulation Setup . . . 67
4.9 Simulation 1: Homogeneous Formation Convergence - Plot XY . . . 67

4.10 Simulation 1: Homogeneous Formation Convergence - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$ . . . 68
4.11 Simulation 1: Heterogeneous Formation Convergence - Plot XY . . . 68
4.12 Simulation 1: Heterogeneous Formation Convergence - Distance between Robot and Ball and the Determinant of $\Sigma^{\perp}_{\text{Merged}}$ . . . 69
4.13 Simulation 2: Homogeneous Formation Convergence - Plot XY . . . 70
4.14 Simulation 2: Homogeneous Formation Convergence - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$ . . . 70
4.15 Simulation 2: Heterogeneous Formation Convergence - Plot XY . . . 71
4.16 Simulation 2: Heterogeneous Formation Convergence - Distance between Robot and Ball and the Determinant of $\Sigma^{\perp}_{\text{Merged}}$ . . . 71
4.17 Simulation 3: Homogeneous Leader Following - Plot XY and Distance between the Leader Robot and the Followers . . . 72
4.18 Simulation 3: Homogeneous Leader Following - Internal Product Between Leader and Followers and Error Angle of the Followers Orientation Facing the Leader . . . 72
4.19 Simulation 3: Heterogeneous Leader Following - Plot XY and the Distance between the Leader Robot . . . 73
4.20 Simulation 3: Heterogeneous Leader Following - Internal Product Between Leader and Followers and Angle Between the Leader and the Followers . . . 73
4.21 Simulation 4: Homogeneous Keeping Formation - Plot XY . . . 74
4.22 Simulation 4: Homogeneous Keeping Formation - Distance between Robot and Ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$ . . . 74
4.23 Simulation 4: Homogeneous Keeping Formation - Internal Product Between the Robots and the Ball and Error Angle of the Robots Orientation Facing the Target . . . 75
4.24 Simulation 4: Heterogeneous Keeping Formation - Plot XY . . . 75
4.25 Simulation 4: Heterogeneous Keeping Formation - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$ . . . 76
4.26 Simulation 4: Heterogeneous Keeping Formation - Internal Product Between the Robots and the Ball and Angle Between the Robots and the Ball's Velocity Vector . . . 76
4.27 Experiment Setup . . . 77
4.28 Real Experiment 1: Formation Convergence . . . 78
4.29 Real Experiment 1: Formation Convergence - Distance between the robot and the ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$ . . . 79
4.30 Real Experiment 2: Formation Convergence . . . 79
4.31 Real Experiment 2: Formation Convergence - Distance between the robot and the ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$ . . . 80
4.32 Real Experiment 3: Leader Following . . . 80
4.33 Real Experiment 3: Leader Following - Plot XY and Distance between the Leader Robot and the Followers . . . 80
4.34 Real Experiment 3: Leader Following - Internal Product Between Leader and Followers and Error Angle of the Followers Orientation Facing the Leader . . . 81
4.35 Real Experiment 4: Keeping Formation . . . 82

4.36 Real Experiment 4: Keeping Formation - Plot XY and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$ . . . 82
4.37 Real Experiment 4: Keeping Formation - Internal Product Between the Robots and the Ball and Error Angle of the Robots Orientation Facing the Target . . . 82

5.1 Behavior of function in 5.5 . . . 89
5.2 Local Minima Problem Case . . . 90
5.3 Switching approach . . . 91
5.4 Map Cell Decomposition [110] . . . 94
5.5 Obstacle's total radius [110] . . . 94
5.6 Simulation 1: Homogeneous Formation - Plot XY . . . 95
5.7 Simulation 1: Homogeneous Formation - Distance between Robot and Leader and Angle Between the Leader and the Followers . . . 96
5.8 Simulation 1: Heterogeneous Formation - Plot XY . . . 96
5.9 Simulation 1: Heterogeneous Formation - Distance between Robot and Leader and Angle Between the Leader and the Followers . . . 97
5.10 Simulation 2: Homogeneous Formation - Plot XY . . . 97
5.11 Simulation 2: Homogeneous Formation - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$ . . . 98
5.12 Simulation 2: Heterogeneous Formation - Plot XY . . . 98
5.13 Simulation 2: Heterogeneous Formation - Distance between Robot and Ball and the Determinant of $\Sigma^{\perp}_{\text{Merged}}$ . . . 99
5.14 Simulation 3 - Environment and Plot XY . . . 100
5.15 Simulation 3 - Distance between Robot and Ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}$ 100
5.16 Real Experiment 1 - Environment and Plot XY . . . 101
5.17 Real Experiment 1 - Distance between Robot and the Ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}$ . . . 102
5.18 Real Experiment 2 - Environment and Plot XY . . . 102
5.19 Real Experiment 2 - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$ 103
5.20 Real Experiment 3 - Plot XY . . . 104
5.21 Real Experiment 3 - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$ 104
5.22 Real Experiment 3 . . . 104

6.1 Robot State Machine . . . 112
6.2 Membership Functions and Surface . . . 117
6.3 The Problem Simulation: Formation Behavior without the TS-TFA approach . . . 118
6.4 The Solution Simulation: Formation Behavior with the TS-TFA approach 118
6.5 Simulation 1: Distance Robot-Ball and Plot XY of the Robots Path . . . 120
6.6 Simulation 1 . . . 121
6.7 Simulation 2: Distance Robot-Ball and Plot XY of the Robots Path . . . 122
6.8 Simulation 2: Orientation of the robots . . . 122
6.9 Simulation 3: Distance Robot-Ball and Plot XY of the Robots Path . . . 123
6.10 Real Experiment: Distance Robot-Ball and Plot XY of the Robots Path . . 123
6.11 Real Experiment: Coach View . . . 124

A.1 Model of Observation . . . 131

List of Tables

3.1 5dpo input parameters in SimTwo 35
3.2 Motor parameters in SimTwo 36
3.3 Partial Moments of Inertia 37
3.4 Mean Square Error for Model Validation Trajectories 39

4.1 Weights for the Cost Function with Only Four Terms 64
4.2 Weights for the Cost Function with Five Terms 64
4.3 Final Weights for the Cost Function: Holonomic Robots 65
4.4 Final Weights for the Cost Function: Nonholonomic Robots 66

6.1 Membership Functions-State Transitions. The values NV, VR and VM in the Quality (q) membership function means, respectively, *Not Visible*, *Visible Reasonably* and *Visible Much*, and HT, TR and ST in the Trust (t) membership function mean *High Trust*, *Trust Reasonably* and *Small Trust*, respectively. 113

Abbreviations and Symbols

AGV	Automated Guided Vehicles
APF	Artificial Potential Fields
AUV	Autonomous Underwater Vehicles
BP	Back-Propagation
CAD	Computer Aided Design
CC	Continuous Current
DC	Direct Current
FEUP	Faculty of Engineering from Porto University
HAL	Hardware Abstraction Layer
LP	Linear Problem
MPC	Model Predictive Control
NGN	Normalize Gaussian Network
NLP	Non Linear Problem
NM	Number of Mates
NMPC	Nonlinear Model Predictive Control
NMPFC	Nonlinear Model Predictive Formation Control
NO	Number of Obstacles
ODE	Open Dynamics Engine
PI	Proportional and Integrative
PIC	Programmable Interface Controller
PTS	Periodic Team Synchronization
RHPC	Receding Horizon Predictive Control
RPROP	Resilient PROPagation algorithm
RTDB	Real Time Data Based
TS	Takagi-Sugeno
TS-TFA	Takagi-Sugeno Type Fuzzy Automaton
UDP	User Datagram Protocol
XML	Extensible Markup Language

Chapter 1

Introduction

In this initial chapter, a generic introduction of the work is presented, by contextualizing it in a larger project and identifying the motivation for the development of this thesis. The objective and contributions made by the author will also be defined and explored. The structure of this work is later presented.

1.1 Motivation

Automation is being implanted in all economic sectors and therefore, mobile robots have been inserted in autonomous industry plants as AGVs, in domestic services (domotics), entertainment and so on (see Fig. 1.1). The automation of these sectors has suffered a great impact due to the insertion of these robots which has accelerated the whole process done by these autonomous equipments. The navigation and control problems for a single robot has been the topic of researches for a long time [1], [2] , [3], [4], [5], [6] and [7]. Nowadays, algorithms involving multi-robot motion have been conceived creating a new area of research inserted in the field of mobile robotics: the formation control algorithms for multi-robot systems.

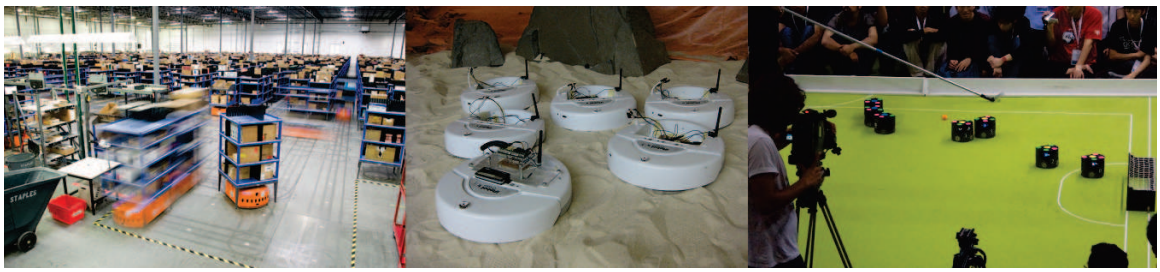

Figure 1.1: Examples of Mobile Robots

Formation control became a popular topic of study in multi-robot systems [8], [9], [10], [11], [12]. The problem of formation control is defined as the coordination of multiple mobile robots to follow given paths (in case of given trajectories, to track parameterized time reference trajectories) and to maintain a desired spatial formation. Solutions for formation control can be applied to a wide range of applications, such as search and rescue missions and land mine removal. The advantages of using a team of multiple robots include robustness, flexibility, and adaptability to unknown dynamic environments [13]. These are clearly important when considering applications such as search and rescue missions, deep ocean mapping, forest fire detection, mine removal detection, surveillance applications, intruder detection or even soccer robot competition.

The formation control research can be divided in three major approaches: the virtual structure approach, the behavior-based approach, and the leader-following approach. The virtual structure forms a rigid formation having high computational cost and breaking formation problems in case of dynamic mobile obstacles. The behavior-based approach is a well explored technique, even though there are problems in describing group dynamics and guarantying the group stability. The leader-following approach is the most explored approach for its versatility with mobile obstacles and formation convergence. This work presents some contributions that will help improve this last approach in the formation control issue when applied to the active target tracking problem.

It is important to mention that the leader-following approach with a formation controller algorithm can be applied in several cases of study. Several examples of applications can be seen in Fig. 1.2. Some of them are cooperative search and rescue, soccer robot competitions, forest fire detection and security and monitoring environment.

Figure 1.2: Examples of Leader Following Formations

Many control strategies have been investigated for solving formation control problems in the leader-following approach (e.g., [14], [15], [16], [17], [18] and [19]). Some of them present nonlinear model predictive control (NMPC) solutions. The NMPC controller is

based on a finite-horizon continuous time minimization of nonlinear predicted tracking errors with constraints on the control inputs and the state variables. It predicts system outputs based on current states and system model, finds an open-loop control profile by numerical optimization, and applies the first control signal in the optimized control profile to the system. Furthermore, to guarantee control stability many approaches have been investigated, e.g., using terminal region constraints and/or a terminal penalty term (see [20]). In this work, the NMPC was extended to a distributed approach to be applied in formation control and it is called throughout this work as Nonlinear Model Predictive Formation Control (NMPFC).

Another popular topic equally important to this thesis is the active target tracking problem. The cooperative active sensing literature, in particular, has grown steadily in the last decade: a list of representative works, yet far from being complete, is [21], [22], [23], [24], [25]. In the simplest instance of the cooperative active sensing problem, n mobile sensors have to fuse their local measurements and move in 2-D in order to attain the best estimate of the position of a moving target. This mechanism is sometimes referred to as "information-driven mobility" [23] and it amounts to minimize a certain scalar function of the covariance matrix of the position estimates: in optimum experimental design theory [26] this function is typically the determinant, the trace or the maximum eigenvalue of the covariance matrix. Cooperative active sensing leverages the mobility of a team of agents, typically equipped with sensors providing range-only or range-bearing measurements to the target, to enhance the target-tracking performances. Mobile sensor networks offer, in fact, distinctive advantages over arrays of static sensors in terms of quality of sensing and estimation, area coverage, and robustness against failures.

Therefore, this work is inserted in both active target tracking problem [27], [28], [29], [30] and formation control problem [31], [32], [33], [13], [34], [35], [36] where the formation is not rigid and must be such that the target's observation is minimized while keeping a minimum formation between the robots.

Finally, it is important to call the attention that this thesis is part of a project funded by FCT (*Fundação para Ciência e Tecnologia* - project PTDC/EEA-CRO/100692/2008 - Perception-Driven Coordinated Multi-Robot Motion Control) from Portugal which uses this proposed controller in a soccer robot competition scenario (indoor environment).

1.2 Objective

This doctoral thesis has the main objective of proposing a novel technique in the leader-following approach of formation control applied to the active target tracking problem. This novel approach has the following specific objectives:

- Select, conceive and implement a controller to perform a formation control on a group of mobile robots;
- Actively track and follow a target using a multi-robot system in formation;
- Enable the robot's controller to avoid moving or static obstacles, as well as other robots in formation;
- Enable the controller to be applied to an heterogeneous group of mobile robots with different dimensions (mechanical constrains) when joining a multi-robot system in formation;
- Implement a framework that is able to change controllers according to the presence or absence of the target.

1.3 Contributions

The envisaged main contributions in this doctoral thesis are:

- Conception and implementation of a nonlinear model predictive formation control (NMPFC) to perform a formation control with a group of mobile robots. This controller is used in a distributed configuration where each robot receives data from the other robots and processes its own task in the formation without the need of any kind of supervisor;
- While tracking a target, the nonlinear model predictive formation control (NMPFC) minimizes the total amount of uncertainty (merged uncertainties) in the target's perception during the control loop;
- A methodology to search for a set of weights for the NMPFC's cost function that shows the controller's efficiency through the minimization of an objective function acording to a chosen criteria.
- In the NMPFC's control loop, the controller enables the robot to avoid obstacles in movement or static, as well as other robots in formation by using a potential function within the controller's cost function. To avoid entrapment situation (singularities) a modified version of the A* global path planner algorithm leads the robot out of the entrapment situation;
- The NMPFC is also applied on holonomic and nonholonomic robots with different sizes when joining a multi-robot system in formation. A switching function enables changes in the model used to predict the robot's behavior. As the controller acts directly on the robot's velocities (in the robot frame) rather than the motor's velocities, it enables the controller to be generalized;

- A TS (Takashi-Sugeno) type fuzzy automaton is used to change cost functions or controllers according to the presence or absence of the target and to evaluate the change in the role given to the robot. Each robot uses the fuzzy systems to weight between the target's observation quality and its distance to this target, and therefore change the robot's state.

1.4 Thesis Structure

This work was here contextualized. In the following chapter a state of the art review is presented showing the principal researches done in the three major approaches of formation control and in the active target tracking problem. In chapter 3 the mobile robots used in this thesis and the simulation environment setup are presented. The validation of the robot's modeling in this simulator is also shown in this chapter. The nonlinear model predictive formation control conceived through this work and applied in formation control is presented in the chapter 4. The convergence of the controller is presented in this chapter along with simulations and experimental results. The controller here conceived is also applied to different robots and types of formation groups (heterogeneous group of robots). The obstacle avoidance problem is discussed and simulations and experiments are presented in the following chapter. Inside chapter 5 the global path planner algorithm used by the leader robot or in local minima entrapment obstacle avoidance problem is presented as well. In chapter 6 the final contribution is explained where the leader-following approach is submitted to the problem of presence and absence of the target and its influence to the formation and the search. Finally, the conclusions and future works are presented in the last chapter.

Chapter 2

State of the Art

This chapter summarizes the approaches most used in formation control of multi-robot systems and active target tracking problem.

2.1 Introduction

A formation is usually defined as the spatial arrangement of a group of agents, where the relative pose of its elements is steady even if the formation is moving. The concept of formation is not the product of human ingenuity. Formation comes from observable behaviors common in nature. For example, it is customary to glimpse flocks of migratory birds flying in a V formation (also called the *Four-Finger Formation*), which allows them to maximize the distance traveled and minimize fatigue by reducing air friction which is possible due to them flying in the air tunnel left by the bird traveling directly in front of them (see Fig. 2.1). This is true for all birds except the one that occupies the front pose. However, the specific bird that is given such task changes cyclically in order to distribute fatigue among members of the flock. Column formations are also quite common. For example, in the cases of a bird that leads a column made by their offspring allowing the mother to ensure a safe path as well as having the offspring maintaining eye contact with the bird directly in front in order to avoid getting lost (very common in ducks).

In the human sphere, formations are used in different situations ranging from military operations to team sports. A formation is used in order to gain a tactical advantage in the military context. Team sports also make extensive use of formations. For example, football is a sport where the highly tactical formations take on a role. In fact, there are over 30 of these arrangements varying from T or I to highly complex arrangements. In soccer, training describes the number of players assigned to each tactical pose in the field

Figure 2.1: Flocks of Migratory Birds in a V Formation

(generally defense, midfield and attack). Formations are less rigid in this case, being it an allocation of the action area of each player [20].

In robotics, there are two types of formation control approaches: centralized and decentralized. This chapter gives the advances in the decentralized multi-robot motion coordination techniques over the last decade [37], [38]. The three major decentralized approaches for formation control studied are: Virtual Structure, Behavior-Based and Leader-Following. This last one being one of the most studied in multi-robot formation [39], [40], [41], [42] and [43]. A good review of the three major approaches in formation control can be seen in [44].

2.2 Approaches in the Formation Control Problem

This section will discuss the latest developments, in all three approaches, as well as summarize the controllers used in the chosen approach for this work.

2.2.1 Virtual Structure Approach

Rigid-formation approaches have been emerging for almost a decade [45], [46], [47]. The virtual structure approach, was first presented by [48]. In the Virtual Structure approach the robot's formation no longer consists of leaders nor followers, i.e. no hierarchy exists in the formation (see Fig. 2.2). Control methods are developed to force a group of robots to behave in a rigid formation. In the virtual structure approach, the controller is derived from three steps. First, the desired dynamics of the virtual structure is defined. Second, the desired motion of the virtual structure is translated into desired motions for each agent. Finally, individual tracking controllers for each agent are derived from agent tracking. Later on, many variants of this approach have emerged in recent years. In [49] the authors consider two cooperative control problems for nonholonomic mobile agents. In the first problem, the design of cooperative control laws is discussed so that a group of

nonholonomic mobile agents cooperatively converges to some stationary point under various communication scenarios. Dynamic control laws for each agent are proposed with the aid of σ-processes and results from graph theory. In the second problem, the authors discuss the design of cooperative control laws such that a group of mobile agents converges to and tracks a target point which moves along a desired trajectory under various communication scenarios.

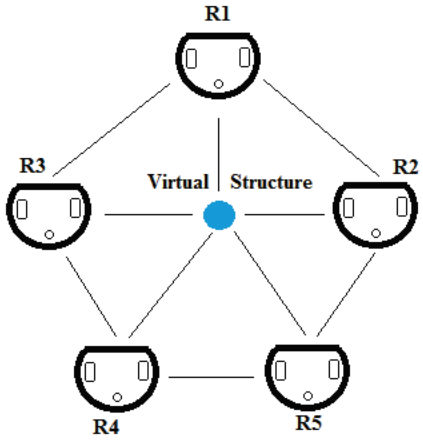

Figure 2.2: Virtual Structure Formation.

The authors in [50] present a synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations in a rigid structure. The main idea is to control each robot to track its desired trajectory while synchronizing its motion with those of other robots to keep relative kinematics relationships, as required by the formation. First, they pose the formation-control problem as a synchronization control problem and identify the synchronization control goal according to the formation requirement. Second, the authors develop a synchronous controller for each robot's translation to guarantee that both position and synchronization errors approach zero asymptotically. The rotational controller is also designed to ensure that the robot is always oriented toward its desired position. Both translational and rotational controls are supported by a centralized high-level planner for task monitoring and robot global localization.

Another variation of the Virtual Structure approach is the Cluster Representation. The authors in [51] used a cluster space state representation of mobile multi-robot systems as a means of enabling enhanced control of mobile multi-robot systems. A conceptual framework was proposed for the selection of appropriate cluster space state variables for a n-robot system, the development of formal kinematics that associate the cluster space state variables with robot-specific variables, and the implementation of a cluster space control system architecture. The cluster space approach was then demonstrated by examples of two- and three-robot clusters consisting of differential drive robots operating in a plane.

The control issue in a virtual structure is then approached by [52] were they present a

multi-layer scheme that controls a formation of n mobile robots, which includes a strategy for obstacle avoidance. The controller adopted is able to guide the robots to compose the desired formation and to track a desired trajectory, avoiding obstacles during the navigation. Two planning layers are responsible for: organizing the navigation of the individual robots towards the desired formation, minimizing energy consumption, and changing the robots reference signals in order to avoid collisions. Stability analysis performed on the closed-loop system shows that the formation errors are ultimately bounded.

Following the paper in 2007, [53] proposed a consensus tracking algorithm explicitly accounting for bounded control effort. This algorithm was then analyzed under a directed fixed interaction topology. Furthermore, convergence analysis for a consensus tracking algorithm was provided when the time-varying consensus reference state was available to a dynamically changing subgroup of the team under directed switching inter-vehicle interaction topologies. Experimental results of a formation control application were demonstrated on a multi-robot platform to validate one of the proposed consensus tracking algorithms.

Another approach was done by [54]. Here, an adaptive formation control method was proposed for multiple uncertain nonholonomic mobile robots at the actuator dynamics level. All parameters of the robot kinematics and dynamics, and actuator dynamics are unknown. The virtual structure with path parameters and the dynamic surface design methodology were combined to design an adaptive formation control scheme simpler than the previous backstepping-based control system. Using the Lyapunov stability theorem, the authors presented the adaptation laws for tuning all unknown parameters of multiple mobile robots regardless of path parameters in the reference trajectories.

One of the latest developments in the virtual structure approach was presented by [55]. Here, the authors combined the virtual structure and path following approaches to derive the formation architecture. A formation controller was proposed by the authors for the kinematic model of two-degree-of-freedom unicycle-type mobile robots. The approach was then extended to consider the formation controller by taking into account the physical dimensions and dynamics of the robots. The controller here was designed in such a way that the path derivative was left as a free input to synchronize the robot's motion.

The virtual structure can also be used in many fields besides ground vehicles such as UAV applications [56], or manipulators [12]. Nevertheless, the virtual structure approach has many issues regarding computational costs. As it can be seen, only few works that have used this approach consider obstacle avoidance ([3] and [52]). Even though, the obstacles are static and these works only have simulations results. When making experiments, most of the works do not consider obstacle avoidance and the average velocity is relatively low giving time for a centralized controller to process the information as it can be seen in [51], [50], [9], [53] and [55].

2.2.2 Behavior-Based Approach

Behavior-based approach and potential field approach are often combined in the application of formation control. In behavior-based approach [7], [57], [8], each robot has a basic motor schema. Each schema generates a vector representing the desired behavior response to sensory input. Possible motor schema include collision avoidance, obstacle avoidance, goal seeking, and formation keeping. The control action of each robot is a vector weighted average of the control for each motor schema behavior. Another good mixture of Behavior-based approach with rigid graph theory, can be seen in [58].

In [59], the authors describe the control of robot teams in the framework of Hilbert spaces. The focus of the paper was the intrinsic properties of robot control architectures, namely the conditions under which a generic mission can be successfully executed. The proposed paradigm was developed in two levels: (i) single robot control supported on a monotonic and non-expansive projection map defined over some behavioral space such as the robot configuration space or the velocity space, and (ii) team control supported on a supervised scheme over a set of neighboring relations among the teammates, accounting for their relative motion. Each robot monitors its own neighboring relations for relevant changes and adapts its motion to the objectives of the team using a finite state automaton supervisor.

In the same year, [60] proposed a feedback control strategy that achieves convergence of a multi-agent system to a desired formation configuration while avoiding collisions. The collision avoidance objective was handled by a decentralized navigation function that vanishes when the desired formation tends to be realized. Here, it was shown that under certain assumptions, formation infeasibility forces the agents' velocity vectors to a common value at steady state. This provides a connection between formation infeasibility and flocking behavior for the multi-agent system.

Another famous application of the behavior-based approach can be seen in robot soccer competitions. Here, [61] shows that multi-robot coordination is one crucial aspect in robotic soccer. The way each team coordinates its individual robots into cooperative global actions define the foundation of its strategy. In their work the authors explain how the robots from the CAMBADA team explore their behavior to win a match. Here, each robot is an independent agent. It coordinates its actions with its teammates through communication and information exchange. The resulting behavior of the individual robot should be integrated into the global team strategy, thus resulting in cooperative actions by all the robots. New roles were created to add to the team strategy and some of the previous existing roles were improved. Some of the existing behaviors were also improved to better fit the desired goals. Each role and behavior is described as well as the changes made.

A problem called the initial formation problem, within the multi-robot task allocation

domain is addressed in [62]. This problem consists in deciding which robot should go to each of the positions of the formation in order to minimize an objective. Two different distributed algorithms that solve this problem are explained in [62]. The second algorithm presents a novel approach that uses cost means to model the cost distribution and improves the performance of the task allocation algorithm. Also, the authors present an approach that integrates distributed task allocation algorithms with a behavior-based architecture to control formations of robot teams.

Many decentralized architectures use behavior-based approach in formation control. In [63], a decentralized formation control was proposed which enables collision free coordination and navigation of agents. The authors presented a simple method to define the formation of multi-agents and individual identities (IDs) of agents. Two decentralized coordination and navigation techniques were proposed for the formation of rovers. Agents decide their own behaviors onboard depending upon the motion initiative of the master agent of the formation. In these approaches, any agent can estimate the behavior of other agents in the formation. This reduces the dependency of individual agent on other agents while taking decisions. These approaches also reduce the communication burden on the formation where only the master agent broadcasts its motion status per sampled time. The main idea of this paper was to develop an adequate computational model, under which agents in the formation performed with the goal of coordinating among each other.

Recently, a behavior-based approach was implemented in [64]. The authors were inspired by a society of animals, which they studied, along with the coalition formation of robots for detecting intrusions using game theory. The authors consider coalition formation in a group of three robots that detect and capture intrusions in a closed curve loop. In their analytical model, individuals seek alliances if they "think" that their detect regions are too short to gain an intrusion capturing probability larger than their own. The authors also assumed that coalition seeking has an investment cost and that the formation of a coalition determines the outcomes of parities, with the detect length of a coalition simply being the sum of those of separate coalition members.

Finally, many applications beside ground vehicle applications also use the behavior approach. Keviczky et. al. [65] describe the application of a novel methodology for high-level control and coordination of autonomous vehicle teams and its demonstration on high-fidelity models of the organic air vehicle developed at Honeywell Laboratories. The scheme employs decentralized receding horizon controllers that reside on each vehicle to achieve coordination among team members. An appropriate graph structure describes the underlying communication topology between the vehicles. On each UAV, information about neighbors is used to predict their behavior and plan conflict-free trajectories that maintain coordination and achieve team objectives.

This approach is the most versatile for dynamic environments. Nevertheless, the difficulty to model the group dynamics causes this approach to be too specific. Each application environment requires a different behavior and, therefore, a different model of the system's dynamics and controller. Another issue addressed to this approach is the stability of the group, which is hard to guarantee.

2.2.3 Leader-Following Approach

Finally, the leader following approach is one of the most studied [1], [14], [16], [15], [17], [5], [18], [19], [11] and [66]. It is based on the existence of a leader (real or virtual) that follows the precise desired trajectory while the other robots of the formation just follow it, maintaining a preset distance and relative position. The leader robot can be a real robot (see Fig. 2.3) or it can be a virtual leader such as a target (see Fig. 2.4).

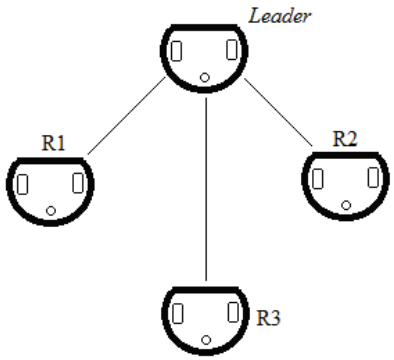

Figure 2.3: Leader-Following Formation without Target.

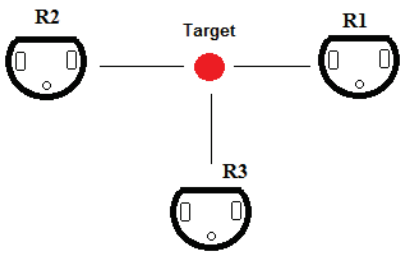

Figure 2.4: Leader-Following Formation with Target.

This approach was chosen as being the most adaptable to the active target tracking problem further discussed in this chapter. Here, many controllers can be used to maintain the usually rigid formation. In [67], the authors investigate the control and localization of a heterogeneous group of mobile robots. Therefore, the nonholonomic leader robot

guides a group of nonholonomic followers when the follower robots are different from the leader robot. The group considered had several inexpensive sensor-limited and computationally limited robots, which follow a leader robot in a desired formation over long distances. Complex sensing and computation were performed by the leader, while the followers performed simple operations under the leader's guidance. This architecture allowed followers to be simple, inexpensive, and to have minimal sensors. Theoretical and statistical analysis of a tracking-based localization method was provided. A simple follow-the-leader control method is also presented, including a method for changing follower's configuration. Nevertheless, in the proposed article, the average velocity of the group is no greater than 0,15m/s.

The authors in [68] introduce and discuss the cooperative leader following task for multi-robot teams. They describe the design and implementation of a distributed technique to coordinate team level and robot level behaviors for this task, as well as a multi-threaded framework for the implementation of a multi-robot system with heterogeneous sensing capabilities. This approach enables robots to remain in formation as they deal with other obstacles that may appear within the formation. The authors show some of the results of the team implementations in indoor and outdoor environments.

In [69], a good implementation of a linear controller can be seen. In their proposed paper, they study the problem of modeling and controlling leader-follower formation of mobile robots. First, a kinematics model for leader-follower robot formation was formulated based on the relative motion states between the robots and the local motion of the follower robot. Using this model, the relative centripetal and Coriolis accelerations between robots were computed directly by measuring the relative and local motion sensors, and utilized to linearize the nonlinear system equations. A formation controller, consisting of a feedback linearization part and a sliding mode compensator, was designed to stabilize the overall system including the internal dynamics. The control gains were determined by solving a robustness inequality and assumed to satisfy a cooperative protocol that guarantees the stability of the zero dynamics of the formation system. The proposed controller generated the commanded acceleration for the follower robot and made the formation control system robust to the effect of unmeasured acceleration of the leader robot. Furthermore, a robust adaptive controller was developed to deal with parametric uncertainty in the system.

Many other examples of linear controllers applied to the leader-following approach emerged in the last decade such as [70], [71], [72] and [73]. Nevertheless, in many cases this cannot be generalized or sometimes are not even feasible [74], [75]. The problem with linear controllers for formation control problems is the constrains of the formation group. If the trajectory is dynamically generated or there are dynamic uncertainties, this type of controller does not generate as good results as nonlinear controllers.

A new position feedback based formation control method for heterogeneous multi-robot teams is presented and evaluated in the paper from [76]. The formation behaviors are integrated with dynamic reference object based collaborative navigation and efficient obstacle avoidance to maintain and change formation real-time. The method from [76] is computationally efficient and easy to coordinate in heterogeneous systems. The time to formalize and switch specified formation patterns can be controlled by adjusting the position feedback parameter. Satisfactory experimental results are obtained in simulation and real heterogeneous multi-robot systems which consist of autonomous vehicles and legged robots.

Linear controllers considering dynamic constrains have also been considered in leader-following formation control [77]. In [78], the authors developed a distributed tracking control scheme with distributed estimators for a leader-follower multi-agent system with noise measurement and directed interconnection topology. It was supposed that each follower could only measure the relative positions of its neighbors in a noisy environment, including the relative position of the second-order active leader. A neighbor-based tracking protocol together with distributed estimators was designed based on a novel velocity decomposition technique. It was shown that the closed loop tracking control system is stochastically stable in mean square and the estimation errors converge to zero in mean square as well.

A final approach in the leader-follower area is the Model Predictive Controller (MPC). It has been the target of studies in multi-robot motion control for almost a decade [1]. The MPC is proven to be more effective than other types of linear controllers specially in low dynamic systems and multi-variable systems. Amongst other reasons to use the MPC, there are the following:

- As the future is known (reference behavior) it is possible to predict future system behaviors;
- The generic structure of the MPC offers the possibility of controlling different types of robots;
- The robot's dynamics can be considered more or less precise depending on the applied system;
- The nonlinearities can be included in the prediction model;
- The prediction model can be switched while maintaining the same control structure;
- The MPC can be implemented in a distributed fashion;
- In a single optimization problem, energy costs, obstacle avoidance and other issues can be considered;

- Different behaviors and actions can be changed by a simple change in the cost function parameters;
- MPC allows the explicit consideration of state, input and output constraints;
- In MPC, a specified performance criteria is minimized on-line;

The next sections will discuss some of the MPC techniques used in the leader-following approach. The nonlinear model predictive control (NMPC) as well as the active target tracking problem will also be considered. Furthermore, a final discussion about the technique used as a start point of this work is presented.

2.2.4 Other Used Approaches

Some approaches were created that cannot be classified as the three decentralized approaches previously explained. Relevant examples are the potential fields that are one of the oldest techniques used [79], [80], [81]. In [82], for instance, the authors present an improvement of the classical potential field approach for motion coordination in formations of Multi-Robot Systems. Different forces belonging to other robots, obstacles and the aspired shape of formation are combined and used to move each robot to its desired position inside the formation. The group is able to avoid obstacles and approach towards a specified target, while moving in formation

Another technique is used by [83]. Here, the authors address the problem of planning the motion of a team of cooperating mobile robots subject to constraints on relative configuration imposed by the nature of the task they are executing. They model constraints between robots using a graph where each edge is associated with the interaction between two robots and describes a constraint on relative configurations. The authors also develop a decentralized motion control system that leads each robot to their individual goals while maintaining the constraints specified on the graph.

Also in the decentralized approach, the authors in [84] studied and analyzed the information flow and its relation to stability of the motion of vehicles in a rigid formation. The authors also consider the effect of information flow on the propagation of errors in spacing in a collection of vehicles trying to maintain a rigid formation during translational maneuvers. The motion of each vehicle is described by using a linear time-invariant (LTI) system. It is also considered undirected and connected information flow graphs, assuming that each vehicle can communicate with a maximum of vehicles

Finally, neural networks controllers have also emerged among the ones used in formation control [85]. In [86], the authors addressed a formation control issue of a group of robots with dynamic interaction topology. They introduced a decentralized formation control to realize formation based in Neural Networks. Moreover, based on analysis of switched systems, the sufficient conditions on formation pattern and interaction topology

are fingered out to ensure the feasibility of this formation control, even if the interaction topology often changes suddenly. The simulation illustrates that within a field with obstacles (although not so clear) that may interrupt interaction between robots, a leader-following formation must reach the destination by using this formation control despite the low velocities of the group here presented. The same low velocity problem occur later in [87].

2.3 Model Predictive Control

Model Predictive Control (MPC) originated in the late seventies and has developed considerably since then [88]. The term MPC (also called Receding Horizon Predictive Control, or RHPC) does not designate a specific controller but rather an ample range of control methods which make explicit use of a model of the process to obtain the control signal by minimizing a cost function. These design methods lead to controllers which have practically the same structure and present adequate degrees of freedom. The various MPC algorithms only differ amongst themselves in the model used to represent the process, in the noises, and in the method/algorithm to minimize the cost function. The concepts appearing in greater or lesser degree in the predictive control family are basically:

1. Explicit use of a model (linear or nonlinear) to predict the process output at future time instants (prediction horizon);
2. Calculation of a control sequence minimizing a cost function;
3. Receding strategy, so that at each instant the horizon is displaced towards the future.

According to [89], the model predictive control problem is formulated as solving online a finite horizon open-loop optimal control problem subject to system dynamics and constraints involving states and controls. Fig. 2.5 shows the basic principle of model predictive control. Based on measurements obtained at time t, the controller predicts the future dynamic behavior of the system over a prediction horizon T_p and determines (over a control horizon $T_c \leq T_p$) the input such that a predetermined open-loop performance objective functional is optimized. If there were no disturbances and no model-plant mismatch, and if the optimization problem could be solved for infinite horizons, then one could apply the input function found at time $t = 0$ to the system for all times $t \geq 0$. However, this is not possible in general. Due to disturbances and model-plant mismatch, the true system behavior is different from the predicted behavior. In order to incorporate some feedback mechanism, the open-loop manipulated input function obtained will be implemented only until the next measurement becomes available. The time difference between the recalculation/measurements can vary, however it is assumed often fixed, i.e the measurement will take place every δ sampling time-units. Using the new measurement at

time $t+\delta$, the whole procedure (prediction and optimization) is repeated in order to find a new input function with the control and prediction horizons moving forward.

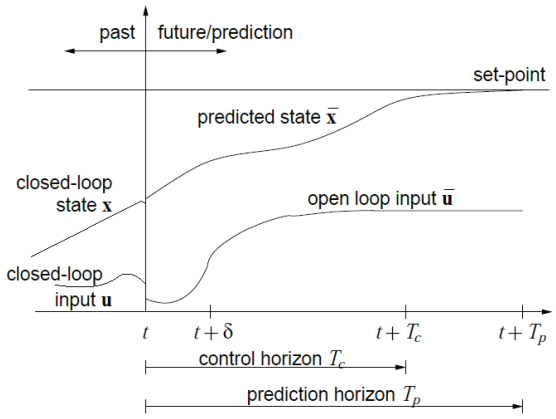

Figure 2.5: Basic Principle of the MPC [89]

The optimization of the cost function has several peculiarities, such as regarding how it is performed. If the criterion used in the cost function is quadratic, the model is linear, and there are no constraints on the system, then an explicit solution can be found offline through the implementation of a simple lookup table or an explicit function of past inputs, outputs and future set-point/trajectory, if available. In the usual case of non-linear constraints, or when the cost function takes a less usual form, it is necessary to do a real-time online optimization, using a numerical method. In order to guarantee the convergence of minimization and achieving smooth control signals, it is also customary to include in the cost function a term that penalizes the effort of control. This control strategy is typically implemented using the basic structure of the model presented in Fig. 2.6.

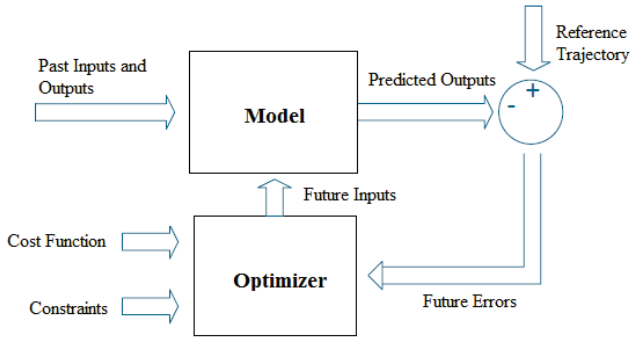

Figure 2.6: Generic MPC Structure.

There are the two main elements of a predictive controller illustrated above: the optimizer and system model. The model uses past information to predict the system response

of the future control inputs. These are calculated by the optimizer, which minimizes errors in the future by optimizing the cost function for ideal control signals. The system constraints are taken into account in the optimization [89], [88].

The first year in which a Model Predictive Controller was applied in the Leader-Following approach of multi-robot formation was 2002 [1]. After that many works improved the technique through time, such as [90]. Here, the authors investigated the leader-following formation control of mobile robots through the MPC. They established its control stability by adding a terminal state penalty to the cost function and a terminal state region to the optimization constraints. The authors also designed a terminal state region based on an input-output feedback linearization controller for the MPC. A suboptimal stable solution is thought to reduce the computational time used in the MPC. Nevertheless, this approach possesses still a high computational cost.

Fukushima, Kon and Matsuno present in [16] a distributed model predictive control (MPC) method for unicycles formation with collision avoidance constraints. This is a good implementation of the MPC in leader-following approach despite the fact that this work did not consider environment obstacles or even obstacles with uncertainties. The proposed method first stabilized the system by using a feedback linearization, then a collision avoidance method based on MPC was applied to the linearized system. One of the features of the proposed method was that each vehicle sequentially solves its optimal control problem at different time steps. Unlike other MPC collision avoidance methods in which all vehicles solve optimal control problems at each time step, only one vehicle can solve its optimization problem at one time step. The authors also derived a condition for the proposed method to ensure the feasibility of the optimization method and stability of the closed-loop system.

In 2006, an elaborated MPC controller was conceived by [91], although it was not tested in real robots. In the same year, a good simulation was made by [92]; they developed a receding horizon (RH) controller (also called Model Predictive Controller) for tracking control of a nonholonomic mobile robot. The control stability was guaranteed by adding a terminal-state penalty to the cost function and constraining the terminal state to a terminal-state region. The stability analysis in the terminal-state region was investigated, and a virtual controller was found. The analysis results showed that the RH tracking control has simultaneous tracking and regulation capabilities.

In the following year, [93] considered the problem of controlling a team of mobile robots with nonholonomic constraints to leader-following formations. The authors proposed that it was more convenient to put the nonholonomic constraints inside the model predictive control (MPC) framework. As the first step of exploration, a dual-mode MPC algorithm was developed. The stability of the formation was guaranteed by constraining the terminal state to a terminal region and switching to a stabilizing terminal controller at the boundary of the terminal region. The effectiveness of the method was investigated by

numerical simulations, therefore, the computational cost was left aside to investigate the stability of the system.

In 2008 the first use of a MPC applied in the leader-following approach using holonomic robots was done by [94], [95]. In these approaches only the formation maintenance is discussed. In these works the circle trajectory and the eight trajectory were used. The robots should follow the paths in a preset time while changing their formation (column or triangle). It is known that in a highly dynamic environment, if the trajectory is predefined, the linear MPC, even if being applied to a nonlinear system, can control the system maintaining the set-point. This was the exact result given by the authors.

In 2009, [96] made another implementation using this time a Nonlinear Model Predictive Controller in the formation of multi-robot system of holonomic robots. The authors presented experimental results of formation control problems of omnidirectional mobile robots using distributed nonlinear model predictive control (NMPC). Two main objectives were (i) to maintain a desired flexible formation pattern and (ii) to follow a reference path. Both pose errors and formation errors which were included into a local objective function, later minimized at each update time. The strategy was in such a way that the exchange of the most recent optimal state trajectory between coupled subsystems had been employed. The distinct features of NMPC were that constraints could be explicitly accommodated, and nonlinear systems, as well as time-varying ones, could be easily handled. Experiments with three omnidirectional mobile robots were presented in order to illustrate the validity of the author's proposed method. Nevertheless, here again the robots did not converge to a formation, and neither dynamic uncertainties nor fixed obstacles were considered.

Also in 2009, [97] and [98] applied a Linear Model Predictive Controller in the leader-following approach. [97] used it to control a set of AUVs (Autonomous Underwater Vehicles) while [98] applied it in nonholonomic robots using the separation principle to make a NMPC control the trajectory while a MPC would be used to formation control. Both works only had simulation results and no obstacles were considered.

A work done by [38] used a MPC applied to formation control. In his work, the author considers also a heterogeneous group of mobile robots in formation. The author applied a strategy into a mobile robot team formed by small robots with a simple microcontroller and modest sensors such as wheel encoders, and a larger leader robot with more computational power. The leader is responsible for group navigation and the control of team formation. It has an omnidirectional camera which allows it to visualize other robots. Color segmentation and a Kalman filter are used to obtain the positions of the followers, related to the leader, while the orientation measured by a compass sensor is sent to the leader robot. The omnidirectional visual feedback has the advantage of allowing the leader to localize all the followers around itself by taking just one image.

The linear model predictive control (or just MPC) is characterized by a family of predictive control strategies in which the dynamic model used to predict the evolution of the system is linear, even though the dynamics of the closed-loop system are nonlinear due to the existence of constraints. This simplification is made by considering that the system operates in a controlled area where it can actually be linearized. However, there are many systems that are inherently nonlinear, in which case a nonlinear model has to be necessarily used. This fact led to the development and use of the nonlinear model predictive control (or NMPC). Finally, it is important to mention that the objective of this work does not include a stability analysis of the proposed controller.

2.3.1 Nonlinear Model Predictive Control

In NMPC, the choice of the appropriate model is not the only important issue. Changing to a nonlinear model transforms the control problem from a convex quadratic program to a non-convex nonlinear problem, which is much more difficult to solve. Furthermore, in the discussed situation there is no guarantee that the global optimum can be found, especially in real-time control when the optimum has to be obtained in prescribed time [88].

In this thesis, the prediction of the system's behavior in an instant k is given by $\hat{P}(k+i|k)$ with $i = 1..N_p$ where N_p is the prediction horizon. This system's prediction depends on the future output control signals $\hat{U}(k+i|k)$ with $i = 0..N_p - 1$ that are applied to the system's model. The set of future output control signals are calculated through the optimization process with respect to a chosen criteria (for example a cost function) in order to keep the system behaving as desired and penalizing its deviation. The output control signal $U(k|k)$ is then sent to the system while the other future output control signals are not considered. As soon as the next instant $P(k+1)$ of the controlled system is known, the process is repeated in the same fashion.

The system's model is the most important element of the NMPC because it is through the model that it can calculate the predicted system's outputs in the future instants $\hat{P}(k+i|k)$. The model used by the controller in this thesis is presented in the chapter 4. This model is a nonlinear representation of the behavior of the formation of a multi-robot system. The discrete model used to predict the system's behavior is given by $P(k+1) = f(P(k),U(k))$ where $U(k) = \begin{bmatrix} v_{ref}(k) & vn_{ref}(k) & w_{ref}(k) \end{bmatrix}^{\mathrm{T}}$ are the robot's velocities.

In NMPC, optimization algorithms such as the Steepest Descent algorithms can work as long as the gradient function is not unpredictable. Here, the Resilient Propagation algorithm (RPROP) will be used. The RPROP algorithm is an adaptive learning rate algorithm which automatically varies the rate of weight adjustment with the general goal of speeding up convergence and handling large-scale nonlinear optimization [99].

2.4 Active Target Tracking Problem

Optimally tracking a moving target under motion and process constraints is necessary in a number of applications such as surveillance, environmental monitoring, defense applications and so on. In most studies about target tracking, the sensors involved are static and the emphasis is on the optimal processing of the available information. In contrast to using static sensors, the deployment of *mobile* sensors (or *robots*) for tracking offers significant advantages. For example, a larger area can be covered without the need to increase the number of nodes in the sensing network. The idea of optimally choosing the mobile sensors' locations in order to maximize information gain (also known as adaptive sensing or active perception) has been applied to the problems of cooperative localization [100], simultaneous localization and mapping [101], parameter estimation [102], and optimal sensor selection [103].

Yang et al. [24] present an active sensing strategy using distance-only measurements, where both the *trace* and the *determinant* of the target position estimates of the covariance are considered as the objective functions. The authors propose a control law, with constant step size, based on the *gradient* of the cost function with respect to each sensor's coordinates.

In [21], Martínez and Bullo address the problem of optimal sensor placement and motion coordination strategies for *homogeneous* sensor networks using distance-only measurements, where the emphasis is on the optimal sensor *placement* for (non random) *static* target position estimation. The objective is to minimize the *determinant* of the covariance matrix. The resulting control law requires that the sensors move on a polygon surrounding the target so that the vectors from the target to the sensors are uniformly (in terms of direction) spaced.

Recently, Frew [104] has investigated the problem of *single sensor* trajectory generation for target tracking using bearing measurements. In their discussed problem, motion constraints on the sensor's trajectory are explicitly incorporated in the problem formulation and the objective function (*determinant* of the target's covariance matrix) is minimized over a finite time horizon using *exhaustive search* through a discretized set of candidate sensor headings.

Olfati-Saber [23] addresses the problem of distributed target tracking for mobile sensor networks with a dynamic communication topology. The author tackles the network connectivity issue using a flocking-based mobility model and presents a modified version of the distributed Kalman filter algorithm for estimating the target's state. In this case, the sensors use both distance and bearing measurements to a target that moves in 2D with constant velocity driven by zero-mean Gaussian noise, and seek to *minimize their distances* to the target, while avoiding collisions.

Chung et al. [22] present a decentralized motion planning algorithm for solving the

multi-sensor target tracking problem using both distance and bearing measurements. The authors employ the *determinant* of the target's position covariance matrix as the cost function. The decentralized control law in this case is based on the gradient of the cost function with respect to each of the sensor's coordinates with constant step-size of 1.

The main drawback of the previous approaches is that *no physical constraints* on the motion of the sensors are considered. The only exceptions are the works presented in [105] for distance-and-bearing observations, and in [104] for bearing-only observations. However, in both cases, the proposed grid-based exhaustive search algorithm, when extended to the multi-sensor case, has computational complexity *exponential* in the number of sensors, which becomes prohibitive when the number of the sensors is large and/or the size of the grid cell is small. In addition, teams of heterogeneous sensors using mixed (i.e., distance and/or bearing) relative observations are only considered in [24], whose gradient-based algorithm can only guarantee achieving local minimum, while its convergence rate is not addressed. Furthermore, none of the previous works contemplates issues regarding the mobile robots' formation with respect to the mobility such as obstacles and mates collision avoidance, energy issues, online control law changing and so on.

2.5 Discussion

This chapter presented two distinct problems: the formation control problem and the active target tracking problem. Studies usually consider one problem or the other separately. Nevertheless, in some situations these two problems can appear together, making the solution of either one of them hard to conceive. For example, in the formation control problem (with the leader-following approach) usually the formation must be with pre-defined distances between the robots (rigid formation) and the controller is said to be working if the formation can be kept during a pre-defined trajectory. In the active target tracking problem the issue lies within the observation of a target disregarding a formation between the robots. An optimal observation is the main task in this last problem.

Therefore, the problem addressed in this thesis lies in the boundaries between the formation control problem and the active target tracking problem. The problem here is to conceive a formation controller capable of control a multi-robot system in a distributive fashion considering obstacle and mates avoidance, the formation itself and the maximization of the target's observation by the group of robots in formation.

After a careful analysis of the state of the art, the nonlinear model predictive control theory was considered as the main theoretical approach. Amongst the reasons that makes the nonlinear model predictive control theory the most suitable to solve the addressed problem, the following are the relevant ones:

- Sometimes the future reference behavior is known and it is possible to use this information to predict future behaviors of the system;
- By obtaining the same generic structure of the NMPC, the control of different types of robots becomes possible while maximizing the observation with different types of sensors and mobility in the formation;
- The nonlinearities can be included in the prediction model. As an example, the total amount of uncertainty was predicted for each robot and minimized in the controller's cost function;
- A change in the model of the target's observation can be easily made in the controller's code generalizing the application for different sensors;
- Energy costs, avoidance of obstacles, uncertainties and formation characteristics can be considered in a single optimization problem;
- Different behaviors and actions can be changed by a simple modification in the cost function parameters allowing for better observation and a change in the formation;
- Regarding the formation geometry, available formation control methods often give little relevance to the requirements imposed by target localization and/or tracking.

Finally, among the disadvantages, the NMPC has the following:

- It is difficult to theoretically prove the systems' stability;
- It possesses reasonable computational costs in the system's processing;
- In NMPC, the model of the system must be minimally realistic (an accurate model of the system) in order to guarantee accurate predictions of the system's behavior.

Chapter 3

Robots and Experimental Setup Description

This chapter presents the robots used in this thesis. It also explains the simulation and experimental setups. Finally, the simulator SimTwo and the validation of the robot model in this environment are presented.

3.1 Introduction

The study on mobile robotics is one of the most growing and investigated topics in robotics. Its application is relevant in a large set of environments such as industrial, medical and domestic. In this chapter, aspects of the ground mobile robots used as a mechanism to validate the proposed controller will be presented.

The respective used models are further briefly described in this chapter. Therefore, this chapter is structured presenting a brief consideration about mobile robots in the following section. The multi-robot system used is presented in section 3.3. Then, the SimTwo environment is explained in section 3.4. In section 3.5 the model of the 5dpo robot is validated in the SimTwo environment and the conclusion is presented in the final section.

3.2 Mobile Robots

A mobile robot must be capable to move itself through an environment, avoiding collisions with static or moving obstacles in its trajectory to reach the objective. The robot can be a water robot, an air robot or a ground mobile robot. The autonomous ground mobile robots (or simple mobile robots) can have a variety of locomotion elements such as wheels, roller balls, legs or crawlers.

This work uses two kinds of mobile robots with wheels, called holonomic and non-holonomic robots. There are a variety of wheels classified in five categories by [106]: normal standard wheels, standard maneuver wheels, beaver wheels, spheric wheels and omnidirectional wheels. Each type of wheel has a different degree of freedom. The non-holonomic mobile robots present some constrains on their wheels that holonomic robots do not. The omnidirectional wheels are used in holonomic robots and the normal standard wheels are used in most industrial AGVs (Automated Guided Vehicles).

The mechanical structure of an holonomic mobile robot is designed to contain three or more motors connected to an omnidirectional wheel (see Fig. 3.1). In this thesis, the omnidirectional robots have three wheels distributed in 120 degrees. These wheels have small rollers on their surface which lower the friction of lateral slippering enabling orthogonal movement to the traditional direction.

Figure 3.1: Omni-directional Wheel

In a nonholonomic mobile robot, the mechanical structure can assume a much larger variety of forms. The ones most used have four wheels, as it can be seen in [107]. The geometry, the number of wheels and their geometric positions on the chassis of the robot are responsible for the robot's mechanical stability, capacity of maneuver (maneuverability) and its degree of mobility.

3.3 The Multi-Robot System

Several robotic tasks require or benefit from the cooperation of multiple robots: transportation of large-sized objects, large area coverage (e.g., for cleaning) or surveillance (e.g., for fire detection), pollutant plume tracking, or target detection and tracking, to name but a few. Usually, in multi-robot systems, the formation can be rigid or not, and the control strategy can be centralized or distributed. Also, the formation must have low computational and power requirements and high communication efficiency.

This research used an indoor highly dynamic and adversarial environment for heterogeneous and homogeneous multi-robot system with a common target (tracking a ball), often subject to occlusions and *kidnappings*. The used robots are omnidirectional robots from the middle size league (see Fig. 3.2) of robot soccer championships. Furthermore, a differential robot is used to demonstrate the application of the same controller in a non-holonomic robot (see Fig. 3.3).

Figure 3.2: The 5DPO Mobile Soccer Robots

3.3.1 Mini-AGV Robot

In Fig. 3.3 the automated guided vehicle (AGV) robot also used in this work, here called Mini-AGV, can be seen. The main characteristic of the non-holonomic Mini-AGV mobile robot is the fact that it possesses a differential mechanical traction. This system consists of a set of two normal standard wheels coupled in two continuous current (CC) motors, set in the same central axis and symmetric opposite to one another. To give stability and support to the structure, two free wheels are also placed in the robot's structure. This configuration can be seen in a simple locomotion mechanism, allowing the platform freedom in its movement in comparison to its size, such as the ability to rotate in its own axis. The kinematic and dynamic models of this mobile robot, as well as the use of the SimTwo simulator, can be seen in [107].

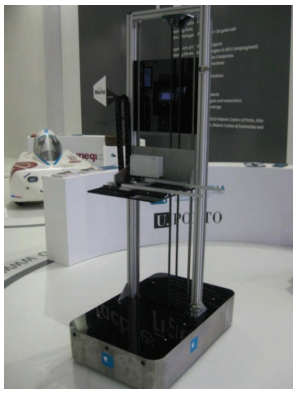

Figure 3.3: The Mini-AGV Robot

3.3.2 The Middle-Size League 5dpo Robot

The 5dpo is the RoboCup middle size league soccer team of the FEUP and INESC TEC, Portugal. The construction of this robot was the result of various people working on several different areas which contributed for the development of the mechanical structure of

the robot, its hardware architecture and controllers, as well as the software development in areas such as image analysis and processing [108], [109], real-time path planning [110], modeling [111], [112] and control [113] [114]. Nevertheless, it is important to mention that, despite these robots are used in this thesis as a testbed for the controller proposed here, many of the features described bellow such as the RTDB and the formation controller itself are not used in the RoboCup games, but in the project of which this thesis is a part of (FCT project PTDC/EEA-CRO/100692/2008 - Perception-Driven Coordinated Multi-Robot Motion Control).

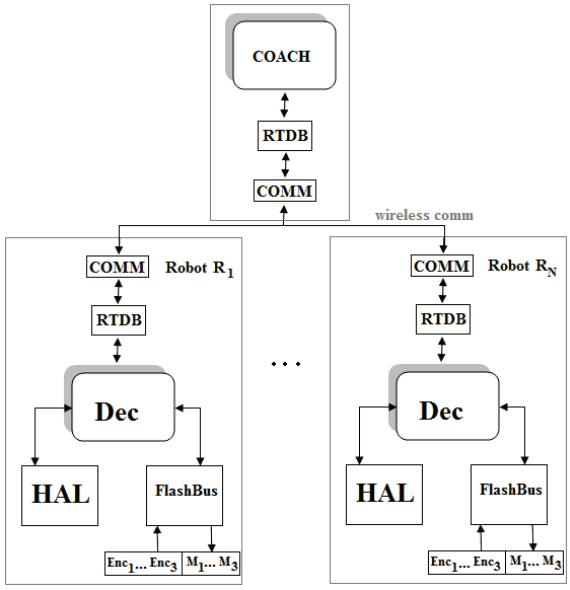

Figure 3.4: The General Architecture

Fig. 3.4 shows that the Dec is the control software, HAL is the vision system application, FlashBus is the hardware communication application, Coach is a central computer application to visualize and log the formation activities and positions, RTDB is the communication protocol, COMM is the communication application, Enc_i is the encoder of motor i and M_i is the motor i.

The general architecture of the 5dpo robot is primarily based on a multi-agent system. Each agent is placed in a main processing unit (a laptop), which is responsible for the high-level coordination. This main processing unit handles external communication with other robots, intermediated by the shared real time data base (RTDB) through wireless communication. All robots read and write their shared variables on their local real time data base (RTDB), which is then broadcasted by an ad-hoc communication system [115]. This unit (the above mentioned laptop), besides having a vision system directly attached to it, receives also a low bandwidth sensing information and sends actuating commands to control the robot actions by means of a distributed low-level system (Fig. 3.4).

The low level sensing/actuating systems follows a distributed model where most of

elementary functions (closed-loop PID control of the actuators) are encapsulated in small microcontroller based nodes connected to a motherboard by a RS485 electric specification. For this purpose, a protocol called FlashBus was created to support communication to exist between the main processing unit and the microcontroller based nodes.

3.3.2.1 Vision System

The vision application is called HAL (Hardware Abstraction Layer) and it communicates with the Dec (control software of the robots) by local UDP protocol. The HAL interface can be seen in Fig. 3.5. The function of this application is to treat the images captured by the camera and send to the features to the Dec software to be processed. The features are related to field lines, ball and obstacles.

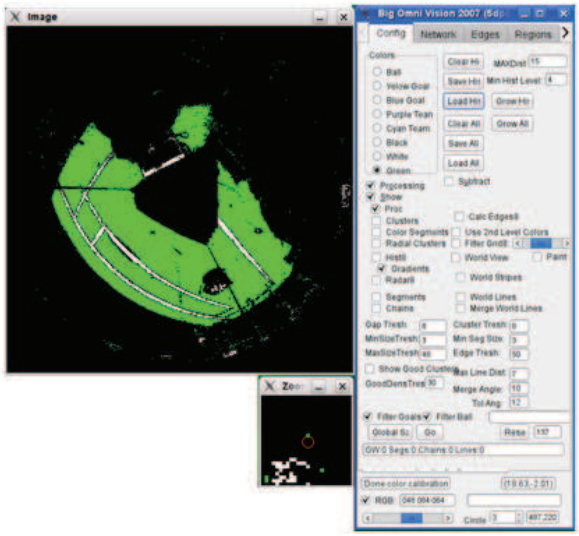

Figure 3.5: The HAL Software

3.3.2.2 The Coach Software

The Coach software (Fig. 3.6) sets the roles of each robot in the team and performs the selection of the "best" ball. In the Coach was inserted the intelligent state machine described in chapter 6. This state machine activates the roles that each robot must execute. The Coach shares the roles that each should take and which ball from the observed balls is the "best", or the true ball. In this same fashion, it collects the information from the robots about their position with respect to the ball.

As it can be seen in Fig. 3.6, the color status is green when the robot is active and ready. It changes to red when the robot is inactive. Furthermore, the Coach map shows the position of each robot in the field the location of the true ball.

Figure 3.6: Coach Application

3.3.2.3 The Dec Software

The Dec software computes each robot's decision. Here are implemented the Roles, Tasks and Actions layers. Through the command received from the Coach using the RTDB, the robot takes a specific Role, and then executes the proper Task and Action programmed for that function (Role). The Dec environment can be seen in Fig. 3.7.

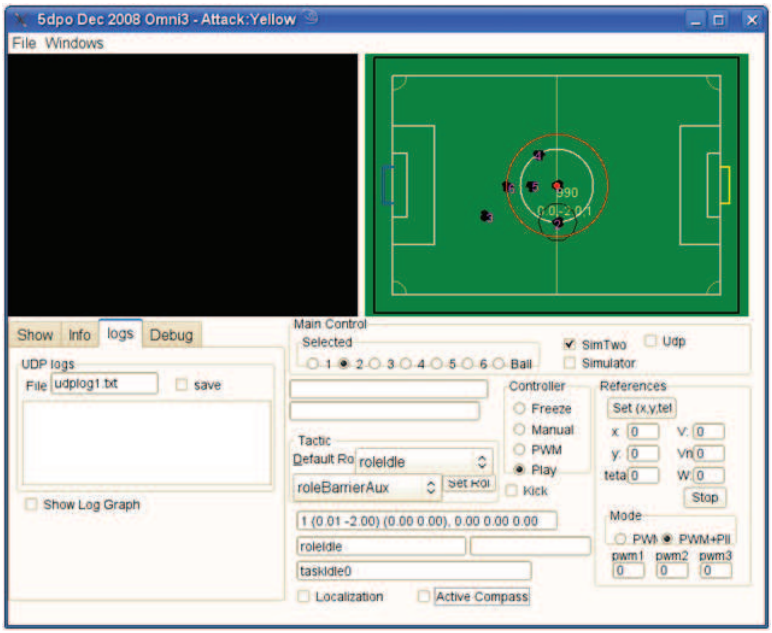

Figure 3.7: The Dec Software

In Dec software all the information from the sensors (odometry, features received from the vision software, compass, etc.) is processed. The program also observes which Role

is given by the Coach. This Role is relevant to a series of tasks yet to be done (path generation, trajectory tracking controlling , obstacle avoidance, kicking strength, etc.).

The Roles are the designed functions for each player. The functions that each player receives are given by the Coach software in agreement with a set of conditions, such as: its physical characteristics, its position in the field, etc. Examples of roles are the following: forward, defense, keeper, etc. In this work only three roles are used: Search, SearchFollower and Formation further explained in chapter 6.

Each roles execute a set of Tasks. Tasks are jobs that each player must execute, depending on its function. The act of passing the ball to a teammate or to block an adversary player are good examples of tasks. Furthermore, each Task means the execution of a set of Actions. The Actions are responsible for the trajectory tracking control, trajectory generation, obstacle detection, kicking strength, etc.

3.4 The SimTwo Simulation Environment

The SimTwo simulator creates the replication of the entities that populate the world, based on a set of *.xml* (Extensible Markup Language) files that are read at the time of the application startup. These states can be read from a script, thus giving rise to the movement of the world.

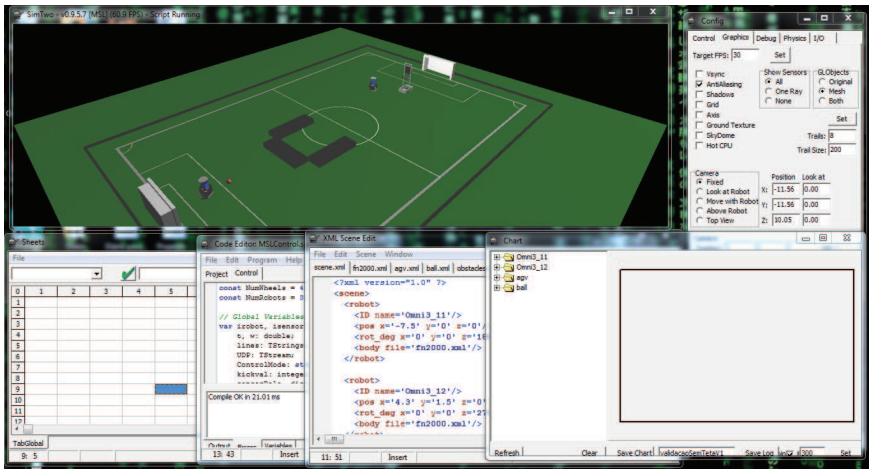

Figure 3.8: The SimTwo Main Windows

The simulation platform has been developed within the group of mobile robotics 5dpo by [116] and it is a free software available for download [117]. This platform shows a high level of realism, especially in terms of dynamic models. These include nonlinear characteristics, which are proven important to the behavior of real objects. The simulator also allows the viewing of real-time 3D simulation, which is beneficial to the supervision of the simulation. A screen shot of the simulator windows is shown in Fig. 3.8. The

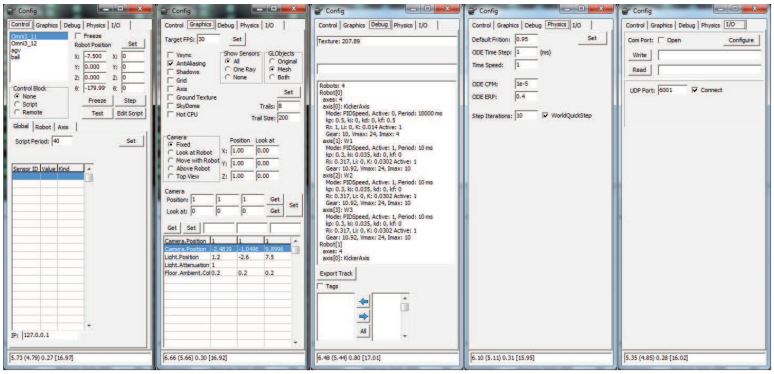

Figure 3.9: SimTwo Configuration Window

3D visualization window is used to overlook the simulation, and the visualization of the parameters defined in the Graphics tab of the configuration window, visible in Fig. 3.9.

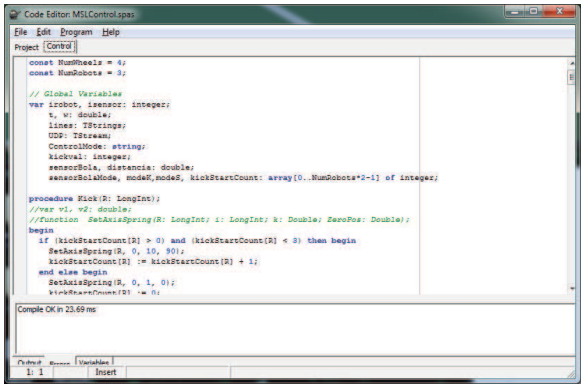

Figure 3.10: SimTwo Code Editor

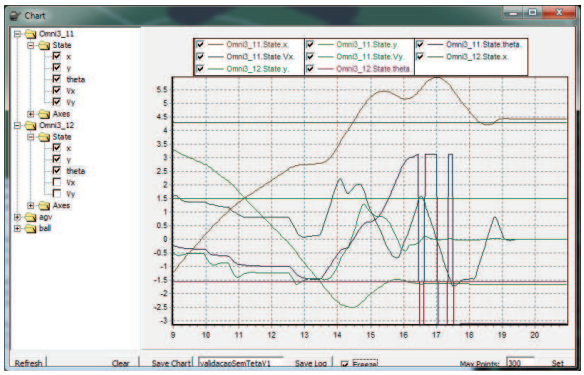

Figure 3.11: SimTwo Plot Window

The control parameters of the robots are arranged in the Control tab of the configuration window. Also, the parameters related to the input and output interfaces are configured in the IO tab. The robot behavior is controlled by a Pascal interpreter, which is included in the simulator using the open source library Pascal script. Using this resource, only the

control of the robots can be implemented in Pascal script. The code can be also used as an interface layer with an external controller (see Fig. 3.10).

The simulator also allows recording the progression of some variables over time, as shown in Fig. 3.11. Finally, Fig. 3.12 shows the window of the XML editor. The 3D graphical facet of the simulator, as well as the input of the physical constrains and characteristics of the robot using ODE (Open Dynamics Engine) library [116], is built in this editor.

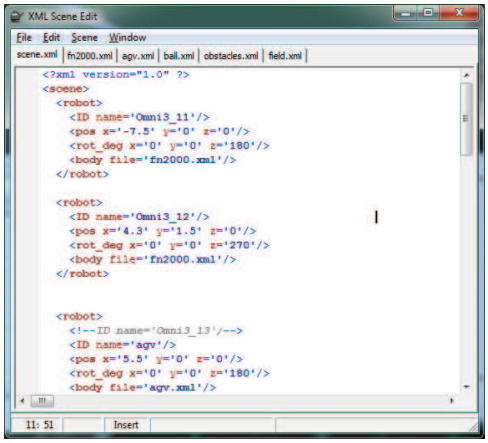

Figure 3.12: SimTwo XML Editor

3.4.1 The ODE Library

The SimTwo uses the ODE (Open Dynamics Engine) library [116] for the simulation of rigid bodies. This library is targeted for the simulation in real time, so that priority is given to the speed and stability compared to the precision. The library enables the implementation of two levels of simulation: the level of dynamic simulation and the level of simulated collisions. These levels are purposely separated in order to maximize flexibility and usefulness of the library. Thus, it is possible to use an external library of simulated collisions with the library of the dynamic simulation, or vice versa.

The simulation of the dynamic component is directly related to the dynamic properties of bodies, such as masses or velocities, while the component simulation of collisions is concerned only of the shape of the bodies. In terms of dynamic simulation, the bodies are devoid of form, being represented by their mass and moments of inertia.

In the simulation of rigid body dynamics there are two types of entities: the *body* entities and the *joint* entities. The *body* entities represent the rigid body, and correspond to arbitrary distributions of mass, represented by its mass and the $3x3$ matrix of the moments of inertia. Torques and forces are exerted on the bodies, which can be applied directly or through junctions with other bodies. *Joint* entities correspond to junctions between bodies, allowing the transmission of forces and torques between them. The junctions

impose restrictions on the movement, and every type of junction corresponds to certain restrictions. When *joint* entities operate as a *hinge*, they acquire restrictions which are similar to those offered by a hinge. These junctions only allow angular movement axis and prevent any type of linear motion.

The simulation of collisions is done separately from the dynamics simulation, and deals with the shape of the bodies. Its function consistes of determining the points of "contact" between objects. These points are used to create special junctions called contact, which are commonly used to prevent objects from penetrating each other. These joints also allow to simulate friction between the contact points, applying the friction forces in the direction perpendicular to the normal surface.

The configuration of the simulator is done through XML files. This format allows the definition of hierarchical information. The structures take the form of objects, placed in a virtual world, and grouped in a hierarchical scheme of sections and subsections. At the top of the configuration hierarchy is the section *scene*, which is inserted as the file *scene.xml*. This section contains all the sections used to define the surrounding world. Within the *scene* section, there are segments which in turn divide the elements of the surrounding world into classes. These segments are as following:

1. Robot: Determines the physical constrains of the robots, namely the graphical body and its physical characteristics. Their behavior (movement) is defined outside in the Code Editor;

2. Obstacles: Define immovable structures or fixed obstacles, their position and graphical body;

3. Things: Movable obstacles are described here with the position and graphical body. Their behavior (movement) is defined outside in the Code Editor;

4. Ball: Defines a movable object's graphical body and its physical characteristics;

5. Field: The size of the field and the drawing on the field are created in this file.

The interaction with the simulation is done through procedures and functions that take into account this hierarchical information. These procedures and functions are placed in the Code Editor and the ones that apply globally to a robot must have the identifying number of the robot as the first argument. Each robot has an identification number, starting at zero, and assigned in the order that it is placed on file *scene.xml*. In turn, each robot can have several controlled axes and several bodies, or solids. The simulator allows the communication with the outside world through a network interface. The communication allows the controller to be totally or partially made out of the simulator, and in this case the program, Pascal script, acts as a link layer. The communication network is carried over a universal data protocol (UDP) configured in the IO tab of the configuration window.

Finally, in SimTwo environment, forces can be added in the actuators generated by modeled electric motors with high realism (i.e. nonlinear friction, dead zone, saturation, current limit, etc.). Furthermore, great variety of sensors can be added which have been modeled with high realism (i.e. noise model, measured distances by laser or infra-red sensors, feedback light sensors, CCD camera, etc.)

3.4.2 The 5dpo Robot Model

In control theory the model of the system is an important part of the solution by influencing the behavior of the controller itself. In robotics there are two models to be considered: the model implemented in the robot's controller [111], [118], and the model implemented in the simulator [119]. For the control to be properly parametrized, both models must behave similarly. Nevertheless, the model in the controller must be much simpler in order to lower the computational cost in the control loop.

In this thesis, the model implemented in the proposed controller is explained in detail in chapter 4. The model of the robot implemented in the simulator, which emulates the real robot, must be much more precise considering also the environmental dynamics. Usually, these dynamics are embedded in an ODE library, which aids in the avoidance of the need to handle the equations directly. In the simulator used (SimTwo) the parameters introduced were:

Table 3.1: 5dpo input parameters in SimTwo

Parameter	Measure	Unity
Robot Mass	26.6	kg
Wheel Mass	0.605	kg
Wheel Radius (r_i)	0.051075	m
Distance between the robot-wheel centers (d)	0.1885	m
Wheel Width	0.042	m
Maximum Allowed Current	3.3	A

The motor PID controller's parameters $k_p = 0.3$, $k_i = 0.035$, $k_d = 0$, $k_f = 0$ and the control period $m_t = 10ms$ have also to be input in the simulator's XML code as parameters of the robot. Finally, there are the motor parameters acquired from the motor's datasheet [120] which can be seen in table 3.2.

Nevertheless, two sets of parameters must also be estimated in order to tune the simulator model correctly: the robot's moment of inertia (in Z, Y and X) and the robot's friction constants (Coulomb and Viscous).

3.4.2.1 The Moments of Inertia

The moments of inertia of a mobile robot must always be considered when analyzing a controller, specially if the robot has high velocities. The study of the 5dpo's main moment

Table 3.2: Motor parameters in SimTwo

Parameter	Measure	Unity
Motor Reduction (N)	12	-
Resistance of the Motor	0.317	*Ohm*
Electric Constant of the Motor	0.0302	Nm/A
Maximum Tension on Motor	24	*V*
Encoder	12548	*PPR*

of inertia (the one with respect to the Z axis) was extensively studied by [121], who found it as being $J_z = 0.705kgm^2$. Nevertheless, the estimation of the other two moments of inertia (J_x and J_y) is much more complex. For such assessment, a 3D geometry is needed for an accurate evaluation of the robot. The fact that such robot has three main parts: the low body, the kicking part and the head is relevant for the estimation. The Fig. 3.13 shows all three sections and their respectively heights.

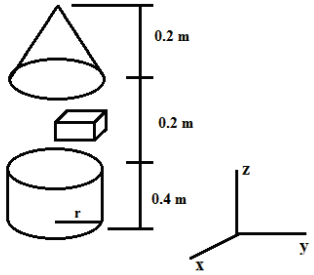

Figure 3.13: Geometric Representation of the Robot's Body

The cube has a length of $x = 0.3m$ and a width of $y = 0.2m$. Furthermore, it is relevant to remember that the cube is positioned slightly towards the front of the robot, which causes the robot's asymmetry. Another key information is the robot radius (the distance between the center of the robot and the outside border part of the robot's chassis), which in this case is $0.25m$. The first step to a successful calculation of the cylinder is to know that

$$J_x = J_y = (\frac{1}{12}.m_{cylinder}).(3.r^2 + h^2_{cylinder}) \tag{3.1}$$

Then, the cone uses

$$J_x = J_y = (\frac{3}{5}.m_{cone}).(\frac{r^2}{4} + h^2_{cone}) \tag{3.2}$$

Finally, there is the third part, or the cube, which uses

$$J_x = (\frac{1}{12}.m_{cube}).(h^2_{cube} + x^2) \tag{3.3}$$

and

$$J_y = (\frac{1}{12}.m_{cube}).(h_{cube}^2 + y^2) \tag{3.4}$$

Now, replacing the names with their values in the four formulas above and knowing that $m_{cylinder} = 13.7kg$, $m_{cone} = 5.5kg$ and $m_{cube} = 7kg$, the following moments of inertia can be calculated:

Table 3.3: Partial Moments of Inertia

Part	$J_x(kgm^2)$	$J_y(kgm^2)$
Cylinder	0.39673	0.39673
Cone	0.18356	0.18356
Cube	0.049058	0.078225

Applying the superposition principle for the moment of inertia in each axis, it results in $J_x = 0.629$ and $J_y = 0.658$. Note that both values agree with the value of J_z when applying the Perpendicular Axis Theorem.

3.4.2.2 The Friction Constants

The estimation of both Coulomb and Viscous friction coefficients used in the dynamic model can follow the experiments done by [111] and [122]. However, to parameterize the simulator, only the wheel's Viscous and Coulomb friction coefficients are needed. The estimation of the wheel's friction constants can be also based on the fact that

$$\frac{d\omega_r(t)}{dt} = \frac{\mu_r}{J_{r_z}}.\omega_r(t) - \frac{C_r}{J_{r_z}} \tag{3.5}$$

where $\omega_r(t)$ is the rotation speed of the wheel (output from the gear box).

Reproducing the simulations that were made in [111], the relationship between both friction coefficients (μ_r and C_r) and the wheel's moment of inertia in z (J_{r_z}) where approximately,

$$\frac{\mu_r}{J_{r_z}} = 5.62(N/kg.m) \quad and \quad \frac{C_r}{J_{r_z}} = 43.9(N/kg.m) \tag{3.6}$$

It is known from the motor's data sheet that $J_{r_z}N^2 = 0.0000138Kgm^2$, where N is the gear box reduction ratio. Therefore,

$$\begin{cases} \mu_r = 7.76x10^5(N.m/rad/s) \\ C_r = 6.06x10^4(N.m) \end{cases} \tag{3.7}$$

3.5 Model Validation

In this section, the methodology of the 5dpo robot model validation is presented. The 5dpo robot was set to reach two points in the field, performing a difficult trajectory in a desired velocity (1m/s) to validate the model. A PI (Proportional plus Integral) controller was used for the motors' velocity in this two sets of experiments [113]. According to Fig. 3.14, the trajectory determined for the performance of these tests is L-shaped.

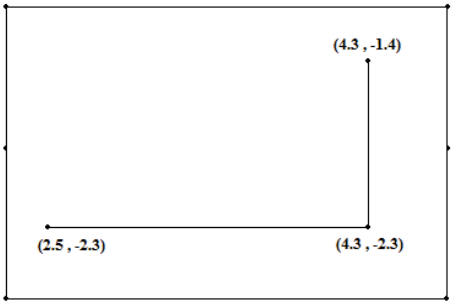

Figure 3.14: Trajectory for Model Validation

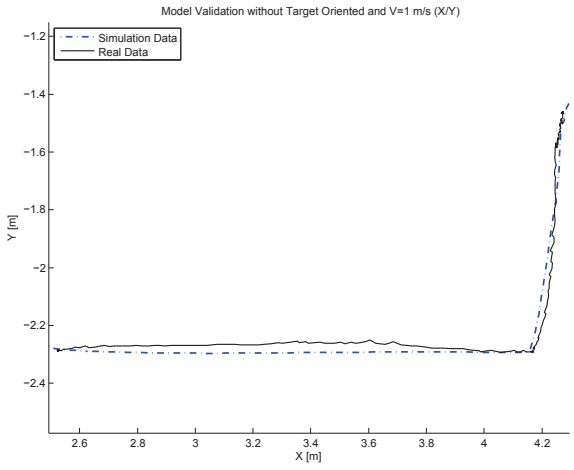

Figure 3.15: Test with $1m/s$ and $\theta = 0^\circ$

The 5dpo robot was set to perform the above described trajectory ten times. The ten performances were divided into two sets. A roof camera was used as ground truth (real position data), for its precision in the world frame position of the robot is much higher than the one of the robot's sensors (the error of the roof camera is less than $0.5cm$). The first set of tests were conducted without varying the robot's angle (with $\theta = 0^\circ$). The average result achieved from the first set of tests can be seen in Fig. 3.15.

In Fig. 3.15, the comparison between the behavior of the robot in SimTwo environment (blue traced line) and the real robot environment (black full line) can be witnessed.

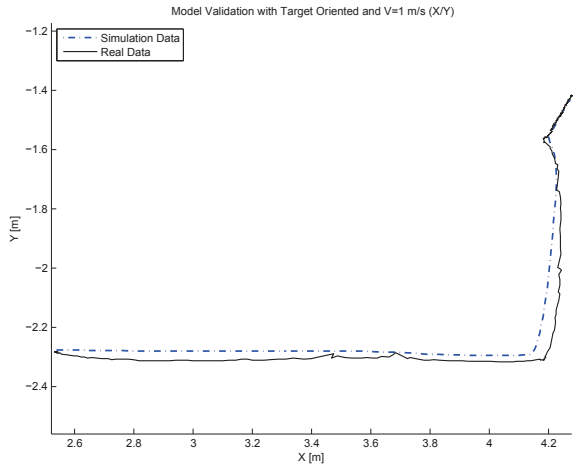

Figure 3.16: Test with $1m/s$ and $\theta = Variable$

The behavior of the two results (simulated and real), disregarding control issues, is similar; this fact can be witnessed by analyzing the proximity of both trajectories.

The second set of tests were conducted while variating the θ in a way that the robot faced its moving direction. The accuracy of the model can be noted also in Fig. 3.16. This figure also shows the comparison between the behavior of the robot in SimTwo environment (blue traced line) and the real robot environment (black full line). The proximity of both trajectories is a point which is relevant to be observed in both figures.

Although the graphs show the proximity between the trajectories, formal analysis must be performed to compare simulation and real results in both groups. Therefore, the mean square error was used to analyze the model behavior. After small adjustments in the parameters, the minimum values found during all the tests for both groups can be seen in table 3.4.

Table 3.4: Mean Square Error for Model Validation Trajectories

$Trajectory$	**MSE in x**	**MSE in y**
$\theta = 0$	0.1368	0.0428
$\theta = varing$	0.0856	0.0265

This model was then used in the simulations with the controller which can be seen in the following chapters.

3.6 Conclusion

This chapter presented the mobile robots used in this research: the middle size league omnidirectional 5dpo robot and the differential mobile robot. A detailed description of the simulator SimTwo was also presented while demonstrating its advantages. Finally,

the 5dpo mobile robot model implemented in the SimTwo simulator was validated by comparing the simulations with real experiments with the real robot.

Chapter 4

Formation Control in Active Target Tracking

This chapter describes a novel approach in formation control for mobile robots in the active target tracking problem. A nonlinear model predictive formation controller (NMPFC) for target perception was implemented to converge a group of mobile robots towards a desired target. The team must also maintain a desired formation following a target while it is moving, or follow a leader in the case of target absence. The structure details of the controller, as well as a mathematical analysis of the formation model used, are presented in this chapter. As a final objective of this chapter, the weight tuning is addressed in order to minimize an objective function that reflects the controller's efficiency with respect to a given criteria. Furthermore, results of simulations and experiments with real robots are presented and discussed.

4.1 Introduction

The main objective of this chapter is to present a novel nonlinear model predictive formation control (NMPFC). This work is inserted in the active target tracking [27], [28], [29], [30] and formation control problems [32], [33], [13], [34], [35], [36]. Therefore, this controller was conceived to converge and keep the formation of a mobile robot team towards a target. The examples and tests will be performed by using a homogeneous group of robots formed from 5dpo omnidirectional mobile soccer robots [123] (Fig. 3.2), and also using heterogeneous group of robots formed from 5dpo holonomic robots and the nonholonomic robot seen in Fig. 3.3. This controller can be applied to N robots in formation.

Most of the past and current work on motion coordination of multiple (possibly heterogeneous) vehicles focuses on controlling a vehicle formation with a given nominal geometry and a pre-determined trajectory or a static destination location, possibly compliant with the presence of obstacles on the formation trajectory [95]. Such methods typically:

- assume full knowledge of the formation state, expressed as the relative distances and bearings among all the vehicles, and/or
- rely on local memory-less interactions, often jeopardizing global formation stability.

The formation used in this thesis differs from the usual rigid formation where the relative pose of a team element must be precisely maintained. Here, the ideal configurations are the ones that minimize a cost function where it penalizes the total amount of uncertainty in the team perception of an element (target), converges the robots towards this element in a desired pose, avoids collisions with mates and obstacles and penalizes oscillations in the control effort.

A vehicle formation is supposed to serve one or more mission objectives [124], such as search rescue missions and land mine removal. One such interesting case concerns localizing or tracking relevant objects, here and henceforth denominated as targets. Available formation control methods often give little relevance to the requirements imposed by target localization and/or tracking of the formation geometry, so as to improve the target detection and tracking quality (e.g., accuracy). Many control strategies have been investigated in order to solve formation control problems (e.g., [20], [96] and [125]), some of which present nonlinear model predictive control (NMPC) solutions.

The NMPC is based on a finite horizon continuous time minimization of nonlinear predicted tracking errors with constraints on the control inputs and the state variables. It predicts system outputs based on current and future states using a model of the system. Then, it finds an open-loop control profile by numerical optimization, and applies the first control signal in the optimized control profile to the system. However, due to the use of a finite horizon, control stability becomes one of the main problems. To guarantee control stability, many approaches have been investigated, e.g., using terminal region constraints and/or a terminal penalty term [126], [127].

In [98], a two layers predictive controller that controls the formation of nonholonomic mobile vehicles was proposed. In their study, the authors considered that there are two sub-problems to be solved to fulfill the main goal: the trajectory control problem and the formation control problem. To solve the first sub-problem a nonlinear controller was proposed to control the trajectory, while a linear model predictive controller was proposed to solve the second sub-problem and control the formation.

Moreover, the authors in [128] propose a distributed model predictive control scheme based on a cooperative game in which two different agents communicate in order to find

a solution to the problem of controlling two constrained linear systems coupled through the inputs. They assume that each agent only has partial information of the model and the state of the system. In their proposed scheme, the agents communicate twice each sampling time in order to share enough information to take a cooperative decision. The theoretical results and the design procedure are illustrated using two different examples.

Usually, the state of the art researchers study rigid formation on teams of robots. In said studies [58], the relative poses between robots are fixed. This chapter presents an approach where the desired formation is not rigid due to the need for the robots to reposition and reorient themselves to better estimate a target and simultaneously avoid mates, obstacles and so on.

Therefore, the problem addressed in this chapter lies in the frontier between the formation control problem and the active target tracking problem. After a careful analysis of the state of the art, the nonlinear model predictive control theory, as stated and explained in chapter 2, was considered as the main theory approach of this work. This chapter is then structured as follows: the next section explains the adopted control architecture. The nonlinear model predictive formation control is presented and explained in section 4.3, the optimization algorithm is presented in section 4.4, the following section presents a methodology for weight tuning. The results are then presented in section 4.6 and the conclusion is presented in the last section.

4.2 Control Architecture

The nonlinear model predictive control is usually implemented in a centralized fashion. It holds full knowledge of the entire system and computes all the control outputs. A centralized control using a non-convex optimization scheme applied in large-scale interconnected systems, such as water distribution systems, traffic and power systems, manufacturing systems and economic systems, may be a too complex solution or not even feasible. With the quick development of communication networks, centralized control has been gradually replaced by distributed control such as in multi-robot systems and applications in manufacturing and process industries where multiple units cooperatively produce a good. In distributed control schemes, agents share information in order to improve closed-loop performance, robustness and fault-tolerance [128]. In the approach described in this thesis, the agents (robots) share their local measurements (states) and receive the other agents' states, computing a control input using a reduced order model of the formation system dynamics. The challenge in this case is to formulate a simpler and decentralized problem which leads to a behavior that is similar to the one obtained using a centralized approach [38].

The objective of this chapter is to formulate a nonlinear model predictive formation controller (NMPFC) for a multi-robot systems formation control. The general structure

of a coordinated multi-robot system can be classified in three categories: distributed, centralized, or hybrid (partially distributed). These classifications are based on how the control signals of each robot are calculated. In this case, the control architecture is fully distributed [128].

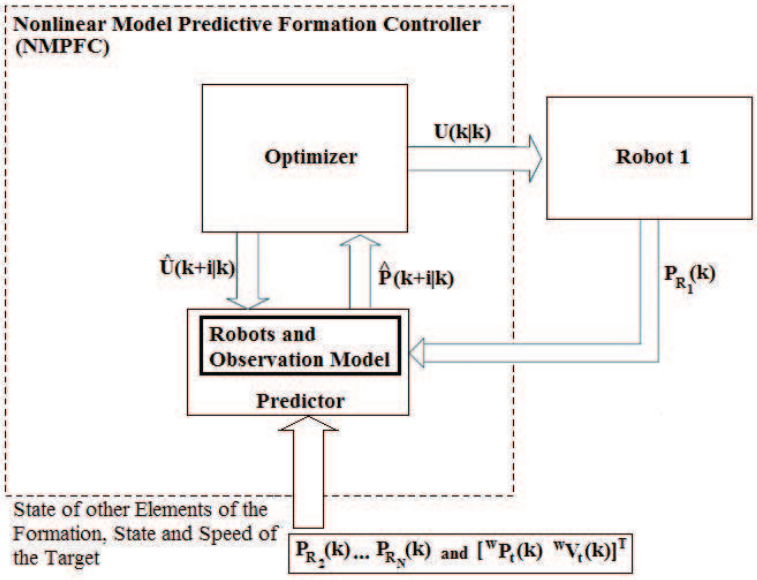

Figure 4.1: Structure of the Nonlinear Model Predictive Formation Controller Applied to robot 1

Fig. 4.1 illustrates the structure of the NMPFC used in this work, where $U(k|k) = U(k) = \begin{bmatrix} v_{ref}(k) & vn_{ref}(k) & w_{ref}(k) \end{bmatrix}^{\mathrm{T}}$ is the output control (see also Fig. 4.3) signal in the first prediction step, $\hat{U}(k+i|k)$ with $i = 0...N_c - 1$ is the output control signal from the optimizer sent to the predictor, and $\hat{P}(k+i|k)$ with $i = 1...N_p$ is the response of the predictor block to each $\hat{U}(k+i|k)$. Here, at an instant k, robot 1 (R_1) sends its pose $P_{\mathrm{R}_1}(k) = \begin{bmatrix} x_{\mathrm{R}_1}(k) & y_{\mathrm{R}_1}(k) & \theta_{\mathrm{R}_1}(k) \end{bmatrix}^{\mathrm{T}}$ to the NMPFC. Furthermore, the NMPFC also receives the other robots' poses $[P_{R_2}(k)...P_{R_N}(k)]$, the position of the target t in the world frame ${}^{w}P_{\mathrm{t}}(k) = \begin{bmatrix} {}^{w}x_{\mathrm{t}}(k) & {}^{w}y_{\mathrm{t}}(k) \end{bmatrix}^{\mathrm{T}}$ and the velocity of the target t in the world frame ${}^{w}V_{\mathrm{t}}(k) = \begin{bmatrix} {}^{w}v_{x\mathrm{t}}(k) & {}^{w}v_{y_{\mathrm{t}}}(k) \end{bmatrix}^{\mathrm{T}}$.

The NMPFC's ability to create and maintain a formation is due to the fact that the cost functions used by the controllers of each robot in the team are coupled. The above mentioned coupling occurs when the teammates' states are used in the cost function of each robot's controller to penalize the geometry or the deviation from the desired objective. This means that the actions of each robot affect every other teammate. Each robot keeps the formation state (pose and speed of the robots in formation, and position and speed of any target that should be followed), updating them in each control loop. This information is received by the controller of each robot in the formation which in turn creates the formation geometry where the actions of each robot affect the other teammates. The

NMPFC can be divided into two sub-blocks:

- **Optimizer** - This sub-block uses an online numeric optimization method to minimize the cost function and generate the signals of optimal control. The resilient propagation (RPROP) method is used here and it guaranties quick convergence [99];

- **Predictor** - The predictor performs the state evolution of the robot itself, the teammates and the target based on pre-defined models. It uses a simplified dynamic model to emulate the robot's evolution. The velocities of the teammates and target are assumed to be constant and equal to the last known velocities during the entire prediction horizon. The obstacles (moving or static) are assumed to have zero velocities during the control loop and the evolution of the relative distance between the obstacle and the robot is predicted. The predictor also emulates the evolution of the target's merged state covariance matrix.

After receiving the states of the robot, teammates, obstacles and target, the controller's optimizer sub-block provides the control input $\hat{U}(k+i|k)$, in a limited control horizon, to the predictor sub-block, which then predicts the formation state evolution $\hat{P}(k+i|k)$ for N_p steps (prediction horizons), and provides a cost value to the optimizer in accordance with $\hat{U}(k+i|k)$. The iterative minimization process is repeated in cyclic fashion. Finally, the control output in the first step $U(k)$ is sent to the robot.

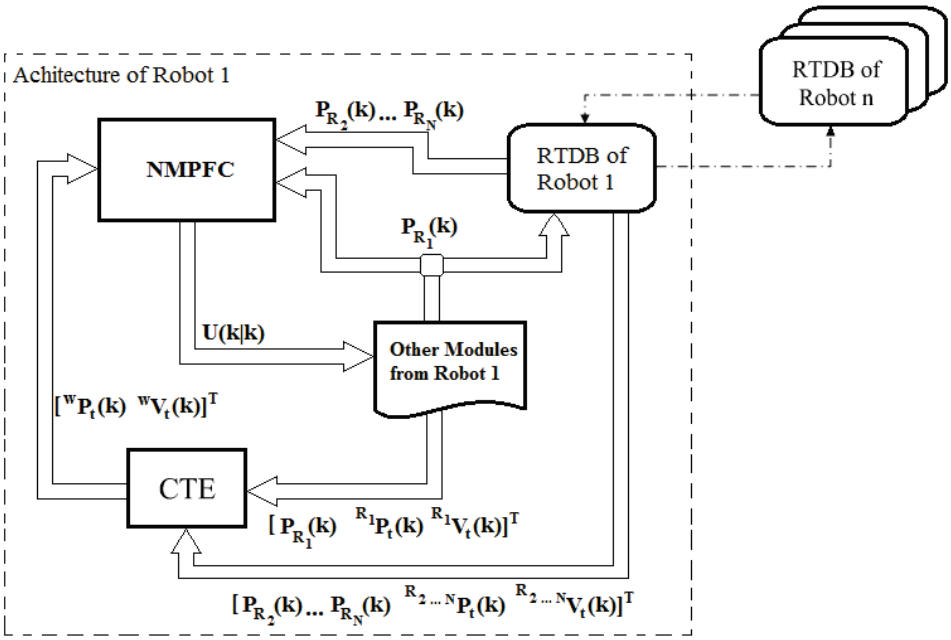

Figure 4.2: Controller Diagram

4.3 Nonlinear Model Predictive Formation Control

The nonlinear model predictive formation controller (NMPFC) proposed in this study was used to control the formation of a multi-robot system in the active target tracking problem.

Fig. 4.2 shows the block diagram of the proposed formation control framework for robot R_1 in a formation with N robots. Here, the subscripted R_n is used to denote the robot n where $1 \leq n \leq N$ and N is the total number of robots in formation and the subscripted t to denote the target. Each robot has a NMPFC, a cooperative target estimator (CTE) [129] and a real time data base (RTDB) communication application [115]. Other functions such as localization and the vision system are embedded in other software modules represented here as the block *Other Modules from Robot 1*. At each instant k the robot R_1 sends its pose ($P_{R_1}(k)$) to the controller and to the RTDB to be shared with the formation. The robot also sends its pose ($P_{R_1}(k)$), the position of the target in its local frame (${}^{R_1}P_t(k)$) and the velocity of the target in its local frame (${}^{R_1}V_t(k)$) to the CTE. Then, the CTE, represented as a single block in Fig. 4.2, receives from the RTDB the pose, as well as the position and velocity of the target in the local frame of each other robot in formation, and also communicates locally to the NMPFC the information on the fused target position (${}^{w}P_t(k)$) and the fused target velocity (${}^{w}V_t(k)$), both in the world frame. An additional layer in communication using a real time data base (RTDB) shared memory makes it possible to convey information. Then, each robot writes its own pose and reads the teammates'. The NMPFC also receives the pose of each robot in formation ($[P_{R_2}(k)...P_{R_N}(k)]$) from the robot's RTDB and then sends an output control signal $U(k)$ to the robot (the tangential and normal components of linear velocity and angular velocity).

The robot's pose and velocity estimation is performed using a localization algorithm that receives data from the odometry, from a digital compass, and from the omnidirectional camera. Then, using the omnidirectional camera to detect the white lines of the field combined with a map matching algorithm, the robot is localized. During the movement, the velocity is estimated also using data from odometry.

To achieve convergence in the formation, and hence cost function minimization, the NMPFC's predictor sub-block produces the evolution of the formation's behavior, as well as the behavior of the target's merged state and covariance matrix which is used by the NMPFC's optimizer and predictor sub-blocks for the cyclic minimization process. After processing the control calculations, the NMPFC sends the desired control output back to the robot (controller's reference velocities).

In the NMPFC control mode, the goal is to create and maintain a formation of multiple robots in order to improve the merged perception of the target by the team while avoiding teammates, obstacles and so on. The strategy consists of having each robot aware of the current pose of every teammate in the formation, while configured in a fully distributed architecture. Similarly to the strategy used in trajectory tracking, here:

- Each robot predicts the evolution of the entire formation, by using the simplified nonlinear dynamic model for itself and purely kinematic models for the other robots or moving targets in the formation;
- The cost function is minimized in order to find the control outputs that keep the robot in formation;
- The first control input is applied to the robot and the process is repeated when new measurements are available.

Each formation is defined by a set of cross-coupled cost functions which penalize deviations from the desired formation. One of these cost functions is attributed to each robot in the formation, which propels the NMPFC controller to generate its own control inputs. The following subsection describes the model used in the predictor to perform the evolution of the formation during each step (prediction horizon).

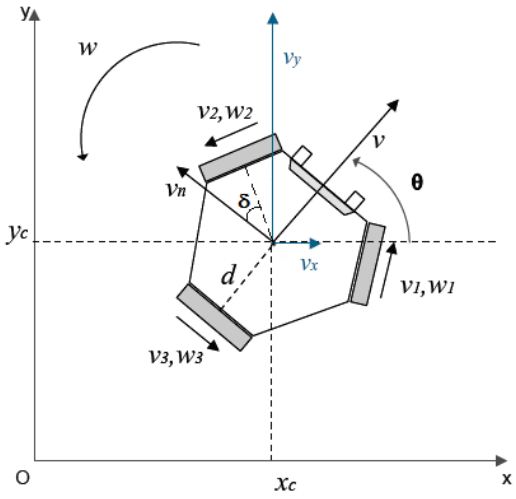

Figure 4.3: The 5dpo robot graph representation.

4.3.1 The Model

In a robot R_n (where $1 \leq n \leq N$ and N is the total number of robots in the formation) the wheel angular velocities, $\omega_{rR_n}(k) = \begin{bmatrix} \omega_1(k) & \omega_2(k) & \omega_3(k) \end{bmatrix}^{\mathrm{T}}$ is given by the odometry and the wheel linear velocity is given by $V_{\mathrm{rR}_n}(k) = \begin{bmatrix} v_1(k) & v_2(k) & v_3(k) \end{bmatrix}^{\mathrm{T}} = r.\begin{bmatrix} \omega_1(k) & \omega_2(k) & \omega_3(k) \end{bmatrix}^{\mathrm{T}}$. Through a geometric analysis of Fig 4.3 the robot velocities can be found using the equation below.

$$\begin{bmatrix} v_{\mathrm{R}_n}(k) \\ vn_{\mathrm{R}_n}(k) \\ w_{\mathrm{R}_n}(k) \end{bmatrix} = (B)^{-1}. \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} \tag{4.1}$$

with

$$B = \begin{bmatrix} cos(\delta) & sin(\delta) & d \\ -cos(\delta) & sin(\delta) & d \\ 0 & -1 & d \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & d \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & d \\ 0 & -1 & d \end{bmatrix} \tag{4.2}$$

where $\delta = 30°$, resulting:

$$(B)^{-1} = \begin{bmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & -\frac{2}{3} \\ \frac{1}{3d} & \frac{1}{3d} & \frac{1}{3d} \end{bmatrix} \tag{4.3}$$

The omnidirectional mobile robot model used in the **Predictor** illustrated in Fig. 4.1 is a nonlinear simplified model that, when properly parameterized, can reduce the computational cost of each cycle of the control algorithm. This translates into the ability to use greater prediction horizons in the predictive controller, even in computers with modest specifications, keeping the cycle time control within the required limits. The model is initialized with the limitation of the motor's velocity by detecting saturation and proportionally scaling the other motors (which can be seen as an input constraint) [130]. Furthermore, the prediction of the wheels' velocities of robot R_n follows a first order discrete model, where:

$$\begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} = a. \begin{bmatrix} v_1(k-1) \\ v_2(k-1) \\ v_3(k-1) \end{bmatrix} + b.(B^T). \begin{bmatrix} v_{ref_{\mathrm{R}_n}}(k) \\ vn_{ref_{\mathrm{R}_n}}(k) \\ w_{ref_{\mathrm{R}_n}}(k) \end{bmatrix}$$

and

$$a = e^{-\frac{t_s}{t_p}}, \quad b = (1-a) \tag{4.4}$$

with $t_s = 0.01$ as the sampling period and $t_p = 0.04$ as the plant time constant. Furthermore, a and b are coefficients in the range from 0 to 1, and coefficient a tends to be close to 1.

Then, the result is once more inserted in the equation (4.1) to be computed. Therefore, its state (pose $(P_{\mathrm{R}_n}(k))$ and velocity $(V_{\mathrm{R}_n}(k))$) are defined as:

$$P_{\mathrm{R}_n}(k) = \begin{bmatrix} x_{\mathrm{R}_n}(k) & y_{\mathrm{R}_n}(k) & \theta_{\mathrm{R}_n}(k) \end{bmatrix}^{\mathrm{T}} \quad V_{\mathrm{R}_n}(k) = \begin{bmatrix} v_{x\mathrm{R}_n}(k) & v_{y_{\mathrm{R}_n}}(k) & w_{\mathrm{R}_n}(k) \end{bmatrix}^{\mathrm{T}} \tag{4.5}$$

and the simulation of the state evolution is given by:

$$\begin{bmatrix} x_{\mathrm{R}_n}(k) \\ y_{\mathrm{R}_n}(k) \\ \theta_{\mathrm{R}_n}(k) \end{bmatrix} = \begin{bmatrix} x_{\mathrm{R}_n}(k-1) \\ y_{\mathrm{R}_n}(k-1) \\ \theta_{\mathrm{R}_n}(k-1) \end{bmatrix} + t_s . \begin{bmatrix} v_{x\mathrm{R}_n}(k) \\ v_{y_{\mathrm{R}_n}}(k) \\ w_{\mathrm{R}_n}(k) \end{bmatrix}$$

and

$$\begin{bmatrix} v_{x\mathrm{R}_n}(k) \\ v_{y_{\mathrm{R}_n}}(k) \\ w_{\mathrm{R}_n}(k) \end{bmatrix} = \begin{bmatrix} cos(\theta_{\mathrm{R}_n}(k)) & -sin(\theta_{\mathrm{R}_n}(k)) & 0 \\ sin(\theta_{\mathrm{R}_n}(k)) & cos(\theta_{\mathrm{R}_n}(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} v_{ref_{\mathrm{R}_n}}(k) \\ vn_{ref_{\mathrm{R}_n}}(k) \\ w_{ref_{\mathrm{R}_n}}(k) \end{bmatrix}$$

Taking into account the elements presented in Fig. 3.2, the position $P_{\mathrm{t}}(k)$ and velocity $V_{\mathrm{t}}(k)$ of the target t (ball) in the world frame in a instant k are defined as:

$$P_{\mathrm{t}}(k) = \begin{bmatrix} x_{\mathrm{t}}(k) & y_{\mathrm{t}}(k) \end{bmatrix}^{\mathrm{T}} \quad V_{\mathrm{t}}(k) = \begin{bmatrix} v_{x\mathrm{t}}(k) & v_{y_{\mathrm{t}}}(k) \end{bmatrix}^{\mathrm{T}} \tag{4.6}$$

where

$$\begin{cases} x_{\mathrm{t}}(k) = x_{\mathrm{t}}(k-1) + t_s.(v_{x\mathrm{t}}(k)) \\ y_{\mathrm{t}}(k) = y_{\mathrm{t}}(k-1) + t_s.(v_{y_{\mathrm{t}}}(k)) \end{cases} \tag{4.7}$$

and

$$\begin{cases} v_{x\mathrm{t}}(k) = v_{x\mathrm{t}}(k-1).B_{FC} \\ v_{y_{\mathrm{t}}}(k) = v_{y_{\mathrm{t}}}(k-1).B_{FC} \end{cases} \tag{4.8}$$

where B_{FC} is the ball friction coefficient.

The target's velocity unit vector is then defined as:

$$\tilde{V}_t(k) = \begin{bmatrix} \tilde{v}_{x\mathrm{t}}(k) & \tilde{v}_{y_{\mathrm{t}}}(k) \end{bmatrix}^{\mathrm{T}} = \frac{V_{\mathrm{t}}(k)}{||V_{\mathrm{t}}(k)||}, \tag{4.9}$$

where $||\cdot||$ represents the euclidean norm.

The position of the target relative to the robot R_n in a instant k is defined as:

$$P_{\mathrm{t}}^{\mathrm{R}_n}(k) = \begin{bmatrix} x_{\mathrm{t}}^{\mathrm{R}_n}(k) & y_{\mathrm{t}}^{\mathrm{R}_n}(k) \end{bmatrix}^{\mathrm{T}} \tag{4.10}$$

where

$$\begin{cases} x_{\mathrm{t}}^{\mathrm{R}_n}(k) = x_{\mathrm{t}}(k) - x_{\mathrm{R}_n}(k) \\ y_{\mathrm{t}}^{\mathrm{R}_n}(k) = y_{\mathrm{t}}(k) - y_{\mathrm{R}_n}(k) \end{cases} \tag{4.11}$$

The unit vector that indicates the direction of the target with respect to the robot is defined as:

$$\tilde{P}_{\mathrm{t}}^{\mathrm{R}_n}(k) = \left[\begin{array}{cc} \tilde{x}_{\mathrm{t}}^{\mathrm{R}_n}(k) & \tilde{y}_{\mathrm{t}}^{\mathrm{R}_n}(k) \end{array} \right]^{\mathrm{T}} = \frac{P_{\mathrm{t}}^{\mathrm{R}_n}(k)}{||P_{\mathrm{t}}^{\mathrm{R}_n}(k)||} \tag{4.12}$$

The bearing of the target with respect to robot R_n is defined as:

$$\theta_{\mathrm{t}}^{\mathrm{R}_n}(k) = arctan2(y_{\mathrm{t}}^{\mathrm{R}_n}(k), x_{\mathrm{t}}^{\mathrm{R}_n}(k)) \tag{4.13}$$

The pose of a robot R_n relative to its mate R_j (where $1 \leq j \leq N$, with $j \neq n$, and as a matter of simplicity it is represented in the sums by $1 \leq j \leq NM$, and where $NM = N-1$ is the total number of mates) is defined as:

$$P_{\mathrm{R}_n}^{\mathrm{R}_j}(k) = \left\{ \begin{array}{l} x_{\mathrm{R}_n}^{\mathrm{R}_j}(k) = x_{\mathrm{R}_n}(k) - x_{\mathrm{R}_j}(k) \\ y_{\mathrm{R}_n}^{\mathrm{R}_j}(k) = y_{\mathrm{R}_n}(k) - y_{\mathrm{R}_j}(k) \end{array} \right. \tag{4.14}$$

and with respect to an obstacle O_l (where $1 \leq l \leq NO$, and NO is the total number of obstacles), is defined as:

$$P_{\mathrm{R}_n}^{\mathrm{O}_l}(k) = \left\{ \begin{array}{l} x_{\mathrm{R}_n}^{\mathrm{O}_l}(k) = x_{\mathrm{R}_n}(k) - x_{\mathrm{O}_l}(k) \\ y_{\mathrm{R}_n}^{\mathrm{O}_l}(k) = y_{\mathrm{R}_n}(k) - y_{\mathrm{O}_l}(k) \end{array} \right. \tag{4.15}$$

It is important to mention that in the obstacle's state evolution, all obstacles (moving or static) are considered as having zero velocity at that instant in order to reduce the computation load as the number of obstacles increases.

The model used to predict the nonholonomic Mini-AGV's behavior can be found in [131]. Moreover, the dynamic model of the actuators in the nonholonomic robot follows the same example of the one used in [20].

Finally, in order to model the evolution of the total amount of uncertainty with respect to the relative position between the robot and the target, a covariance model was created based on Fig. 4.4. This model depends on the type of camera used (such as omnidirectional mirror-camera, fish eye camera, normal direct cameras). However, the model can be easily changed using the controller's code, allowing each robot to have a different sensor.

In the omnidirectional 5dpo robot camera, the point on the ground plane directly below the robot's catadioptric system center (also the robot's geometric center) is assumed to be the origin O of the coordinates for the model discussed in this sub-section. The target observation M_{R_n} is represented as a 2D vector $[d_t, \phi]$, where d_t ($d_t \geq 0$, $d_t \in \mathbb{R}$) is the

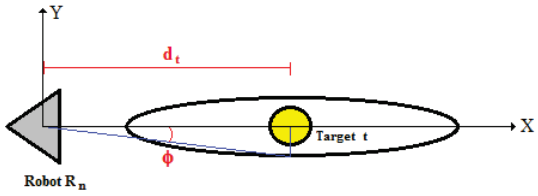

Figure 4.4: Model of Observation

distance to the target from O, and ϕ ($-\pi \leq \phi \leq +\pi, \phi \in \mathbb{R}$) is the bearing of the target from the positive X axis of the robot. The covariance model of a robot R_n in the instant k is given by (4.16).

$$\Sigma_{\mathrm{R}_n}(k) = \begin{bmatrix} \sigma_{d_t}^2 & \rho\sigma_{d_t}\sigma_{\phi} \\ \rho\sigma_{\phi}\sigma_{d_t} & \sigma_{\phi}^2 \end{bmatrix} \tag{4.16}$$

where σ_{d_t} is the variance of the target's distance measurement d_t, σ_{ϕ} is the variance of the target's bearing measurement and ρ is the correlation coefficient. It is assumed that both measurements are uncorrelated and $\rho = 0$.

In the case of the 5dpo robots, an empirical and simplified observation covariance model (4.17) was created where the variance in the target distance is directly proportional to the distance squared and the variance in the target bearing is inversely proportional to the target distance. This observation model was created in a simplified fashion in order to maintain a low computational cost and it was validated with several experiments in which it was also possible to find the values of K_a and K_b.

$$\Sigma_{\mathrm{R}_n}(k) = \begin{bmatrix} K_a d_t^2 & 0 \\ 0 & K_b \frac{1}{d_t} \end{bmatrix} \tag{4.17}$$

Furthermore, it is necessary to represent the observation covariance (4.17) in its canonical form in the Cartesian coordinates centered at O due to the need of ease up the covariances merging that arise from the teammates. Therefore, the canonical representation of the covariance model in the direction of the target and in its perpendicular direction is given by (4.18). The statistical development from equation (4.17) to equation (4.18) can be seen in appendix A.

$$\Sigma_{\mathrm{R}_n}^{\perp}(k) = \begin{bmatrix} K_1 d_t^2 & 0 \\ 0 & K_2 d_t \end{bmatrix} = \begin{bmatrix} {\sigma_x}^2 & 0 \\ 0 & {\sigma_y}^2 \end{bmatrix} \tag{4.18}$$

where $K_1 = K_a$ and $K_2 = (K_b + K_a K_b)$ are constants of proportionality.

The covariance merging is performed using Smith and Cheeseman's formulation [132]. In the formation, each teammate's covariance $\Sigma_{\mathrm{R}_j}^{\perp}(k)$ is also predicted. The teammate's predicted covariance is rotated in the robot's frame and then merged as per the method

presented in [132]. Therefore, given N robots in a formation, where R_n is the robot predicting its formation covariances and robots R_j are the teammates, with $1 \leq j \leq NM$ and $NM = N-1$ is the number of teammates in formation. The merged covariances are given by

$$\Sigma^{\perp}_{\text{Merged}}(k) = ([\Sigma^{\perp}_{\text{R}_n}(k)]^{-1} + [\hat{\Sigma}^{\perp}_{\text{R}_{j=1}}(k)]^{-1} + ... + [\hat{\Sigma}^{\perp}_{\text{R}_{j=NM}}(k)]^{-1})^{-1} \tag{4.19}$$

where $\hat{\Sigma}^{\perp}_{\text{R}_j}(k)$ are the covariance rotated matrices of the mates with respect to the robot R_n.

No noise was introduced in simulation experiments. Therefore, the means of observation estimates from teammates are identical while the uncertainty ellipse around each teammate's observation is formulated as per (4.18) for merging. Furthermore, it is assumed that the same vision system is applied to the nonholonomic robot, where an omnidirectional camera is placed in the robot's geometric center.

4.3.2 The Cost Function During the Presence of a Target

The cost function of a NMPC (here NMPFC) represents the cost to be minimized by the predictive controller. It is typically associated to the dynamical change of the system (formation geometry) over time. Therefore, it reduces the uncertainty of the target's localization and velocity estimates, while keeping the robots apart and assigning costs to the motion of the robots (for instance, to get closer to a ball). The desired formation for the robots possesses the following characteristics:

- When the velocity of the target is zero, the absolute position around the target is undefined and the formation's relative position is determined according to the minimization of the determinant of the merged covariance matrix of the target's position estimation. Nevertheless, when the target's velocity is different from zero, the robot places itself in a position not only that minimizes the determinant of the merged covariance matrix of the target's position estimation but also in a position around the target defined by a constant P_{val} that varies between -1 and 1. For example, if $P_{val} = 1$ the robot goes to the front of the target's velocity vector acting like a receiver and when $P_{val} = 0$ the robot puts itself parallel to the target's velocity vector acting as an observer. If $P_{val} = -1$ the robot acts as a follower behind the target with the velocity vector equal in direction to the target's velocity vector. The minimization of this position term helps stabilize this formation during the target's movement and during the minimization of the total amount of uncertainty on the target's observation;

- As it was not included in the observation model, it must be guaranteed that the robot orientation $\theta_{R_n}(k)$ must be at all times equal to the vector $P_t^{R_n}(k)$ angle, defined by $\theta_t^{R_n}(k)$, in a way that the front of the robot is always turned towards the target;
- As it was not included in the observation model, it must be guaranteed that the robots must maintain a threshold distance D_{val} from the target;
- The robots must not collide with mates, with an obstacle or with the target.

The final cost function is a composition of seven terms described in the following subsections. Nevertheless, it is important to remember that here $|\cdot|$ denotes 1-norm for vector arguments and absolute value for scalars as well as $||\cdot||$ represents the euclidean norm.

4.3.2.1 The Covariance Term

The first term penalizes the total amount of uncertainty given by the merged covariance matrix $\Sigma^{\perp}_{\text{Merged}}(k)$, where λ_a is the function's penalization weight.

$$\sum_{i=N_1}^{N_p} \lambda_a \times |det(\Sigma^{\perp}_{\text{Merged}}(k+i))| \tag{4.20}$$

4.3.2.2 The Distance Term

The second term penalizes the distance between the target and the robot $||P_t^{R_n}(k)||$. To avoid collision, the penalization takes into account a threshold distance that the robot must maintain between it and the target (D_{val}), where λ_0 is the function's penalization weight.

$$\sum_{i=N_1}^{N_p} \lambda_0 \times |(D_{val} - ||P_t^{R_n}(k+i)||)| \tag{4.21}$$

4.3.2.3 The Orientation Term

This third term penalizes the difference between the angle of the robot in the world frame (the orientation of the robot in world frame) and the angle between the robot and the target, which will allow the robot to face the ball. In the case of the 5dpo mobile robot, it faces the target when its concave face (the kicking mechanism) is towards the target. The function $\delta(\cdot)$ receives two angles as arguments and returns their difference scaled between $-\pi$ and π. Finally, λ_1 is the function's penalization weight.

$$\sum_{i=N_1}^{N_p} \lambda_1 \times |\delta(\theta_{\mathrm{R}_n}(k), \theta_{\mathrm{t}}^{\mathrm{R}_n}(k+i))| \tag{4.22}$$

4.3.2.4 The Position Term

This next term will influence the robot's position with respect to the target's velocity vector. It penalizes the robot in the wrong position during the target's movement. Here, the P_{val} will change this position and it must have a value between 1 and -1. Note that this range of values can allow the robot to be in front of the target, behind it, or at its side. Here, λ_2 is the function's penalization weight.

$$\sum_{i=N_1}^{N_p} \lambda_2 \times |P_{val} + (\tilde{P}_{\mathrm{t}}^{\mathrm{R}_n}(k+i) \cdot \tilde{V}_{\mathrm{t}}(k+i))| \tag{4.23}$$

4.3.2.5 The Mates Avoidance Term

The fifth term is a potential function that penalizes the proximity between the robot and its available mates (NM). This is a negative linear function of distance. In the term, the given value where small distances are not penalized is $D_M = 1.5m$. Therefore, the robots must keep a relative distance between them grater than D_M. Here, λ_3 is the term's penalization weight.

$$\sum_{i=N_1}^{N_p} \sum_{j=1}^{NM} \lambda_3 \times \max\left(1 - \frac{||P_{\mathrm{R}_n}^{\mathrm{R}_j}(k+i)||}{D_M}, 0\right) \tag{4.24}$$

4.3.2.6 The Obstacle Avoidance Term

The sixth term works the same way as the previous function. Although it is used here to avoid obstacles. While in the last function the second sum adds for the maximum number of available mates, this term sums all obstacles in the robot's sensor range (NO). Here, λ_4 is the term's penalization weight and $D_O = 1.5m$ has the same purpose of D_M but regarding obstacles.

$$\sum_{i=N_1}^{N_p} \sum_{l=1}^{NO} \lambda_4 \times \max\left(1 - \frac{||P_{\mathrm{R}_n}^{\mathrm{O}_l}(k+i)||}{D_O}, 0\right) \tag{4.25}$$

In the formation, the obstacles are detected as black blobs (everything inside the field which is not green, white or the target's color is considered black blob). Nevertheless, the

teammates are not considered as obstacles once the teammates' pose are shared and any black blob that comes from the mates visual observation is filtered.

4.3.2.7 The Control Effort Term

Finally, the last term penalizes the control effort. In this term, the variation in the output control signal is penalized instead of its absolute value. Penalizing the output control signal would create steady-state error in non-zero velocities (for example when pursuing a moving target).

$$\sum_{i=1}^{N_c} \lambda_5 \times |\Delta U(k+i-1)| \tag{4.26}$$

In this expression, $\Delta U(k+i-1)$ is the variation of the output control signals, where $\Delta U(k+i-1) = \begin{bmatrix} \Delta v(k+i-1) & \Delta vn(k+i-1) & \Delta w(k+i-1) \end{bmatrix}^{\mathrm{T}}$. It is important to notice that the NMPFC does not act directly into the actuators, but in the velocities of the robot. To perform this task, a change should be applied to the model inside the NMPFC used to predict the robot's behavior. The changes in the model could be implemented accordingly to the formation necessity, considering as many models (different mobile robots) as the formation would need.

Therefore, taking into account all the terms previously described, the weights given to each one, and a penalization term for the variation of control effort, the cost function that has to be minimized to control each robot is:

$$
\begin{aligned}
J(N_1,N_p,N_c) = & \sum_{i=N_1}^{N_p} \lambda_a \times |det(\Sigma^{\perp}_{\text{Merged}}(k+i))| + \\
& \sum_{i=N_1}^{N_p} \lambda_0 \times |(D_{val} - ||P_{\text{t}}^{\text{R}_n}(k+i)||)| + \\
& \sum_{i=N_1}^{N_p} \lambda_1 \times |\delta(\theta_{\text{R}_n}(k), \theta_{\text{t}}^{\text{R}_n}(k+i))| + \\
& \sum_{i=N_1}^{N_p} \lambda_2 \times |P_{val} + (\tilde{P}_{\text{t}}^{\text{R}_n}(k+i) \cdot \tilde{V}_t(k+i))| + \\
& \sum_{i=N_1}^{N_p} \sum_{j=1}^{NM} \lambda_3 \times \max\left(1 - \frac{||P_{\text{R}_n}^{\text{R}_j}(k+i)||}{D_M}, 0\right) + \\
& \sum_{i=N_1}^{N_p} \sum_{l=1}^{NO} \lambda_4 \times \max\left(1 - \frac{||P_{\text{R}_n}^{\text{O}_l}(k+i)||}{D_O}, 0\right) + \\
& \sum_{i=1}^{N_c} \lambda_5 \times |\Delta U(k+i-1)|
\end{aligned}
\tag{4.27}
$$

where N_1, N_p are the predicted horizon limits in discrete time, and N_c is the control horizon.

In this cost function, several important parts must be considered to evaluate the proposed contributions of this work:

- There is a penalization function of the total amount of uncertainty of the target's perception, which allows the robots to be in the better position while converging or following a target;

- In the control effort penalization term, the 1-norm is used, which gives the controller more efficiency [88]. However, the disadvantage of 1-norm is its high nonlinearity. Given that it was adopted RPROP [99] and that it is a heuristic optimizer, it can handle the nonlinearities introduced by the use of the 1-norm;

- There is an obstacle avoidance function using a potential field approach and the A* path planner. This function will be better explained in the next chapter;

- The controller's output is the robot's velocities and not the actuator's voltage or velocity, making the controller independent from the robot's dynamics. Therefore, this is an approach that can be applied to different robots' configurations by changing their kinematic model, such as: holonomic, nonholonomic, and so forth.

4.3.3 The Cost Function During Target Absence

The case of the target's absence is an important special case to be considered. To address this issue, another cost function similar to the previous one was created. This cost function is used only in case of the absence of a target in which case each robot has to follow a robot leader that performs a search in a pre-set area using a robust reactive controller [113] with the path planner further explained in chapter 5. Note that this cost function is used only by the follower robots. The case of absence/presence of a target creates some issues that will be discussed in chapter 6, where a solution will be proposed.

The formation geometry, in this case, selects a fixed robot as the leader, while the other two robots become the followers. All the assumptions made for the last cost functions with respect to the target shall be made in this case with respect to the robot leader. An exception is the first term due to the fact that the leader's pose is passed to the other robots (not estimated). Therefore, this cost function is a composition of six terms. The first term penalizes the distance between a leader robot R_L and the follower robot $||P_{\mathrm{R}_n}^{\mathrm{R}_L}(k)||$. To avoid collision, the penalization also does not take into account a fixed distance that the robot must maintain between it and the leader robot (D_{val}).

$$\sum_{i=N_1}^{N_p} \lambda_0 \times |(D_{val} - ||P_{\mathrm{R}_L}^{\mathrm{R}_n}(k+i)||)| \tag{4.28}$$

The second term penalizes the difference between the angle of the robot (a follower) in the world frame and the angle between the follower robot and the leader robot. This will allow the robot to face the leader while it is in movement.

$$\sum_{i=N_1}^{N_p} \lambda_1 \times |\delta(\theta_{\mathrm{R}_n}(k), \theta_{\mathrm{R}_L}^{\mathrm{R}_n}(k+i))| \tag{4.29}$$

This next term will influence the robot's pose with respect to the leader's velocity vector as can be seen in Fig. 4.5. It penalizes the robot in the wrong pose. Here, the P_{val} will change this pose and it must have a value between 1 and -1.

$$\sum_{i=N_1}^{N_p} \lambda_2 \times |P_{val} + (\tilde{P}_{\mathrm{R}_L}^{\mathrm{R}_n}(k+i) \cdot \tilde{V}_{\mathrm{R}_L}(k+i))| \tag{4.30}$$

The fourth, fifth and last terms of this cost function are the same as the fifth, sixth, and seventh terms of the cost function (4.27). Therefore, taking into account all the elements previously described, the weights given to each one of them, and a penalization term to

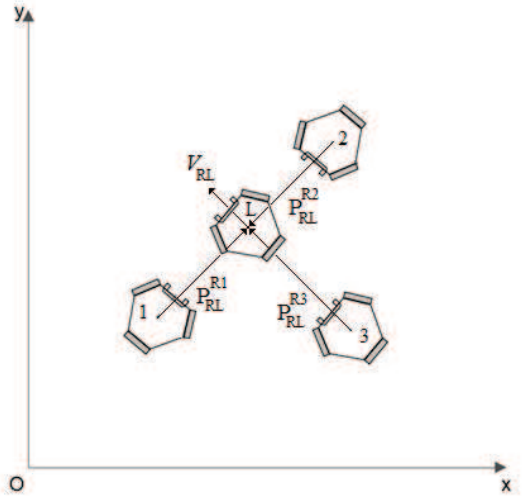

Figure 4.5: Formation Following a Leader

the variation of control effort, the cost function that represents all this, embedded in all robots is as follows:

$$
\begin{aligned}
J(N_1,N_p,N_c) = & \sum_{i=N_1}^{N_p} \lambda_0 \times |(D_{val} - ||P_{\mathrm{R}_L}^{\mathrm{R}_n}(k+i)||)| + \\
& \sum_{i=N_1}^{N_p} \lambda_1 \times |\delta(\theta_{\mathrm{R}_n}(k), \theta_{\mathrm{R}_L}^{\mathrm{R}_n}(k+i))| + \\
& \sum_{i=N_1}^{N_p} \lambda_2 \times |P_{val} + (\tilde{P}_{\mathrm{R}_L}^{\mathrm{R}_n}(k+i) \cdot \tilde{V}_{\mathrm{R}_L}(k+i))| + \\
& \sum_{i=N_1}^{N_p} \sum_{j=1}^{NM} \lambda_3 \times \max\left(1 - \frac{||P_{\mathrm{R}_n}^{\mathrm{R}_j}(k+i)||}{D_M}, 0\right) + \\
& \sum_{i=N_1}^{N_p} \sum_{l=1}^{NO} \lambda_4 \times \max\left(1 - \frac{||P_{\mathrm{R}_n}^{\mathrm{O}_l}(k+i)||}{D_O}, 0\right) + \\
& \sum_{i=1}^{N_c} \lambda_5 \times |\Delta U(k+i-1)|
\end{aligned}
\tag{4.31}
$$

4.4 The Optimization Algorithm

RPROP, short for resilient propagation, is a learning scheme, that performs a direct adaptation of the weight step based on local gradient information. After the cascade correlation algorithm and the Levenberg−Marquardt algorithm, RPROP is one of the fastest weight update mechanisms. The main reason for the success of this algorithm roots in the concept

of direct adaptation of the size of the weight-update. In contrast to all other algorithms, only the sign of the partial derivative is used to perform both learning and adaptation. This leads to a transparent and yet powerful adaptation process, that can be straight forward and very efficiently computed with respect to both time and storage consumption [99].

To overcome the inherent disadvantages of pure gradient-descent, RPROP performs a local adaptation of the weight updates according to the behavior of the error function. In substantial difference to other adaptive techniques, the effort of the RPROP adaptation process is not blurred by the unforeseeable influence of the size of the derivative but only dependent on the temporal behavior of its sign. These characteristics motivated the selection of RPROP to deal with the NMPFC optimization problems [20].

Whereas standard backpropagation (BP) uses fixed proportions (the learning rate) of the error gradient to adjust weights, RPROP introduces a time varying weight step Δ_i that determines the size of the weight-update. This adaptive update value evolves during the learning process based on its local sight on the error function $f(\cdot)$.

Every time the partial derivative of the corresponding weight $w_i(t)$ changes its sign, which indicates that the last update was too big and the algorithm jumped over a local minimum, the update value Δ_i is decreased by the factor η^-. If the derivative retains its sign, the update value increases slightly in order to accelerate convergence in shallow regions.

Once the update value for each weight is adapted, the weight update itself Δw_i follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update value, if the derivative is negative, the update value is increased. However, if the partial derivative changes sign (i.e. the previous step was too large) and the minimum was missed, the previous weight update Δw_i is reverted.

Due to that backtracking weight step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update value, there should be no adaptation of it in the following step. Therefore, the value of $\frac{\partial f}{\partial w_i}(t-1)$ is set to zero in the Δ_i adaptation-rule.

Moreover, with η^+ and η^- as the optimizer parameters, the RPROP algorithm as implemented in this thesis is defined as follows:

Resilient Propagation Algorithm

for i:=0 to N_i **do**
  if $\frac{\partial f}{\partial w_i}(t-1).\frac{\partial f}{\partial w_i}(t) > 0$ **then**
    $\Delta_i(t) = \min(\Delta_i(t-1) * \eta^+, \Delta_{max})$
    $\Delta w_i(t) = -sign(\frac{\partial f}{\partial w_i}(t)) * \Delta_i(t)$
    $w_i(t+1) = w_i(t) + \Delta w_i(t)$
  else if $\frac{\partial f}{\partial w_i}(t-1).\frac{\partial f}{\partial w_i}(t) < 0$ **then**
    $\Delta_i(t) = \max(\Delta_i(t-1) * \eta^-, \Delta_{min})$
    $w_i(t+1) = w_i(t) - \Delta w_i(t-1)$
    $\frac{\partial f}{\partial w_i}(t) = 0$
  else if $\frac{\partial f}{\partial w_i}(t-1).\frac{\partial f}{\partial w_i}(t) = 0$ **then**
    $\Delta w_i(t) = -sign(\frac{\partial f}{\partial w_i}(t)) * \Delta_i(t)$
    $w_i(t+1) = w_i(t) + \Delta w_i(t)$

In this thesis, the same values as [20] were used ($\eta^+ = 1.2$, $\eta^- = 0.8$ and $N_i = 20$, where N_i is the maximum number of iterations), except for Δ_0 and S_tp, where S_tp is the stop criteria for the optimizer. Here, $\Delta_0 = 0.15$ and $S_tp = 0.1$ are used. The stop criteria is the maximum acceptable value for the cost function minimization during the optimization process. Nevertheless, if the optimization process does not converge in less then 20 iterations, the algorithm assumes a sub-optimal solution and also stops the optimization process. In almost all situations from the tests made by [20], the RPROP algorithm in the robots converges in less than 10 iterations and the efficiency of the RPROP algorithm could be proven in his work. Moreover, this algorithm is much more powerful than previously thought, handling even nonlinearities such as the use of the 1-norm in the cost function of the NMPFC controller.

4.5 Weight Tuning Methodology

This section introduces a metric to verify the optimal values for the cost function's weights. The optimal values are the ones that minimize an object function that reflects the controller's efficiency with respect to a given criteria. It is possible to empirically find the initial values for these weights by analyzing the importance and penalization priority for each weight. Nevertheless, to be correct and accurate, it is necessary to provide a criterion to an objective function and use, for example, a numeric gradient descent method in order to search for the optimal value with respect to these weights (lambdas). Therefore, to achieve a final weight tuning this section presents:

1. An empirical methodology which allows the setting of the initial cost function's weights;

2. A gradient search for the values of λ_i, where $\lambda_i = \left[\lambda_a, \lambda_1, ..., \lambda_5 \right]^{\mathrm{T}}$, which evaluates the controller by minimizing an objective function V_{opt} with respect to the adopted criteria further explained.

4.5.1 Initial Analysis of the Terms and Weights of the Cost Function

The main objective of this chapter is to conceive a NMPFC that converges a formation towards a target, minimizing the total amount of uncertainty of the team of robots' perception of the target, while avoiding collision with teammates and obstacles, orienting the robots towards the target, and penalizing oscillations in the control effort. While analyzing this main objective, it is necessary to study the cost function beforehand (4.27), as well as the multi-robot system and the desired formation performed by the team of robots. In this cost function, in order to minimize the total amount of uncertainty in the target perception, the robots in formation first have to approach the target. This approach has to be executed by considering both the best position around the target and the avoidance of teammates or obstacles during the robots' movements. Therefore, it is clear that in a scale of minimization priority, the distance term of the cost function should possess the highest weight followed by the covariance term, the position term, the obstacle avoidance term and the teammate's avoidance term. These last four terms are equally important to the performance of the formation, implying an equal weight to all four terms. The orientation of the robot towards the target comes after the previously cited items in the scale of priority because it does not influence the robots' movement while the formation is converging, but only after the position has been achieved the behavior is affected. Finally, there must be a small weight in the control effort term in order to prevent the robots from shaking in the convergence area, which would result in an unstable formation.

When analyzing the cost function (4.27), it is necessary to decide whether the terms in the cost function should also be active during the entire time. By analyzing the cost function (4.27), it is clear that the position (λ_1) and the orientation (λ_2) are the only terms that do not need to be active at all times during the robots' movements, due to the fact that they only influence the robot's movement when it is near the target (less then $4m$ in the robots' setting).

In order to activate these terms, which penalize the error in position and orientation of each robot, an activation function based on their initial given values was created.

$$f(||P_{\mathrm{t}}^{\mathrm{R}_n}(k)||) = -\left(\frac{\lambda_{imax}}{(Dist_{max} - Dist_{min})}\right) \times ||P_{\mathrm{t}}^{\mathrm{R}_n}(k)|| + \left(\frac{Dist_{max}\lambda_{imax}}{(Dist_{max} - Dist_{min})}\right) \quad (4.32)$$

where λ_{imax} is the initial signed value of λ_1 or λ_2 (depending on the applied term), $Dist_{max} = 4m$ is the maximum distance in which a robot can see a target and $Dist_{min} =$

$1.5m$ is the distance threshold between the robot and the target that activates the weight.

This function is activated every time the robot places itself less than $4m$ away from the target. If its distance is greater than $4m$ the weight is automatically zero. If it is less than $1.5m$, the value is equal to λ_{imax}. Therefore, the value of the weight λ_i will be as follows:

$$\lambda_i = \begin{cases} f(||P_{\mathrm{t}}^{\mathrm{R}_n}(k)||) & \text{if} \quad 1.5 < ||P_{\mathrm{t}}^{\mathrm{R}_n}(k)|| \leq 4 \\ \lambda_{imax} & \text{if} \quad ||P_{\mathrm{t}}^{\mathrm{R}_n}(k)|| \leq 1.5 \\ 0 & \text{if} \quad ||P_{\mathrm{t}}^{\mathrm{R}_n}(k)|| > 4 \end{cases} \tag{4.33}$$

4.5.2 The Tuning

Finally, an objective function was created to measure the NMPFC's efficiency. This function is given by the equation (4.34) and it serves as a metric to evaluate the performance of the robot's controller in formation with the given cost function's weights. Therefore, the objective function is as follows:

$$V_{opt} = \beta_{Traj} \times \varepsilon_{Traj} + \beta_{Time} \times \varepsilon_{Time} + \beta_{OR} \times \varepsilon_{OR} + \beta_J \times J \tag{4.34}$$

with

$$\begin{aligned} \varepsilon_{Traj} &= \sum_{i=1}^{Na} ||\Delta_s||, \quad \varepsilon_{Time} = \tau_{2\%} \\ \varepsilon_{OR} &= \frac{\sum_{i=1}^{Na} |(||P_{\mathrm{t}}^{\mathrm{R}_n}(i)|| - D_{val}^{OR})|}{Na} \end{aligned} \tag{4.35}$$

and

$$\beta_{Traj} = 0.1 \quad \beta_{Time} = 0.01 \quad \beta_J = 0.00001 \quad D_{val}^{OR} = 1.2$$

$$\beta_{OR} = \begin{cases} 0 & \text{if} \quad ||P_{\mathrm{t}}^{\mathrm{R}_n}(i)|| > 1.5 \\ 0.7 & \text{if} \quad ||P_{\mathrm{t}}^{\mathrm{R}_n}(i)|| \leq 1.5 \end{cases}$$

where J is the final value of the cost function in equation (4.27), D_{val}^{OR} is the distance desired value and $\tau_{2\%}$ is determined based on the graph of distance between the robot and the target. Therefore, $\tau_{2\%}$ is the time of convergence when the value of distance is in $+/-2\%$ of its final value. ε_{Time} is the function that penalizes the time to converge, ε_{Traj} is the function that penalizes a bigger trajectory and ε_{OR} is the function that penalizes the overshoot region made by the robots whenever they pass over the threshold distances,

where Na is the total number of steps. Finally, β represents the weights for each term of the objective function (4.34).

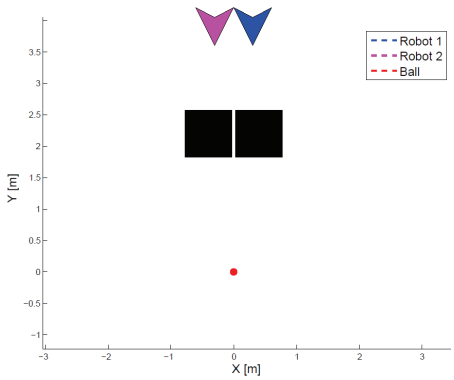

Figure 4.6: Tuning Test Setup

Furthermore, a gradient search with a step of $+/-2\%$ of the initial value was performed with respect to the metric described in equation (4.34) in order to search for a set of weights that could give the minimum value for this objective function. This gradient search uses the values of a simulation environment used as testbed where its configuration is set with robots R_1 and R_2 departing from coordinates (0.3,3.9) and (-0.3,3.9), respectively, as illustrated in Fig. 4.6. Both robots were 90° (degrees) oriented in the world frame. In the path, there were also two fixed obstacles with coordinates (-0.4,2.2) and (0.4,2.2). The target (ball) was placed in the coordinates (0,0) of the field. This testbed pushes the formation to its limits, forcing all terms to be activated at some instant in time during the performed trajectory.

4.5.3 The Influence of the Number of Terms

The process of tuning the parameters of the NMPFC cost function is highly dependent on the criteria adopted and the number of terms in the cost function. This subsection exemplifies this dependency throughout the tuning of the weights. It also demonstrates the change in the final weights values at each time a term is added in the cost function.

Furthermore, let's begin assuming that the cost function in the equation 4.27 possesses initially only the terms related to distance, teammates avoidance, and obstacle avoidance. Let's remember that the weight of the term that penalizes the oscillations in the control effort (λ_5) is always static and off the tuning process, due to the fact that at least one term must be always active (different from zero).

Therefore, the NMPFC cost function would be such as:

$$
\begin{aligned}
J(N_1,N_p,N_c) = & \sum_{i=N_1}^{N_p} \lambda_0 \times |(D_{val} - ||P_{\mathrm{t}}^{\mathrm{R}_n}(k+i)||)|+ \\
& \sum_{i=N_1}^{N_p}\sum_{j=1}^{NM} \lambda_3 \times \max\left(1 - \frac{||P_{\mathrm{R}_n}^{\mathrm{R}_j}(k+i)||}{D_M}, 0\right)+ \\
& \sum_{i=N_1}^{N_p}\sum_{l=1}^{NO} \lambda_4 \times \max\left(1 - \frac{||P_{\mathrm{R}_n}^{\mathrm{O}_l}(k+i)||}{D_O}, 0\right)+ \\
& \sum_{i=1}^{N_c} \lambda_5 \times |\Delta U(k+i-1)|
\end{aligned}
\tag{4.36}
$$

Applying the previous proposed method to tune these parameters (λ_0, λ_3 and λ_4), the weights obtained through this process can be seen in the table 4.1.

Table 4.1: Weights for the Cost Function with Only Four Terms

λ	**Weights**
λ_0	1050
λ_3	615
λ_4	750
λ_5	5.00

Now let's input the merged covariance penalization term with its weight λ_a. Applying once more the proposed method to tune these parameters (λ_a, λ_0, λ_3 and λ_4), the weights obtained through this process can be seen in the table 4.2.

Table 4.2: Weights for the Cost Function with Five Terms

λ	**Weights**
λ_a	475
λ_0	909
λ_3	545
λ_4	605
λ_5	5.00

Notice here that with the addition of the covariance term, the weights of each term of the cost function changed. This implies that each term influences the other during the formation. It also implies the coupling of the cost function terms.

Finally, the final values were found through this gradient search performing exhaustive simulations. These final simulations consider all the terms in the cost function (4.27). These values can be seen in table 4.3.

Table 4.3: Final Weights for the Cost Function: Holonomic Robots

λ	**Weights**
λ_a	505
λ_0	918
λ_1	297
λ_3	500
λ_4	500
λ_5	5.00

The values found by exhaustive simulations suffice for the experiments presented in the results section. The results from this simulation (Fig. 4.6) performed with the values in table 4.3 can be seen in Fig. 4.7.

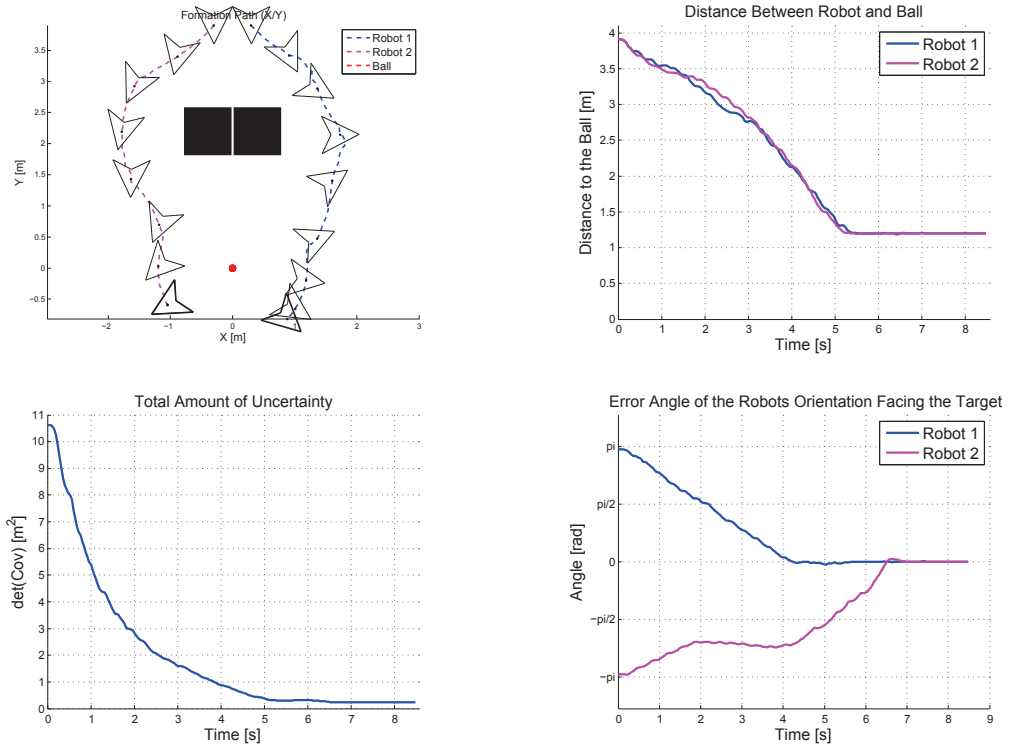

Figure 4.7: Testbed results: using the initial weights

They represent the behavior of the formation for the values found using the gradient search. These four graphs show an XY plot of the robots' movements, the distance from the robot to the ball, the minimization of the merged covariance determinant and the error angle between the robots' orientation and the distance between the robots' and the target.

Furthermore, the final weight parameters had also to be found for the nonholonomic robots. The parameters were found through the iterative methodology proposed and they can be seen in table 4.4. As the nonholonomic robot has more nonlinearities and movement constraints, the control effort had to be increased to better control the nonholonomic

robot's movement.

Table 4.4: Final Weights for the Cost Function: Nonholonomic Robots

λ	**Weights**
λ_a	126.25
λ_0	229.50
λ_1	73.50
λ_3	125
λ_4	125
λ_5	5.00

4.6 Results

The final weights used in the formation were found throughout the gradient search which in turn uses the above testbed to update these weights. Then, a set of results comprising several simulations and experiments with real robots were obtained using these weights to validate the NMPFC controller. In the performed experiments it was possible to see the behavior of the group under communication, vision and localization problems. Each group has convergence and retaining formation experiments. The same parameters were used in the process of obtaining all results. In the experiments, the following assumptions have been made:

- Desired distance between robot and ball = 1.2m;
- Desired distance between robot and leader robot = 1.2m;
- Maximum velocity of the robots in formation or following leader = 1.5m/s in simulation and 0.7m/s in real experiments;
- Maximum velocity of the leader = 0.5m/s;
- The RPROP parameters were the same ones used in [133] with a maximum of 20 interactions;
- Nonholonomic robot diameter = 0.95m;
- The simulations and experiments with the moving ball (keeping formation) include the activation of the Position term. The tuning criteria did not considered this term. The weight for this term was adjusted manually where $\lambda_2 = 510$ for holonomic robots and $\lambda_2 = 127.50$ for nonholonomic robots.
- The control horizon was $N_c = 2$ and the prediction limits were $N_1 = 1$ and $N_p = 7$.

4.6.1 Simulations

A setup was created to simulate the behavior of a formation with three mobile robots in the SimTwo simulator. In order to run the simulator, a notebook was used (Intel Core i5 3Ghz/Core with 8Gb RAM) with Windows 7. Also in this computer, a software to capture the simulator screen was running. Another computer, a workstation (Intel Core i7 3Ghz/Core with 8Gb RAM) with Ubuntu 9.04, was used to run the Coach and the Decs (one for each robot). This configuration can be seen in Fig. 4.8. Both computers are connected by cable to a router. The communication between them is done using UDP.

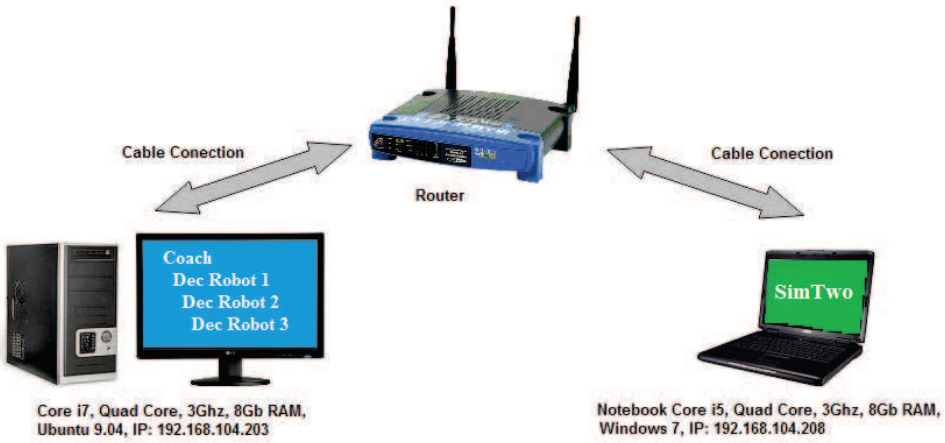

Figure 4.8: Simulation Setup

In simulations, SimTwo sends each respective robot's pose and the position of the target to their Dec. The Dec of a robot R_n also communicates with the other robot's Decs through the Coach sending its own state and the ball state as it observes. Furthermore, the Coach sends back to each Dec the state of all robots in formation. Finally, the SimTwo receives back the control inputs for each robot. In this case, the Coach was used to share the information through cable connection among robots instead of RTDB, which aided in the simplification of the system.

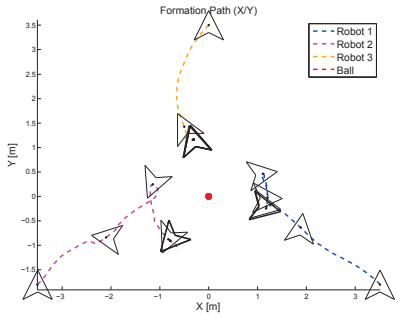

Figure 4.9: Simulation 1: Homogeneous Formation Convergence - Plot XY

4.6.1.1 Simulation 1: Homogeneous Formation Convergence

The first simulation with a homogeneous group of robots puts them initially like demonstrated in Fig. 4.9. The objective here is to converge the robots towards the ball departing from different places in the field and test purely the formation convergence. The coordinates for robots 1, 2, 3 and the target are respectively (3.5,-1.8), (-3.5,-1.8), (0,3.5) and (0,0).

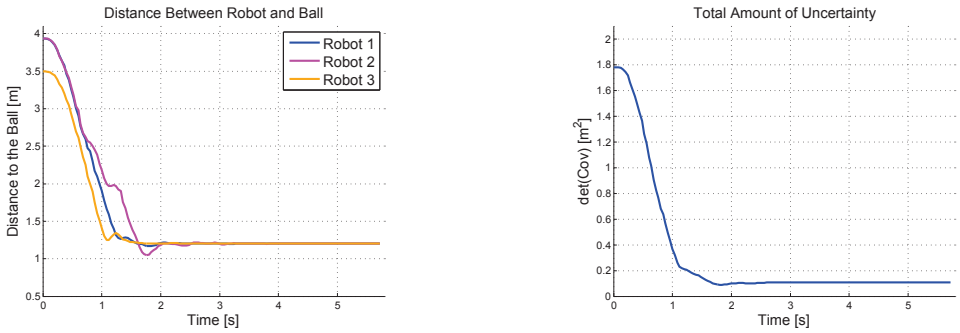

Figure 4.10: Simulation 1: Homogeneous Formation Convergence - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$

In the formation convergence, the target is stopped. Therefore, the fourth term (position) in the cost function is equal to zero. Furthermore, without any obstacles the sixth term (obstacle avoidance) is also zero. The video **C1T1Conv** shows the robots' movement as presented in the plot XY. Note here that the robot always stops facing the ball (with the carved part towards the ball). A graph with the distance between the robot and the ball as well as the minimization of the merged covariance's determinant can be seen in Fig. 4.10 which shows the formation converged to the desired distance from the target, minimizing the total amount of uncertainty and penalizing the deviation in the orientation.

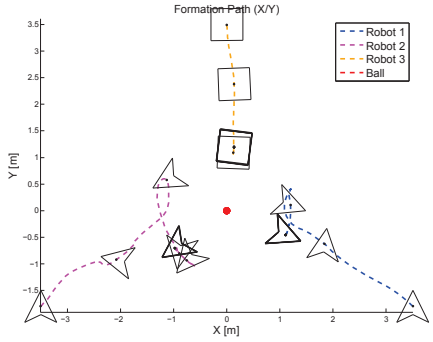

Figure 4.11: Simulation 1: Heterogeneous Formation Convergence - Plot XY

4.6.1.2 Simulation 1: Heterogeneous Formation Convergence

The previous simulation is performed once more but this time with a heterogeneous group of robots. It puts the robots far from each other, one in each part of the field, like demonstrated in Fig. 4.11. Using the same setup as before, this simulation aims to observe the formation convergence when the third robot is the nonholonomic robot making it an heterogeneous formation. In Fig. 4.11, the XY plot of the robot's movement can be seen.

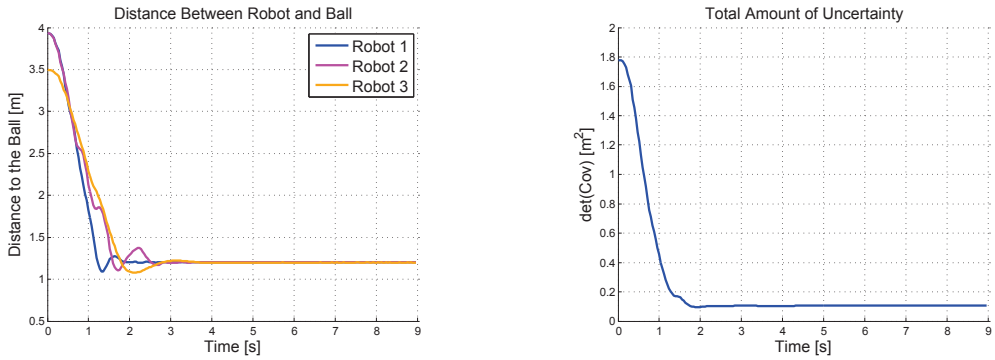

Figure 4.12: Simulation 1: Heterogeneous Formation Convergence - Distance between Robot and Ball and the Determinant of $\Sigma^{\perp}_{\text{Merged}}$

Note here that the robot always stops facing the ball. A graph with the distance between the robot and the ball as well as the minimization of the merged covariance's determinant can be seen in Fig. 4.12. It can also be noticed by analyzing the video **C3T1Conv** that despite the lower velocity of the nonholonomic robot, the formation converges successfully to the desired distance from the target also minimizing the total amount of uncertainty and setting the robots oriented towards the ball.

This first pair of simulations shows a simple environment where the collisions with mates are most improbable. The next two pairs of simulations will test the case where the robots depart from close positions between them.

4.6.1.3 Simulation 2: Homogeneous Formation Convergence

The second pair of simulations places the robots initially like demonstrated in Fig. 4.13. The objective here is to push the robots to converge towards the ball departing from the coordinates (-5.5,0), (-4.5,0), (-3.5,0) and (0,0) for robots 1, 2, 3 and target, respectively. This environment was created to observe whether the robots can avoid collisions between them during the formation convergence.

In Fig. 4.13 the plot XY describing the robot's movements can be seen. The video **C1T2Conv** shows the robots' movements as presented in the plot XY. Here, it is shown that no collisions between the robots occurred while the convergence is performed. A

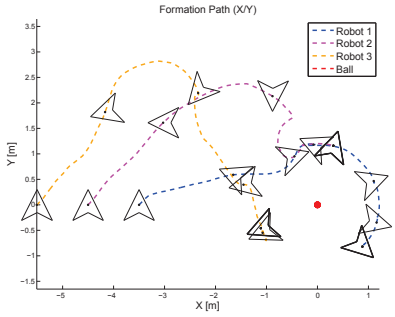

Figure 4.13: Simulation 2: Homogeneous Formation Convergence - Plot XY

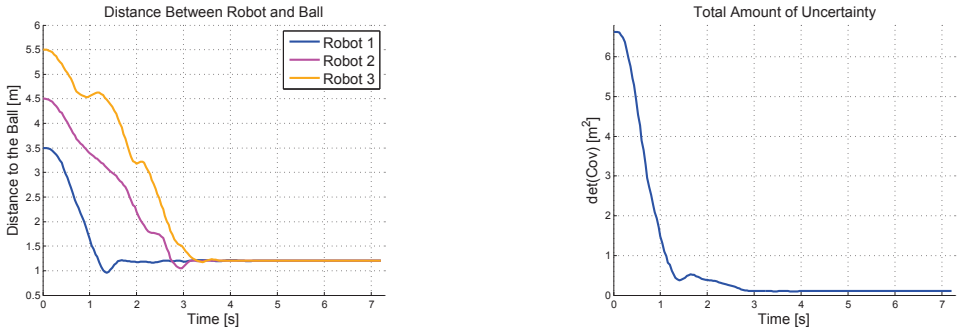

Figure 4.14: Simulation 2: Homogeneous Formation Convergence - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$

graph with the distance between the robot as well as the minimization of the merged covariance's determinant can be seen in Fig. 4.14.

4.6.1.4 Simulation 2: Heterogeneous Formation Convergence

Like in the previous simulation, the robots here are placed like demonstrated in Fig. 4.15. Once more the objective here is to do a formation convergence without the collision between the robots with an initial configuration forming an horizontal line. The difference here is that robot 3 is now a nonholonomic robot and it was set in the closest position to the target to increase the difficulty in this simulation.

In Fig. 4.15 it can be seen the plot XY of the robot's movement. Note in the video **C3T2Conv** that despite the constraints in the movement of the nonholonomic robot, no collisions happened and the formation converged. In this simulation, the movement of robot 3 is more smooth while the movements of the holonomic robot are more tortuous due to the necessity to avoid collision with the nonholonomic robot. A graph with the distance between the robot and the ball as well as the minimization of the merged covariance's determinant can be seen in Fig. 4.16 demonstrating the formation convergence towards the target.

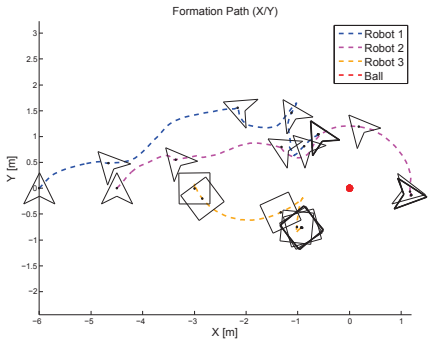

Figure 4.15: Simulation 2: Heterogeneous Formation Convergence - Plot XY

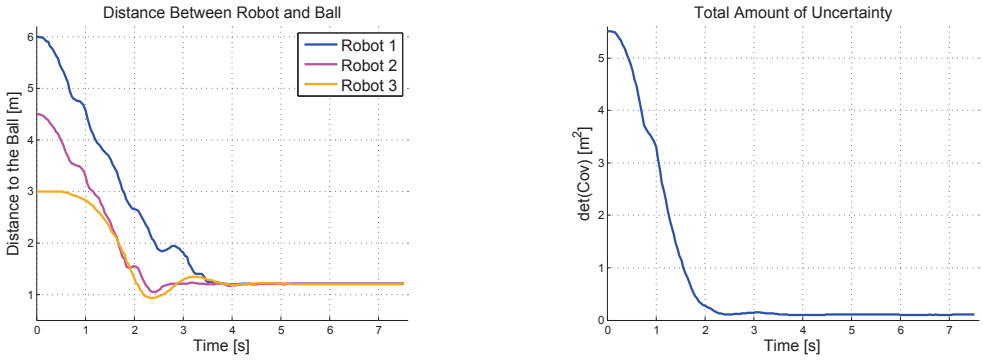

Figure 4.16: Simulation 2: Heterogeneous Formation Convergence - Distance between Robot and Ball and the Determinant of $\Sigma^{\perp}_{\text{Merged}}$

4.6.1.5 Simulation 3: Homogeneous Leader Following

This simulation places the robots initially as demonstrated in Fig. 4.17 and has a main objective of analyzing the leader-following environment. Therefore, two robots will follow a robot leader departing from the coordinates (6.3,-3.1), (6.3,0), (6.3,3.1) for robots 1, 2 and 3, respectively. The follower robots shall converge to the leader robot and keep a formation while following it. In this case, the target is not seen in the field, so robot 1 (as the leader robot) has an A* path planner seen in [134] with a reactive controller [113] while the follower robots (robots 2 and 3) possess the second cost function presented in this study and the position term is now taken into account.

In Fig. 4.17, the plot XY of the robot's movement and a graph with the distance between the followers and the robot leader is presented. The video **C1T3Conv** also shows the robots' movement as presented in the plot XY. No collisions between robots occurs in this simulation.

This case used the same values obtained during the weight tuning methodology demonstrating the robustness of the controller. However, if analyzed carefully, the objective function (4.34) used to perform the gradient search did not contemplate this case. The

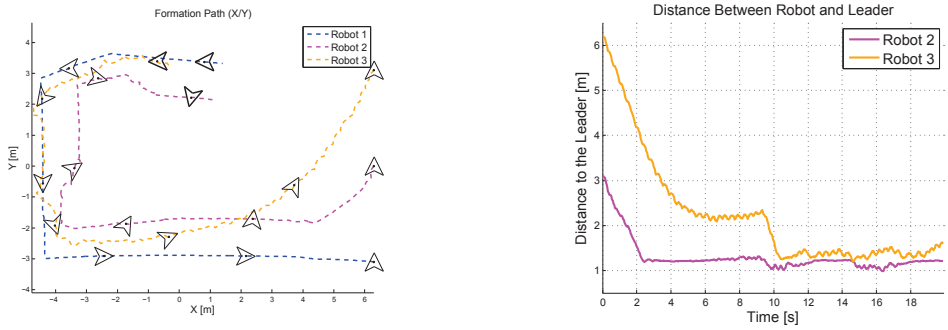

Figure 4.17: Simulation 3: Homogeneous Leader Following - Plot XY and Distance between the Leader Robot and the Followers

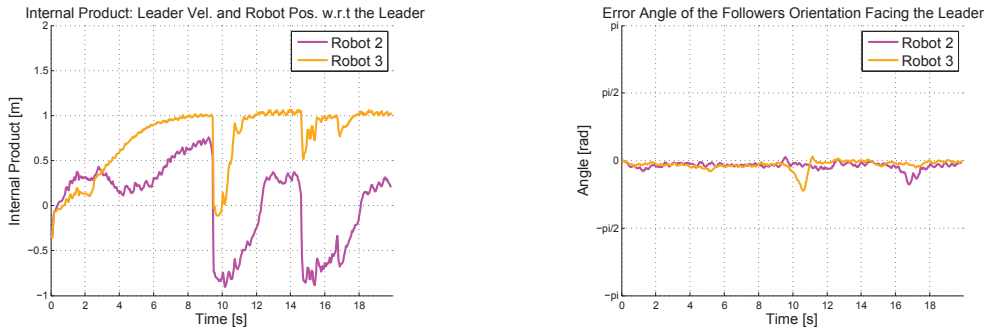

Figure 4.18: Simulation 3: Homogeneous Leader Following - Internal Product Between Leader and Followers and Error Angle of the Followers Orientation Facing the Leader

criteria was composed by the trajectory made, the overshoot in the convergence, the time of convergence and the cost function itself. The internal product for example was not contemplated and that can explain the behavior of robot 2 in Fig. 4.18 where it shows the graphs of the internal product between the leader and the other robots. The given P_{val} for robot 2 was $P_{val} = 0$ and for robot 3 was $P_{val} = -1$. This graph shows two falls in the internal product convergence due to the leader's change of orientation. When the leader robot turns, the followers became oriented in 90° with respect to the leader's orientation until they converge again, also explaining the "jumps" of orientation in Fig. 4.18. Furthermore, regarding the internal product, robot 3 converged successfully while robot 2 did not. Nevertheless, the error angle between the robot leader and the followers as successfully minimized as demonstrated in the graph of Fig. 4.18.

In this work, the choice to create another criteria just for this situation to re-tune the parameters was not taken. That could cause a disturbance in the tuning of other parameters due to the fact that the terms of the cost function are coupled (one penalization term influences the other indirectly). Nevertheless, the tuning could be done with another criteria that could contemplate a better formation behavior regarding the internal product convergence.

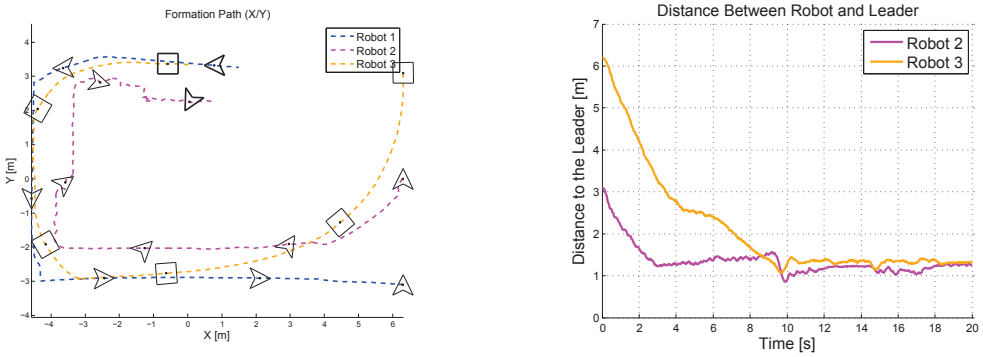

Figure 4.19: Simulation 3: Heterogeneous Leader Following - Plot XY and the Distance between the Leader Robot

4.6.1.6 Simulation 3: Heterogeneous Leader Following

This simulation has the same objective as the previous one where the convergence towards a leader robot and the formation keeping by two follower robots during its movement is performed. The difference here is once more the substitution of robot 3 by a nonholonomic robot.

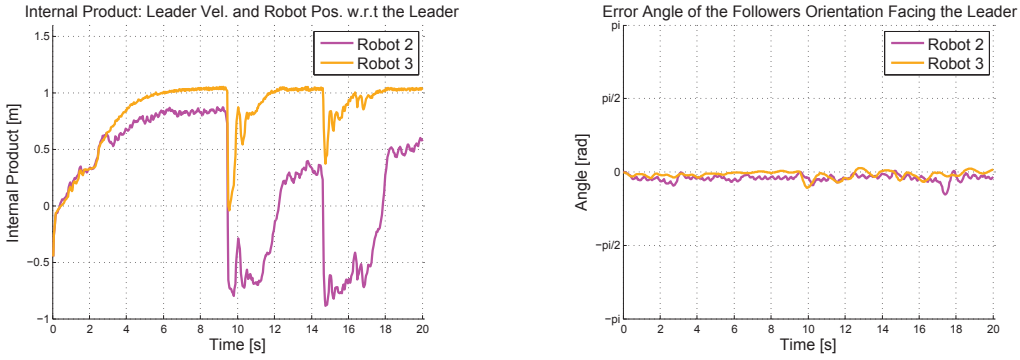

Figure 4.20: Simulation 3: Heterogeneous Leader Following - Internal Product Between Leader and Followers and Angle Between the Leader and the Followers

The XY plot of the robot's movement, as well as a graph with the distance between robot and its leader, can be seen in Fig 4.19. Note that due to the constraints in the mobility of robot 3, it possesses a smoother trajectory with a better behavior of robot 3 than in the previous simulation. The convergence of robot 3 and the behavior of robot 2 regarding the internal product with the leader robot can be seen in Fig. 4.20. The angle between the followers and the leader can also be seen in this figure, where the convergence of this term is also shown. Despite the constraints in the nonholonomic robot's movement, it can be noted in the video **C3T3Conv** that the simulation proves the success of the formation in a leader following approach and the controller's robustness. However, like the simulation

before, a tuning could be also applied with a different criterion in order to improve, for example, the internal product.

4.6.1.7 Simulation 4: Homogeneous Keeping Formation

The main objective in this simulation is to observe the conservation of the formation during the target's movement in a straight trajectory. Robots 1, 2, 3 and the target's initial coordinates are (0,-1.5), (0,1.5), (1.5,0) and (0,0), respectively, as demonstrated in Fig. 4.21. Once more, the position term is taken into account, but this time in the first cost function. The only term that is not active is the obstacle avoidance term. The ball was guided in a straight line forcing it to possess a constant non zero velocity. In Fig. 4.21 the plot XY of the robot's movement can be seen. The video **C1T1Man** also shows the robots' movement as presented in the plot XY. No collisions between the robots occurs and the formation is kept successfully during the simulation. A graph with the distance between the robot and the ball and the minimization of the merged covariance's determinant can be seen in Fig. 4.22.

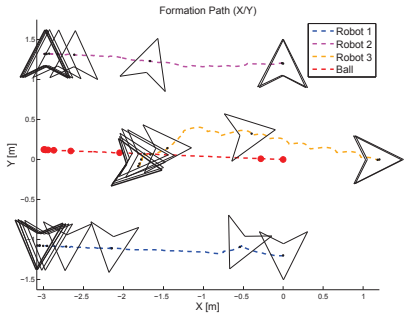

Figure 4.21: Simulation 4: Homogeneous Keeping Formation - Plot XY

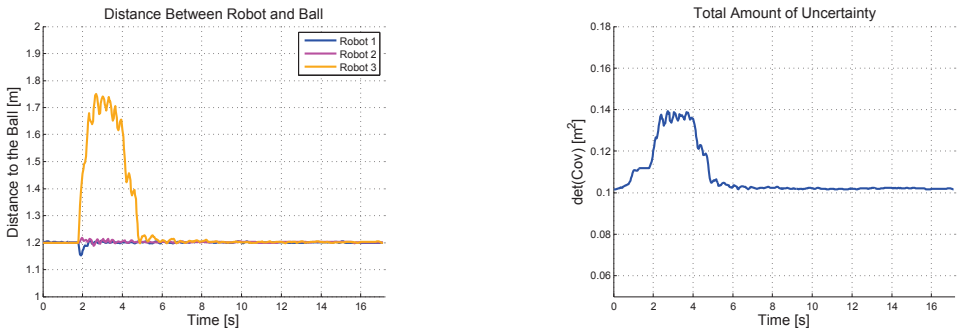

Figure 4.22: Simulation 4: Homogeneous Keeping Formation - Distance between Robot and Ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$

By analyzing the graph distances, it can be witnessed that robot 3 (the one behind the ball) has more difficulty following the target. Its lag in relation to the balls movement is

also summed to its acceleration. Therefore, the initial distance between robot 3 and the ball is grater then the other robots in formation.

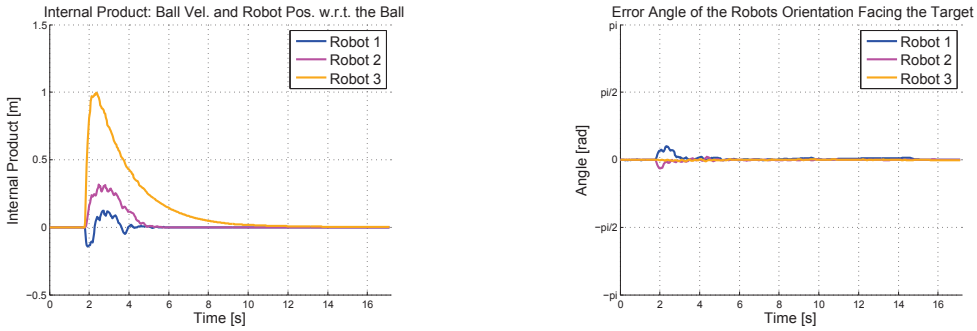

Figure 4.23: Simulation 4: Homogeneous Keeping Formation - Internal Product Between the Robots and the Ball and Error Angle of the Robots Orientation Facing the Target

Fig. 4.23 shows the graphs of the internal product between the robots and the ball as well as the angle between the robots' and the ball's velocity vector. The given P_{val} for robots 1, 2 and 3 were $P_{val} = 0$, $P_{val} = 0$ and $P_{val} = -1$, respectively. Note that during the ball's movement ($V_t(k) \neq 0$) the internal product is different from zero in robot 3. As soon as the velocity tends to zero, the internal product also drops due to the absence of the ball's velocity vector.

4.6.1.8 Simulation 4: Heterogeneous Keeping Formation

This simulation aims to observe the heterogeneous behavior of the formation during the active target tracking problem. As occurred in the previous simulation, the robots are placed initially like demonstrated in Fig. 4.24. The video **C3T1Man** also shows the robots' movement as presented in the plot XY.

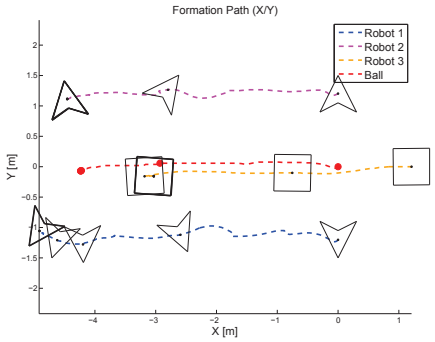

Figure 4.24: Simulation 4: Heterogeneous Keeping Formation - Plot XY

A graph of the distance between the robot and the ball as well as the minimization of the merged covariance's determinant can be seen in Fig. 4.25. Note here that there is

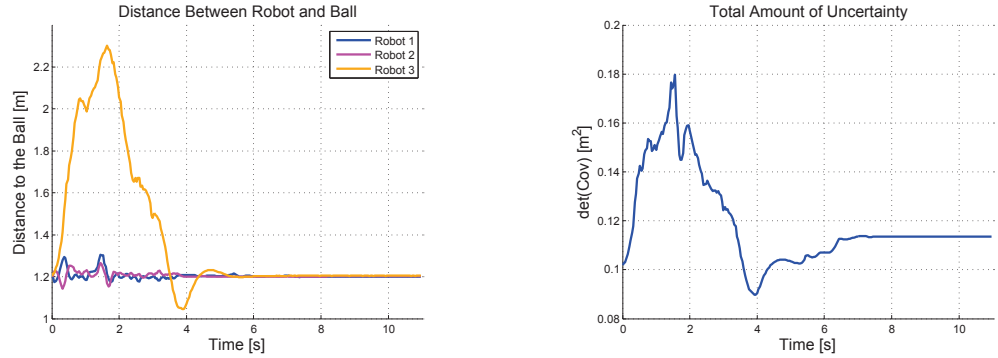

Figure 4.25: Simulation 4: Heterogeneous Keeping Formation - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$

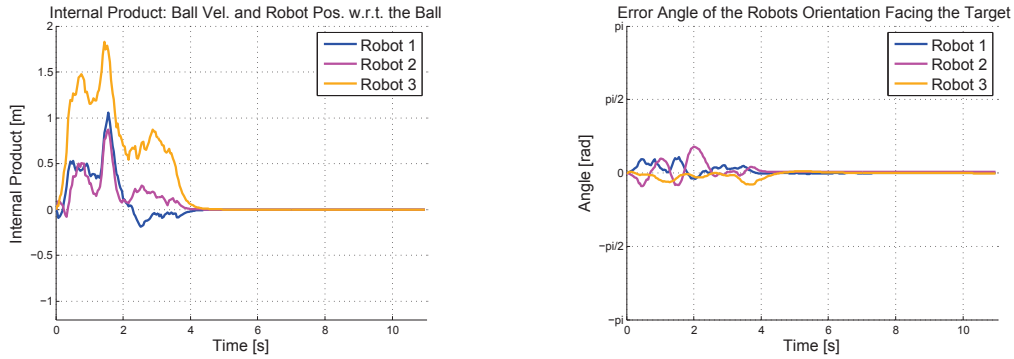

Figure 4.26: Simulation 4: Heterogeneous Keeping Formation - Internal Product Between the Robots and the Ball and Angle Between the Robots and the Ball's Velocity Vector

initially a greater difference in the distance between the target and robot 3 (the follower) as soon as the target starts moving. This distance is greater here than in the simulation of the homogeneous group due to the nonholonomic robot's constraints and the lower maximum velocity this robot has. Nevertheless, it reaches the target and then it readjusts its pose (robot 3) putting itself at the desired distance and orientation with respect to the target. Fig. 4.26 shows the graphs of the internal product between the robots and the ball as well as the angle between the robots and the ball's velocity vector reflecting the same behaviors as the previous simulation. Finally, it can be seen that the formation is kept successfully while the target moves.

4.6.2 Results of the Experiments with Real Robots

A setup was created to perform experiments with real robots in order to analyze the behavior of three omnidirectional mobile robots with the NMPFC which is presented in Fig. 3.2. An external computer running the Coach application and connected to a router by cable was needed in order to serve as a bridge of shared information between the robots,

where the log of the formation could be collected. Therefore, to run the experiments a workstation (Intel Core i7 3Ghz/Core with 8Gb RAM) with Ubuntu 9.04 that runs said bridge was used. Each robot had a computer, a Notebook (Intel Dual Core 2Ghz/Core with 2Gb RAM) with Ubuntu 9.04, running its own NMPFC and applications previously seen in Fig. 4.1. Finally, the experiments with real robots were executed in order to repeat the environment created in the simulations with two experiments in formation convergence, one experiment in leader following and one experiment in keeping formation case. This configuration can be seen in Fig. 4.27.

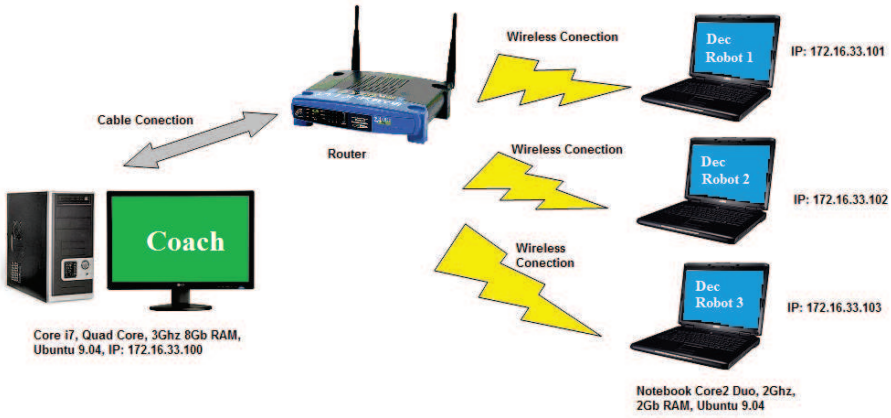

Figure 4.27: Experiment Setup

In the real experiment, the SimTwo is substituted by the vision software (HAL) sending locally, to Dec, information to calculate the pose of its robot and the position of the target. Then, each Dec sends the control inputs to the robot. This main processing unit (Dec) handles external communication with other robots through wireless communication. Finally the Coach collects the information from the robots with their pose and their observed ball state (position and speed of the ball). The Coach then uses its state machine to activate the roles that each robot must execute, sending back also the information about which ball from the observed ball is the "best" ball, or the true ball, that will be used by the formation. In these results the CTE was not implemented and was substituted by the Coach software. The main reason is that the CTE is another thesis work from the FCT project (PTDC/EEA-CRO/100692/2008 - Perception-Driven Coordinated Multi-Robot Motion Control), which in turn is not the purpose of this thesis to evaluate.

Therefore, as the Coach chooses the "best" measurement between the robots measurement, and therefore this becomes the "best ball", it is possible to see a "strange" behavior of the ball in the experiments when it "jumps" from a position to another in an instant of time. This is due to the change the coach performs when choosing the "best" ball between the robots measurement.

Finally, it is important to notice that in all experiments with real robots there is noise in the localization and perception of the ball. The localization noise has a standard deviation in X and Y of $0.05m$, and in θ of $0.045m$ and the noise in perception of the ball has a standard deviation in X and Y of $0.316m$. When comparing the results from the simulations with those with the real robots through the analysis of the videos of both cases it can be seen that this noise influences the robots mainly in the final convergence. This influence puts the robots to shake a little when converging, never completely stopping (they stop because the experiment was over). This shaking can be noticed in that little oscillation around the converging final values presented in the graphics of orientation, covariance penalization and so on. Nevertheless, it also shows the controller's robustness due to the fact that despite the noise, the formation converged to the desired behavior.

4.6.2.1 Real Experiment 1: Formation Convergence

The real experiments were conducted with robots number 1, 2 and 3. The first experiment placed the robots initially like demonstrated in Fig. 4.28. In this figure, the XY plot0 of the robot's movement can also be seen. As it was in the first simulation, there are no obstacles in this experiment and the objective is to see the convergence towards a target placed in the center of the field. Robots 1, 2 and 3 are placed far from each other and their positions are (-3,1.7), (0,-2) and (3,1.7), respectively. All robots were 270° oriented in the world frame. The target's coordinates are (0,0). In Fig. 4.29 the graph of the distance between the robot and the ball as well as the graph of the total amount of uncertainty (merged covariances) minimization can be seen. The video **C1T1ConvReal** shows the robots' movement as presented in the plot XY.

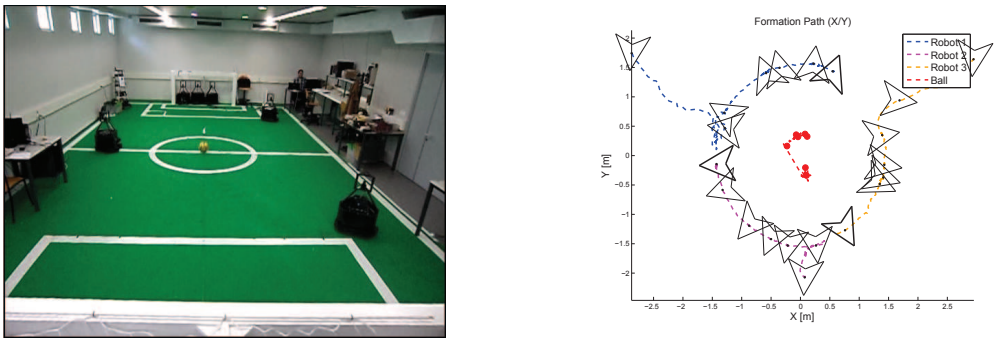

Figure 4.28: Real Experiment 1: Formation Convergence

Note here that no collisions between the robots occur and that the robot always stops while facing the ball (with the carved part towards the ball) as shown in the simulations. As it can be seen, the ball "jumps" from one coordinate to another because the formation chooses the "best" ball amongst the balls' measurement which come from all three robots. Nevertheless, the robots move in a perfect circle around the ball while trying to minimize

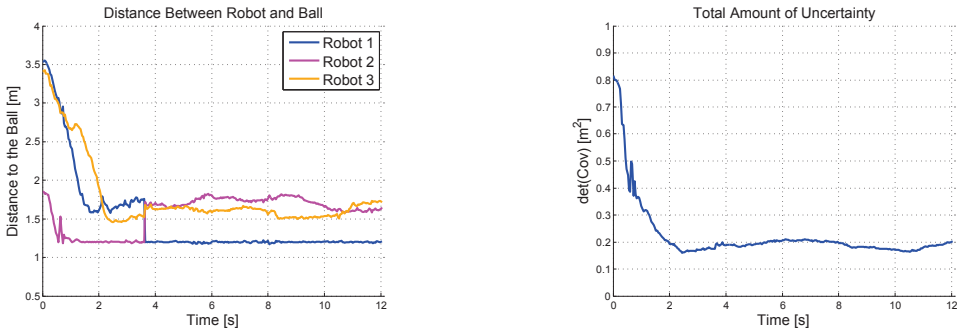

Figure 4.29: Real Experiment 1: Formation Convergence - Distance between the robot and the ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$

the covariance. That is explained by the fact that even if the best ball changes, or "jumps" from one coordinate to another, the total cost of these terms (covariance and distance) are kept very similar as it can be seen in Fig. 4.29. However, the formation converges successfully.

4.6.2.2 Real Experiment 2: Formation Convergence

The second experiment places the robots initially at the coordinates (2,0), (3,0), (4,0) and (0,0) for robots 1, 2, 3 and target respectively, like demonstrated in Fig. 4.30. The objective here is to observe the robots converging to the ball avoiding collisions between them. Similarly to the second pair of simulations, the robots are placed in a horizontal line to increase the difficulty in the convergence forcing the robots to avoid each other. The video **C1T2ConvReal** shows the robots successfully converging towards the target while avoiding collision between them.

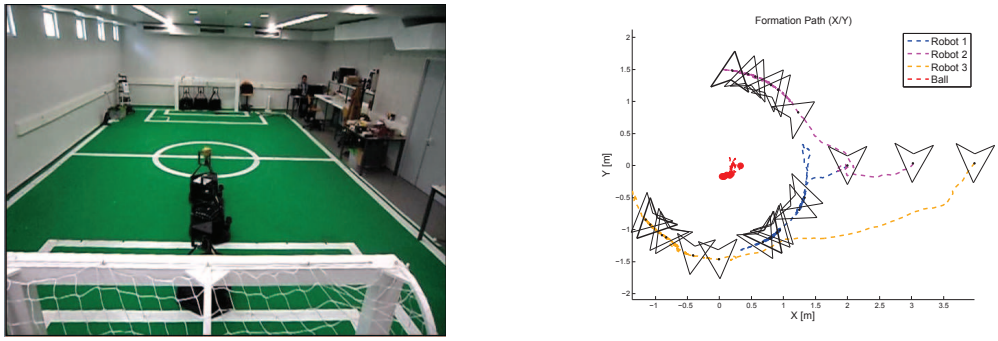

Figure 4.30: Real Experiment 2: Formation Convergence

In Fig. 4.31, a graph with the distance between the robot and the ball as well as the graph of the total amount of uncertainty (merged covariances) minimization can also be seen. Note that this time, there is no "jump" on the "best" ball.

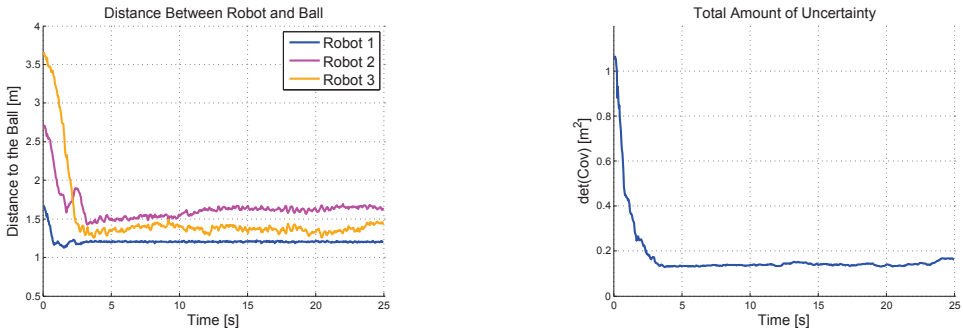

Figure 4.31: Real Experiment 2: Formation Convergence - Distance between the robot and the ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$

4.6.2.3 Real Experiment 3: Leader Following

This experiment, like the leader following simulation, placed the robots initially as demonstrated in Fig. 4.32. The objective here is to converge the two follower robots towards the leader robot departing from the coordinates (3,-2), (3,0), (3,2) for robots 1, 2 and 3, respectively. The follower robots must also keep a formation while following the leader.

Figure 4.32: Real Experiment 3: Leader Following

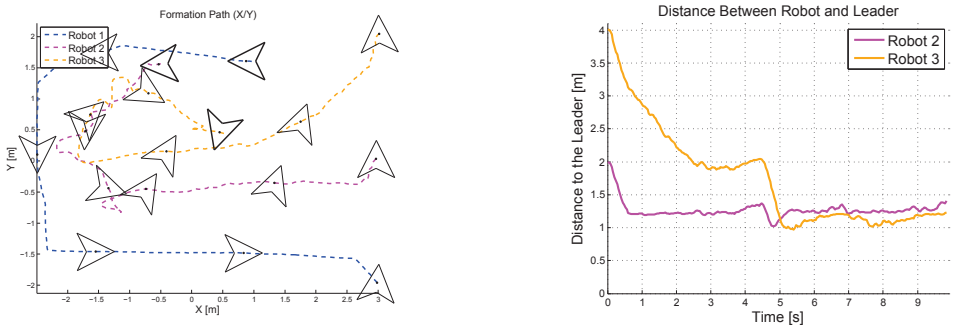

Figure 4.33: Real Experiment 3: Leader Following - Plot XY and Distance between the Leader Robot and the Followers

Therefore, robot 1 (as the leader robot) has an A* path planner with a reactive controller while the follower robots (robots 2 and 3) have the second cost function presented in this chapter. In Fig. 4.33 the XY plot of the robot's movement as well as a graph with the distance between the follower robots and the robot leader can be seen. The video **C1T3ConvReal** shows the robots' movement as presented in the plot XY.

Fig. 4.34 shows the graphs of the internal product between the leader and the other robots as well as the angle between them. Finally, the given value for P_{val} for robots 2 and 3 were $P_{val} = -1$ and $P_{val} = 0$, respectively. Similarly to the simulations and despite the correct orientation towards the leader robot, the followers have trouble in converging the internal product due to the reasons previously explained. The abrupt changes in the robots' internal product are also explained by the same reasons demonstrated in the simulation of this experiment.

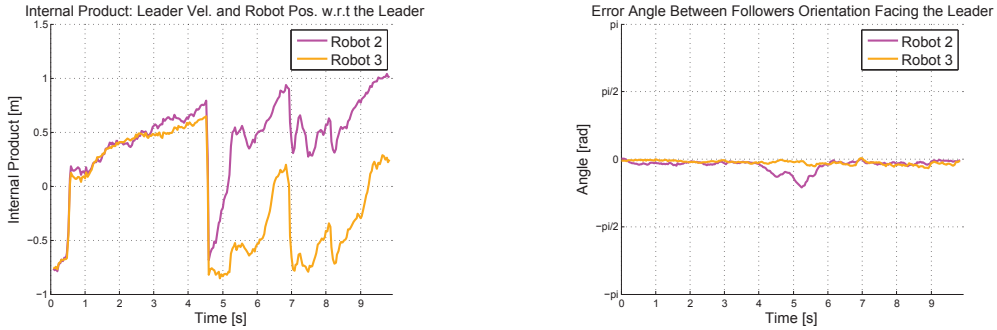

Figure 4.34: Real Experiment 3: Leader Following - Internal Product Between Leader and Followers and Error Angle of the Followers Orientation Facing the Leader

4.6.2.4 Real Experiment 4: Keeping Formation

A last experiment was executed to perform formation maintenance with real robots. As the ball has a non zero velocity, the fourth term in the cost function (equation 4.23) is taken into account. The real experiment consisted of pushing the ball in a straight line with the help of a stick to emulate a small kick. The robots 1, 2, 3 and the target's initial coordinates are (0,1.5), (0,-1.5), (-1.5,0) and (0,0), respectively, like demonstrated in the plot XY of the robot's movement in Fig. 4.35. The video **C1T1ManReal** shows the robots' movement as presented in the plot XY.

In the formation, each robot has, with a small amount of errors in observation, its own measurement of the target in the world's frame. During the formation, this information is exchanged and a "best ball" is chosen from all the measurements that come from the robots. As a "best ball" is chosen between the measured balls, the position of this final ball can create, along with small errors in the localization, differences in the distances between the robots and the ball as it can be seen in the XY Plot and in Fig. 4.36 that

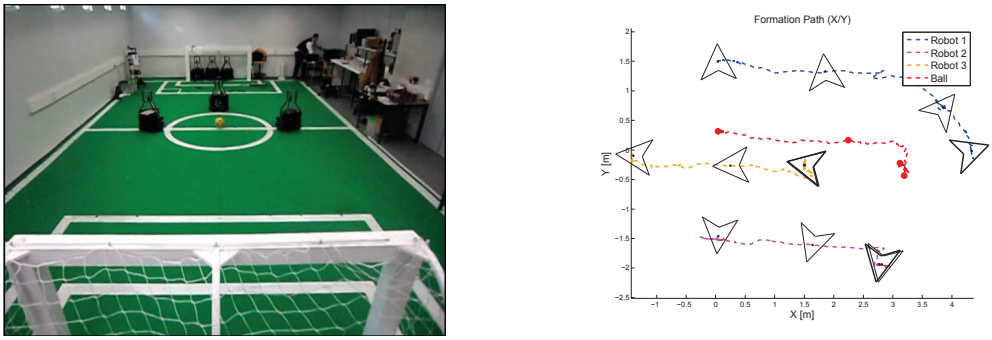

Figure 4.35: Real Experiment 4: Keeping Formation

shows the distance between the robots and the ball and the graph of the total amount of uncertainty (merged covariances) minimization.

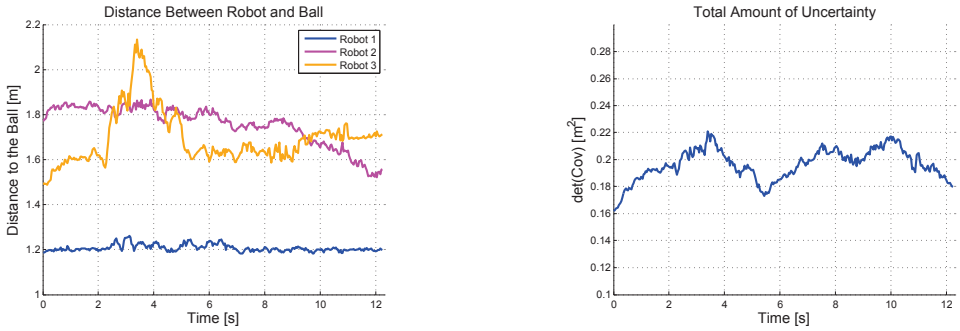

Figure 4.36: Real Experiment 4: Keeping Formation - Plot XY and Determinant of $\Sigma^{\perp}_{\text{Merged}}(k)$

Finally, in Fig. 4.37 the graphs of the internal product between the robots and the ball as well as the angle between the robots and the ball's velocity vector can be seen. The same values of P_{val} for robots 1, 2 and 3 used in the corresponding simulation, were used in this experiment.

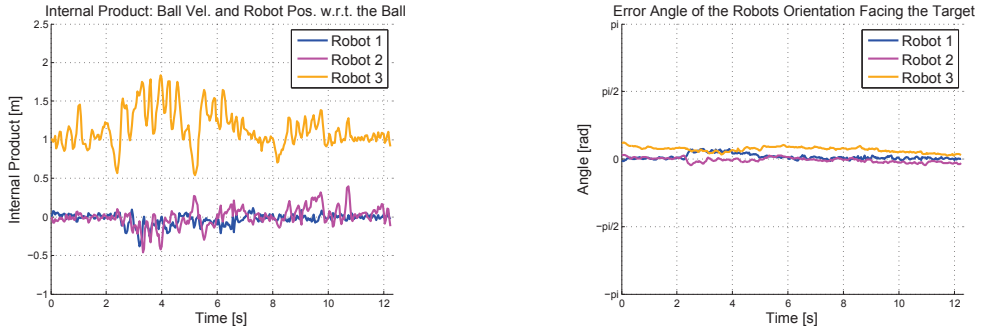

Figure 4.37: Real Experiment 4: Keeping Formation - Internal Product Between the Robots and the Ball and Error Angle of the Robots Orientation Facing the Target

4.7 Conclusion

This chapter had the main objective to present a new nonlinear model predictive controller used in formation control of multi-robot systems, here called a nonlinear model predictive formation controller (NMPFC). The NMPFC was implemented in order to converge a group of mobile robots towards a desired target and also to maintain the formation during the target's movement. Furthermore, a methodology to empirically find the initial values for the weights of the NMPFC cost function was presented. Finally, the case of absence of a target was considered too, choosing a fixed leader robot in this case and using a second cost function to follow the leader robot. The details of the controller structure as well as a mathematical analysis of the formation model were presented.

This chapter presented several novel contributions introduced by this thesis in formation control using a nonlinear model predictive controller, such as the distributive architecture to control robots in formation; a generic cost function for penalization of non desirable behaviors of the formation when following an observed and non controlled target; the minimization of a merged covariances of the target perception in the cost function; the evaluation of which terms of the cost function are active during the formation convergence and maintenance as well as a gradient such to find the optimal values of the cost function parameters with respect to a desired criteria; the ability to control homogeneous and heterogeneous groups of mobile robots; and the controller's robustness despite vision and localization problems present in real robots environments.

The NMPFC showed to be very flexible, easily adaptable and able to be used in holonomic and nonholonomic robots. This controller could be applied for both types of robots due to the fact that the control effort does not take into account the voltages on the motors or the actuator input signals, but instead the robots velocities. Furthermore, the projected controller was capable of making a team of different robots converge to a desired position around the target, even if the robots are very far apart.

The results also showed the influence of the vision system on the controllers efficiency during the formation keeping experiments. Small errors in the ball detection and localization are reflected in the calculation of the distance between the robot and the target and in the covariance minimization. Nevertheless, the controller was successful in converging the robots to the desired pose, minimizing the total amount of uncertainty while avoiding mates and setting the correct pose for all robots. Both convergence and keeping the formation were such that the total amount of uncertainty in the target's perception was minimized. Regarding the experiments without a target, the follower robots were successful in converging towards the leader and keeping the formation while the leader robot was moving despite the weights not being tuned with a criteria that contemplated the internal product term.

Moreover, the methodology proposed to search for a set of weights for the NMPFC's

cost function proved to be efficient. Nevertheless, a tuning aiming the situation regarding the following leader was not performed showing both the controller's robustness and a non optimal convergence of the internal product term (position term) of the cost function.

The NMPFC controller demonstrated a good performance in formation control of multi-robot systems. It could be implemented in different types of robots and penalize several non desirable behavior of the formation with the same generic structure. Non-linearities could be included in the prediction model such as the total amount of uncertainty in the target's perception which in turn gave relevance to the requirements imposed by target localization and/or tracking. Furthermore, energy costs, avoidance of obstacles, uncertainties and formation characteristics were considered in a single optimization problem. Nevertheless, among the disadvantages, the NMPFC still possesses reasonable computational costs in the system's processing. Moreover, there is no guarantee of global minimization of the cost function during the optimization process.

Finally, it can be concluded that the NMPFC was a good choice regarding the active target tracking problem by a formation of mobile robots due to its performance previously seen in the results. In this specific problem, the formation is equally important to the target tracking. The NMPFC's ability to consider both approaches by penalizing non-desirable behaviors of the formation proves the efficiency of this controller when applied to the active target tracking problem.

Chapter 5

Obstacle Avoidance in Formation Control

This chapter presents an explanation of the obstacle avoidance problem considered by the nonlinear model predictive formation control. First, an approach is presented which uses potential functions as terms of the NMPFC. These terms penalize the proximity with mates and obstacles. A strategy to avoid singularity problems with the potential functions using a modified A* path planning algorithm was then introduced. Results with simulations and experiments with real robots are presented and discussed.

5.1 Introduction

The obstacle avoidance problem forms a well known area of research in mobile robotics. It is a study that involves a wide range of subjects which encompasses from a single robot movement to a group of mobile robots moving in a specific formation. Issues like static or mobile obstacle avoidance in single or multiple robots' motion are the main study cases in path planning. This chapter presents a deeper explanation about the obstacle avoidance problem considered by the nonlinear model predictive formation control of multi-robot system.

Motion planning algorithms are widely used nowadays. In areas such as: UAV path planning [135], mobile robot outdoor navigation, mobile robot indoor navigation[136] and even in video games [137] path planning algorithms have been conceived to be the solution to many obstacle avoidance issues.

Therefore, many path planning techniques rose over the years. One among the most famous is the artificial potential field approach. This methodology has been widely used and it states that the collision-free trajectory is generated along the negative gradient of the

defined attractive and repulsive potential field functions. The subsequent studies can be found in [138], [131], and [139]. Nonetheless, the potential field method is not straightforwardly applicable to mobile vehicles with kinematic constraints since, in the potential-field design, the robot is usually treated as a simple particle. Another major problem is the fact that since the method is an essentially faster descent optimization method, the robot can get trapped into local minima of the potential function other than the goal configuration [140].

Among the most famous there is also the Roadmap method. A computational geometry data structure was proposed in [141] to solve the problem of an optimal path generation between a source and a destination in the presence of simple disjoint polygonal obstacles. In [142] a good application of the Roadmap method is applied where the use of multiple mobile robots in a common environment, such as underground mining and warehouse management problems, are considered despite the fact that no randomly moving obstacles are used. The Roadmap method is applied in low-dimension configuration spaces and it is not easy to be implemented, depending of the approach [140].

Finally, the last method among the most classic algorithms for path planning is the Cell Decomposition [140]. In this category are famous and efficient algorithms such as A*, D*, ARA* and AD*. The A* algorithm is the oldest. It is well applied with static [143] and dynamic obstacles [136]. It can be applied also for UAV obstacle avoidance [135] in unknown environments. In [144] an approximate cell decomposition method was developed in which obstacles, targets, sensor's platform, and FOV (Field of View) are represented as closed and bounded subsets of an Euclidean workspace. A good overview about the advantages and disadvantages of these algorithms cam be seen in [145] and [146].

One of the methods that has evolved in recent years is the *Velocity Obstacles*, first used in [147]. This method defines the set of all the velocities of a robot that will result in a collision at some point in time, assuming that the obstacle maintains the current speed. Therefore, its movement planning aims at finding the speeds that fall outside these groups to ensure that there will be no collisions. This method is widely used in simulations of crowds, however it has two problems: first, when dealing with static obstacles, the robot circumvents the edges of the obstacle making the robot slower. The second problem is the processing time, as noted in [148].

In this chapter, a deeper explanation is presented about the obstacle avoidance problem considered by the nonlinear model predictive formation control. In the next section the problem is formulated describing the issue studied. The obstacle avoidance solution with potential field approach and the NMPFC is presented in the following section. Then, the A* path planning algorithm is explained in section 5.4. The results of the experiments with real robots and simulations are shown in section 5.5 and finally the conclusion is shown in section 5.6.

5.2 Problem Formulation

Let A be a single rigid object (the robot) moving in a euclidean space W, called workspace, where $W \subset \Re^2$. The simulation sets a mobile robot (5dpo middle size league soccer robot or the nonholonomic Mini-AGV) as the rigid object A. Let "q" be a position of robot A in the workspace W. Therefore, in this workspace the robot is placed in an initial position q_{init} and in the far side of the workspace W, there is the target point q_{target}. In between the initial and target points there is a bigger circle centered in the middle of the field that symbolizes any crowded environment such as a shopping mall's hall or a factory corridor, where people have to perform random trajectories.

Let $B_1,\ldots,B_n$ be also, in a first case, fixed rigid objects distributed in W. Let the space C be the configuration space of robot A, or in other words, all the configuration reachable by A in the workspace W. Therefore, all obstacles B_i in the workspace W are mapped in C into a region $CB_i = \{q \in C/A(q) \cap B_i \neq 0\}$ which is called **C-obstacle**. The union of all C-obstacles is called **C-obstacle region**, or the region occupied by the obstacles. Moreover, the **free space** in the workspace W is given by:

$$C_{free} = C \setminus \bigcup_{i=1}^{n} CB_i = \{q \in C/A(q) \cap (\bigcup_{i=1}^{n} CB_i) = 0\} \tag{5.1}$$

Therefore, the problem becomes: given an initial position of A and a target position in W, converge to the target point q_{target} avoiding contact with the CB_i's during all the trajectory.

When the C-obstacles are no longer static, in another words they vary their position with time, it can no longer be represented as in the problem before. This issue is solved by simply adding a dimension to C, obtaining therefore, $CT = Cx[0,+\infty)$, which is called **configuration space-time** of A. Thus, every obstacle B_i maps in CT to a stationary region CTB_i, called a **CT-Obstacle**, defined by:

$$CTB_i = \{(q,t)/A(q) \cap B_i(t) \neq 0\} \tag{5.2}$$

5.3 Potential Field Approach in NMPFC

The NMPFC is inserted in the optimal control group. It means that this kind of controller uses an optimizer algorithm to find the best solution, or optimal control output. When considering such controllers in real time optimization many approaches are available. Nevertheless, the obstacle avoidance approach that most rapidly and easily fits into this kind of control is the artificial potential fields (APF) approach. In [88], the author uses the

APF embedded in a nonlinear model predictive controller as an example for avoid static obstacles.

The potential field approach uses a potential function to navigate the robot (attraction function) that drives the robot towards the target, and an avoidance function (repulse function) that repels the robot when it is near an obstacle. If the NMPFC is considered, then the attraction function could be seen as the first and second terms of the NMPFC cost function in equation 4.27. Therefore, a repulsive function had to be made in order to consider the obstacle avoidance problem.

The main idea underlying the definition of the repulsive potential is to create a potential barrier around the C-obstacle region that cannot be traversed by the robots' configuration [140]. In addition, it is usually desirable that the repulsive potential does not affect the motion of the robots when it is sufficiently far away from the C-obstacles. One way to achieve these constraints is to define the repulsive potential function as follows:

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta(\frac{1}{\rho(q)} - \frac{1}{\rho_0})^2 & \text{if} \quad \rho(q) \leq \rho_0 \\ 0 & \text{if} \quad \rho(q) > \rho_0 \end{cases} \tag{5.3}$$

where η is a positive scaling factor, $\rho(q)$ denotes the distance from q to the C-obstacle region CB, i.e.:

$$\rho(q) = \min_{q' \in CB} ||q - q'|| \tag{5.4}$$

and ρ_0 is a positive constant called the distance of influence of the C-obstacles. The function U_{rep} is positive or null, it tends to infinity as q gets closer to the C-obstacle region, and is null when the distance of the robots' configuration to the C-obstacle region is greater then ρ_0.

This thesis divided the problem of obstacle avoidance in two repulse functions. The first considers the mate avoidance, preventing the robots from colliding with themselves. The second function considers the obstacle avoidance, preventing the robots from colliding with static or moving obstacles which may, or may not appear.

5.3.1 Mate Avoidance Function

The first idea of a term in a nonlinear model predictive controller that penalizes the approximation between robots in a formation was presented in [20]. In his work, the author created a sub-function in his nonlinear model predictive controller such as in the equation 5.5.

$$\sum_{i=N_1}^{N_p} \lambda_3 \times \left(\left(\frac{1}{||P_{\mathrm{R}_n}^{\mathrm{R}_{m1}}(k+i)|| - D_M}\right)^2 + \left(\frac{1}{||P_{\mathrm{R}_n}^{\mathrm{R}_{m2}}(k+i)|| - D_M}\right)^2\right) \tag{5.5}$$

Where $||P_{\mathrm{R}_n}^{\mathrm{R}_{m1}}(k+i)||$ is the distance between robot R_n and the mate 1. This function has a nonlinear decreasing behavior as shown in Fig. 5.1

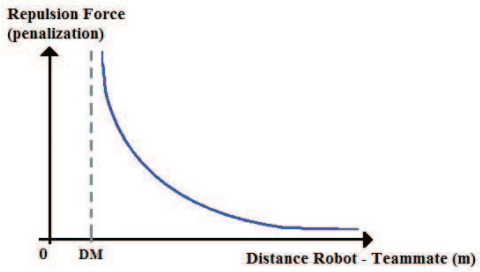

Figure 5.1: Behavior of function in 5.5

As it can be noticed, the first problem is that this approach does not consider a generalized number of mates, only two. However, a more important issue is addressed when analyzing the behavior of this function. The avoidance function for its nonlinearity, takes more time to increase the penalization by proximity, allowing the robots to get too near each other before penalizing it.

A simple solution is proposed here to avoid these problems. The proposed function for mate avoidance can be seen in equation 5.6.

$$\sum_{i=N_1}^{N_p} \sum_{j=1}^{NM} \lambda_3 \times \max\left(1 - \frac{||P_{\mathrm{R}_n}^{\mathrm{R}_j}(k+i)||}{D_M}, 0\right) \tag{5.6}$$

Remembering that NM is the maximum number of mates, $||P_{\mathrm{R}_n}^{\mathrm{R}_j}(k+i)||$ is the distance between robot R_n and the mate R_j and DM is the given value where small distances are not penalized. The proposed function was changed to a linear function which increases the penalization with proximity much more rapidly. The generalization of mates was also considered with a second sum that gives scalability to the NMPFC controller in this thesis.

Finally, an extreme case had to be considered when using potential functions. This extreme case, also studied among the potential field approach, takes into account the possibility of the robots getting too close to each other much more rapidly then allowed. This behavior can occur if the robots are moving in high velocities for instance. To avoid collision in these cases a protection zone is created around the robots where the weight of the attraction and repulsion functions (terms of the NMPFC cost function) are rapidly switched so the robot gives priority to penalize the mate avoidance rather then get to the

target. When the robots are outside this zone once again, the weights of the cost function are set back to normal.

5.3.2 Obstacle Avoidance Function

An obstacle avoidance function was created based on the idea of mates avoidance function and the potential field approach. The repulsion function proposed in this thesis can be seen in equation 5.7.

$$\sum_{i=N_1}^{N_p}\sum_{l=1}^{NO}\lambda_4 \times \max\left(1-\frac{||P_{\mathrm{R}_n}^{\mathrm{O}_l}(k+i)||}{D_O},0\right) \tag{5.7}$$

Remembering also that NO is the maximum number of obstacles, $||P_{\mathrm{R}_n}^{\mathrm{O}_l}(k+i)||$ is the distance between robot R_n and the obstacle O_l and DO is similar to DM.

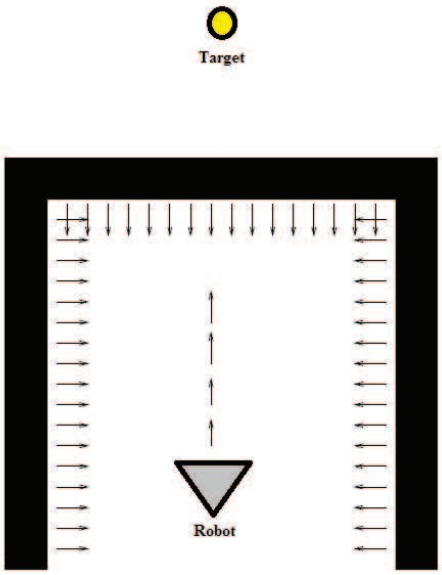

Figure 5.2: Local Minima Problem Case

This function's behavior is similar to the function proposed in the mate avoidance problem. In the obstacle avoidance proposed function, all obstacles (static or moving) are considered to be stopped during the 40ms loop control. This assumption speeds up the calculations in the prediction of the NMPFC by calculating only the robot-obstacle distance evolution in a simplified fashion. Nevertheless, the static obstacles have a major issue in potential fields approach: the local minima problem. The most common problem of APF lies in a robot being stuck in a local minima. It is the case that occurs when the

total force acting on the agent is summed up to zero even though the robot has not reached its goal position yet. Fig. 5.2 shows the local minima problem case.

In order to solve this problem, a switching approach was implemented inside the task *Formation*, which in turn is inside the role *Formation*. In this approach, the NMPFC with the potential function term is substituted by a controller with a modified A* path planning algorithm if the robots gets trapped in a local minima (if the path towards the target is obstructed or if there are obstacles in one of the robot's side). In most of this local minima problems, the robot encounters obstacles at least in two sides (for example in front of it and by one side making an L shape). In this case, it turns off the potential field term and turns on the A* algorithm. Once the robot is outside the entrapment region (there is no more obstruction between the robot and the target), the task *Formation* turns off the A* algorithm and turns back on the potential functions. The switching approach is implemented in the formation task as illustrated in Fig. 5.3. This switch is due to the fact that the potential functions have much less computational costs then using A* path planning and when the A* is executed it is necessary to stop processing some features of the controller in order to speed up the processing.

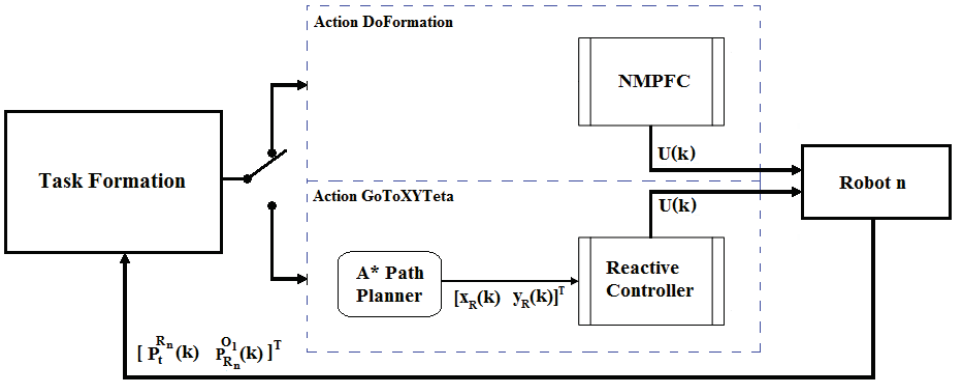

Figure 5.3: Switching approach

Finally, the same consideration made in the mate avoidance function has to be considered here by creating a security zone. An obstacle may appear in the visible zone towards the robot too rapidly for the robot to avoid it. To avoid collision in these cases a protection zone is created around the robots where the weight of the attraction and repulsion functions (terms of the NMPFC cost function) are rapidly switched so the robot gives priority to penalize the mate avoidance rather then get to the target. When the robots are outside this zone once again, the weights of the cost function are set back to normal.

5.4 The Modified A* Path Planning Algorithm

It is known that most environments are highly dynamic, highly complex and have obstacles moving randomly. Considering the dynamic constrains of the robot, the situation studied is often common in the real world, which is to find the fastest solution between the initial state q_{init} and the goal state q_{target} , avoiding as many collisions as possible. Therefore, one of the concepts that is necessary to highlight is that the best solution, in most cases, is not given by the shortest path (optimal path) and can lead to undesired collisions. In another words, the best solution is not the shortest path (the optimal one), but the fastest path (usually the suboptimal one). That is because the velocity of the robot is not constant (the robot has limited acceleration) and the robot controller has difficulty in following trajectories with abrupt changes in direction. To accomplish that, two modifications were made in the A* algorithm to achieve an optimal solution. The A* algorithm can be seen below.

Algorithm A*

1: Add the initial node to O-list
2: **Do**
3: **Choose** $n*$ from O-list in which
4: $F(n*) \leq F(n)\ \forall\ n \in$ O-list
5: Remove $n*$ from O-list and put into the C-list
6: **For** all $n \in$ Star($n*$), which $n \notin$ C-list, do:
7: **if** ($n \notin$ O-list) **then**
8: add n node to O-list
9: **else if** ($g(n*)+c(n*,n) < g(n)$) **then**
10: change the father node from n to $n*$
11: **end**
12: **end**
13: **While** (O-list $\neq$ 0) or ($n*$ = end node)

where:

1. Star($n*$) = The set of neighbors to node $n*$
2. c(n_1,n_2) = Cost from going from node n_1 to node n_2
3. $n*$ = Best note in the neighborhood

Therefore, A* is a graph search algorithm which calculates the shortest path through a graph between the initial and final node. This algorithm uses a heuristic function

$$F(n) = g(n) + h(n) \tag{5.8}$$

that estimates the lowest cost of going from the initial to the target point while passing through node n. This sets the search priority for nodes to find the best path as soon as possible. This function is the sum of two other functions.

1. $g(n)$ = Cost from the origin to node n;

2. $h(n)$ = An heuristic to estimate the cost of the path from node n to the target node.

In this algorithm there are two lists: the O-list and the C-list. The open list, known as the O-list, contains the nodes that are candidates for exploration. The closed list, known as the C-list, contains the already explored nodes. The nodes from C-list where previously in the O-list but as they where explored, they were moved to the C-list. The nodes in these lists store the "father" node, which is the node used to optimally reach them. This is the node that lies in the shortest path from the origin to current node.

Once presented the algorithm, it becomes important and necessary to emphasize some considerations. Mainly, it's known that the path τ is optimal if an heuristic function $h(n)$ is admissible. This happens if the function never overestimates the cost to reach the destination, or in another words if

$$h(n) < h_m(n) \forall n \tag{5.9}$$

where $h_m(n)$ is the lowest cost from n until the destination.

To use the A* algorithm in the calculation of a robot's path, it is necessary to divide the environment map in cells, as stated in the method approximate cell decomposition. Here, each cell represents a node. Each node can be connected to other nodes and moving from one node to the other has an associated cost. In this case, the cost is the metric distance between the cell centers. The A* can calculate the path that minimizes the cost from moving from the starting cell to the target cell. In Fig. 5.4 from [110], the black cells represent the obstacles, the yellow cell represents the initial position (node) and the blue cell represents the destination point (node).

Finally, it is considered that robot A in the workspace W is represented by the initial node and that occupies only a single node, being this last one the geometric center of the objects form an aerial view. The destination node is the target point q_{target}. All other moving objects are considered C-obstacles (CB_i). As the robot is represented by a single cell, the obstacles have to be bigger in a way to represent both obstacle and the robot's body. Each obstacle is represented by a circle with radius equal to the sum of the obstacle's radius and the robot's radius. This representation can be seen in Fig. 5.5 from [110].

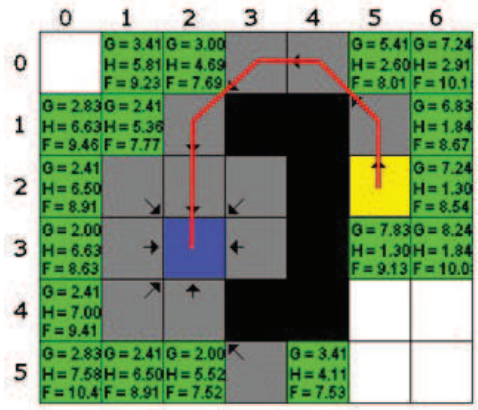

Figure 5.4: Map Cell Decomposition [110]

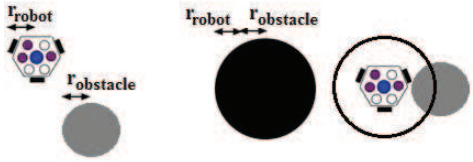

Figure 5.5: Obstacle's total radius [110]

5.4.1 The Modifications

The cell decomposition algorithms such as D* (and its evolutions such as D*-Lite and E*), ARA* and AD* are based in the A* and were developed to solve problems of computational cost, processing time or memory use. The modifications proposed by [110] and [119] are in the configuration space and not in the algorithm core itself. Therefore, in the matter of configuration space, all the previous algorithms form A* family should give an equal or similar solution to the A* algorithm. When applying the modification in any algorithm from the A* family, the final solution is better.

Finally, in this approach the authors in [110] and [119] base the modifications in the method of Cell Decomposition where the modifications are not in the A* algorithm but in the configuration space to later run A* algorithm to find the best path. The advantage comes with the fact that in the Cell Decomposition there are no local minimums, while in the VFH or in other similar approaches the local minima can become a problem when trying to avoid narrow areas. The global modified A* path planned proposed in [110] and [119] was the one used here in the leader robot and in the case of exiting entrapment situations (singularities).

5.5 Results

Both simulation and experiments with real robots results follow the same setup done in the previous chapter. The results regarding the obstacle and mate avoidance results are divided in simulation and experiment with real robots. Here, three situations are presented in simulation environment and are also used in experiments with real robots. Furthermore, the first two simulations are performed with both homogeneous and heterogeneous groups of robots.

5.5.1 Simulations

The simulation setup follows the same configuration as in chapter 4 with two computers and the SimTwo simulator environment. The static obstacles in the simulation were considered to be boxes with $0.75x0.75x0.5m$ (depth x width x height) in dimension. The moving obstacles are considered as self moving spheres in a semi-random path.

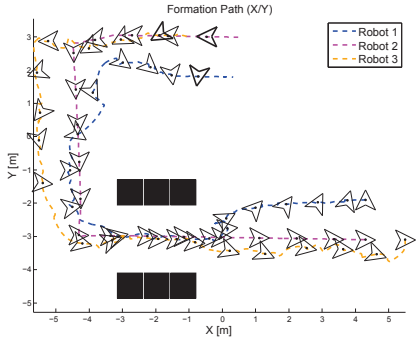

Figure 5.6: Simulation 1: Homogeneous Formation - Plot XY

5.5.1.1 Simulation 1: Homogeneous Formation

The first pair of simulations addresses the corridor problem in obstacle avoidance where the robots have to pass through a corridor while changing the formation in order to avoid the obstacles (corridor's walls). The objective here is to search for the ball departing from the coordinates (4.3,-1.9), (4.3,-3.1) and (5.5,-3.1) for the robots 1, 2 and 3, respectively. The obstacles' walls are made by six blocks in the coordinates (-2.8,-1.7), (-2,-1.7), (-1.2,-1.7), (-2.8,-4.5), (-2,-4.5) and (-1.2,-4.5) as presented in Fig. 5.6. Robot 2 is the leader robot in this simulation. A graph with the distance between the follower robot and the leader robot as well as the angle between them can be seen in Fig. 5.7.

By analyzing the simulation graphs and video **C2T1** attached, it can be said that the robots succeed breaking the formation not by switching it, but by a minimization criteria of avoiding obstacles and mates during the experiment. During the passage through the

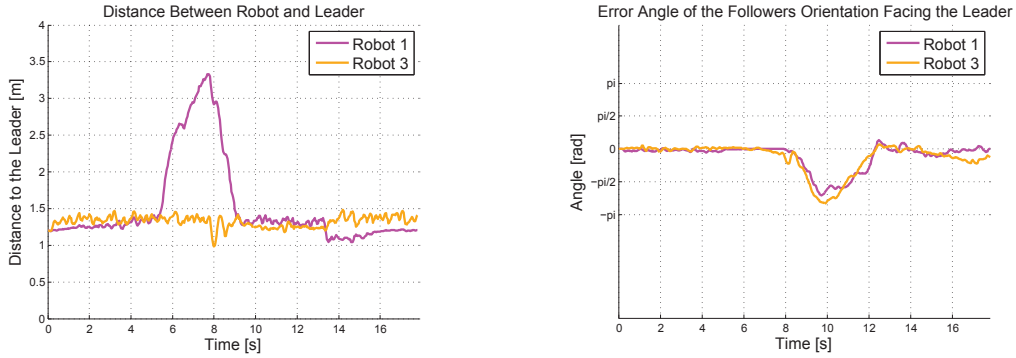

Figure 5.7: Simulation 1: Homogeneous Formation - Distance between Robot and Leader and Angle Between the Leader and the Followers

corridor, the robots form a column and the distance from robot 1 to the leader robot (robot 2) is temporarily increased. After passing through the corridor, the robots return to the desired formation successfully. The disturbance in the angle between the followers and the leader in the end of the graph of the error angle is due to the turn performed by the formation near the end of the trajectory.

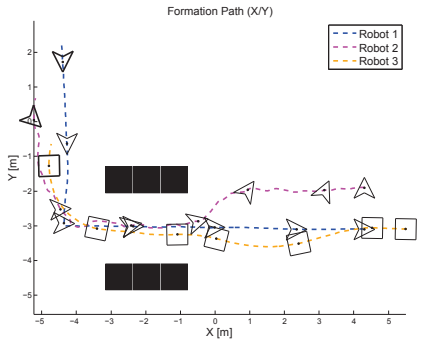

Figure 5.8: Simulation 1: Heterogeneous Formation - Plot XY

5.5.1.2 Simulation 1: Heterogeneous Formation

As in the previous simulation, the objective here is to search for the ball avoiding collisions with the mates and the walls of a corridor that exists in the formation path. Fig. 5.8 presents the path made by the formation in a XY plot, where the initial coordinates form robots 1 and 2 where exchanged and where robot 1 is the leader robot now.

Fig. 5.9 shows the graphs of the distance between the robot and the ball as well as the angle between them. Analyzing both figures, it can be noted that the simulation was stopped right after the first turn in the trajectory. The formation behaved as the previous one despite the nonholonomic robot's constraints in its movement. As presented in the

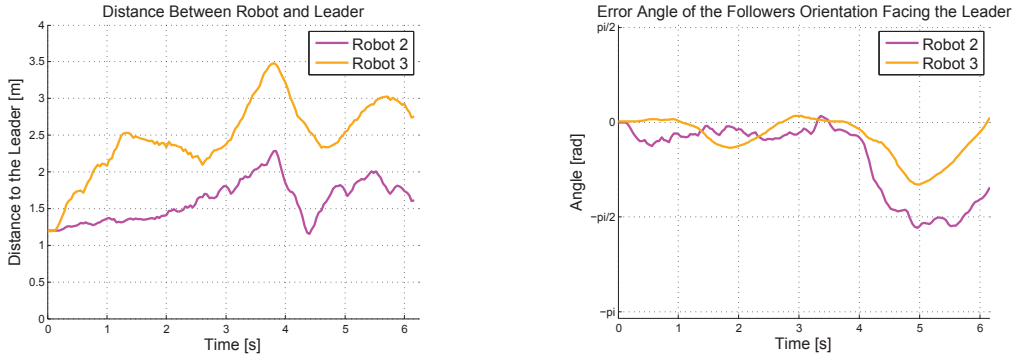

Figure 5.9: Simulation 1: Heterogeneous Formation - Distance between Robot and Leader and Angle Between the Leader and the Followers

video **C3T1Obs**, no collisions between the robots in the formation and between the robots and the obstacles occurred.

5.5.1.3 Simulation 2: Homogeneous Formation

This second pair of simulations addresses the entrapment problem in obstacle avoidance where the formation has to avoid an U shape obstacle during the convergence towards the target. In this pair of simulations, robot 2 is again the leader robot. The objective here is to converge the robots towards the ball departing from the coordinates (4.3,1.5), (-7.5,0), (5.5,0) and (-6.5,0) for the robots 1, 2, 3 and target, respectively. The obstacles' walls are made by nine blocks in the coordinates (-2.8,1.5), (-2,1.5), (-1.2,1.5), (-2.8,-1.5), (-2,-1.5), (-1.2,-1.5), (-3.6,0.8), (-3.6,0) and (-3.6,-0.8). In this case the leader robot is near the ball sending information to the other robots about where the target is located. Therefore, the NMPFC acts by attracting the robots and unavoidably puts the robots in a local minima as the potential function would behave as demonstrated in the path of robot 3 in Fig. 5.10. However, the modified A* path planner starts instantly when the robots enter this situation (at coordinates (-0.5,0)) and puts the robots outside the U shape obstacle.

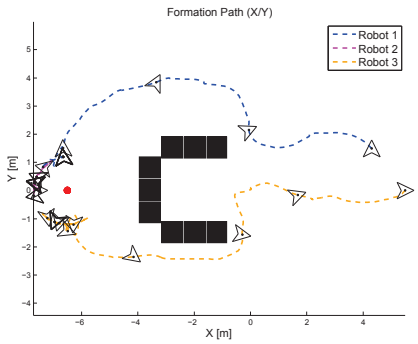

Figure 5.10: Simulation 2: Homogeneous Formation - Plot XY

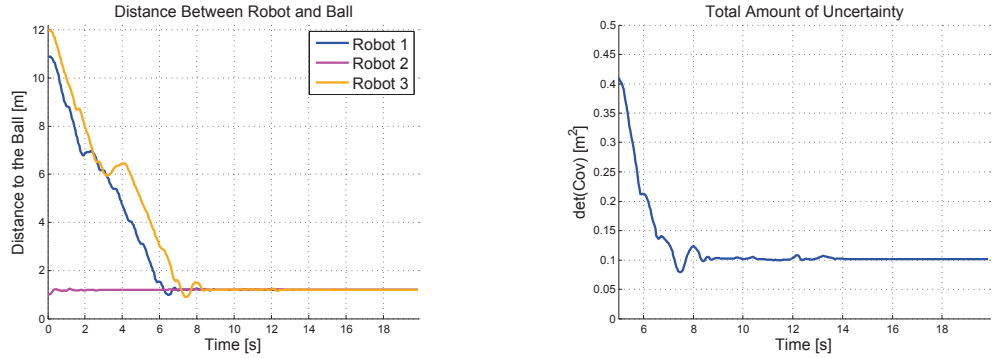

Figure 5.11: Simulation 2: Homogeneous Formation - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$

The video **C2T2A** shows the robots' movement when detecting a singularity point. While one of the robots detect the singularity situation and turns on the modified A* to leave the entrapment situation, the other robot (more aside the obstacle) succeeds avoiding it without the modified A*. Note here that the covariance only starts at approximately 5 seconds when all robots are seeing the ball. A graph with the distance between the robot and the ball as well as the minimization of the merged covariance determinant can be seen in Fig. 5.11.

5.5.1.4 Simulation 2: Heterogeneous Formation

This second simulation addresses the same problem in obstacle avoidance as the previous one, where the formation has to avoid a U shape obstacle. However, this is a heterogeneous formation where robot 3 was substituted by a nonholonomic robot.

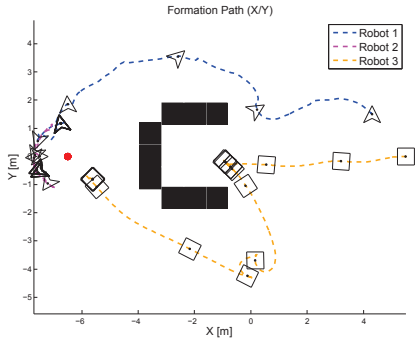

Figure 5.12: Simulation 2: Heterogeneous Formation - Plot XY

The path made by the robots can be seen in Fig. 5.12 and in the video **C3T2ObsA**. Note here that the nonholonomic robot 3 enters in the entrapment situation and leaves it while making a big trajectory due to its constraints in mobility. A graph with the distance between the robot and the ball as well as the minimization of the merged covariance

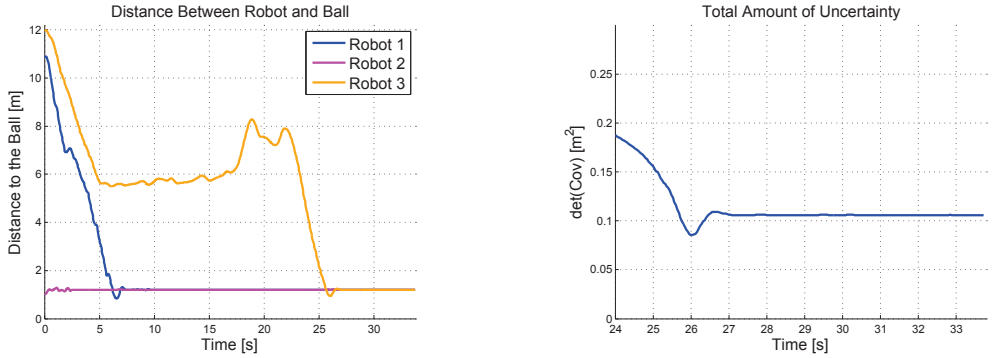

Figure 5.13: Simulation 2: Heterogeneous Formation - Distance between Robot and Ball and the Determinant of $\Sigma^{\perp}_{\text{Merged}}$

determinant starting at 24s (when all robots see the ball) can be seen in Fig. 5.13 proving the convergence of the formation.

5.5.1.5 Simulation 3

In this simulation, the used field has $8x10m$ of size. For the simulations in the unstructured environment the movement of the obstacles is set to be semi random. That's because those movements are set by a simple algorithmic procedure in SimTwo. As it can be seen in the *Set Sphere Speed Algorithm*, this uploads the values from each sphere and sets a force upon each one, making the movements. With a constant velocity for all the spheres, the movements of each one is practically always the same, except if the robot hits one or more sphere or if the spheres hit each other.

Algorithm Set Sphere Speed

If $\exists$ Sphere **then**
  $NexPosition_x$=$Initial_x$+Radius·cos(Speed·(t+δ))
  $NexPosition_y$=$Initial_y$+Radius·sin(Speed·(t+δ))
  Set Force of ith-Sphere with
    V_x = 100*($NexPosition_x$ - $Position_x$)
    V_y = 100*($NexPosition_y$ - $Position_y$)
    V_z = 0
  end
end

In the algorithm, $NexPosition_x$ and $NexPosition_y$ are the next position of the ith-sphere, $Initial_x$ and $Initial_y$ are the initial position of the ith-sphere in the simulation, $Position_x$

and $Position_y$ are the current position of the ith-sphere, Radius is the radius of the ith-sphere, Speed is the velocity modulus of the ith-sphere, t is the increasing time during simulation and δ is the orientation of the ith-sphere's movement. Finally, V_x and V_y are the new ith-sphere's velocities.

This third simulation demonstrates the algorithm's behavior in a crowded environment. The objective is once more to converge the formation towards the ball, here represented as a circle. The robot leader is near the ball and it passes the target's coordinates to the other robots. The simulation environment is as presented in Fig. 5.14 with the path made by the robots in a XY plot. Here no collision happened between the robots and the twelve mobile obstacles and the robots converged towards the target successfully.

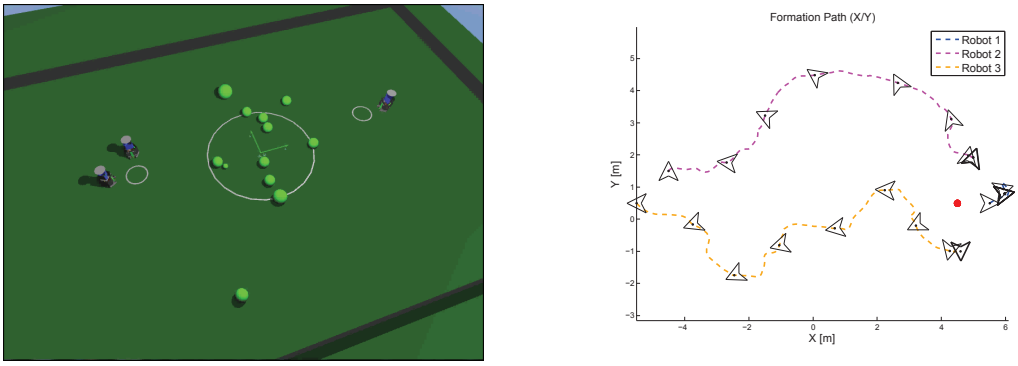

Figure 5.14: Simulation 3 - Environment and Plot XY

A graph with the distance between the robot and the ball as well as the minimization of the merged covariance's determinant starting at 4s (when all robots see the ball) can be seen in Fig. 5.15. The video **C2T3** shows the robots' movement as presented in the XY plot.

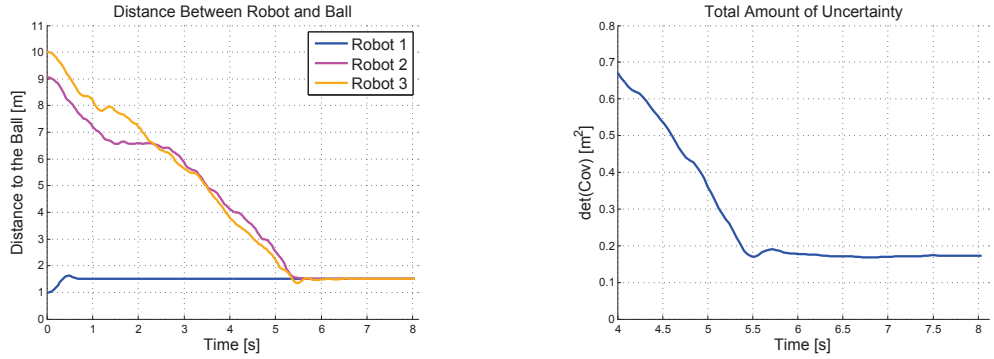

Figure 5.15: Simulation 3 - Distance between Robot and Ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}$

5.5.2 Results of the Experiments with Real Robots

The setup for the real robots experiments follows the same configuration as in chapter 4 with a computer running the Coach as a bridge and each real robot with its computer. The static obstacles were three walls of $0.7m$ height. The moving obstacles used in the experiments were two people passing through the robot's path. In these real experiments only two robots were used, robots 1 and 2. Finally, it is important to notice that now robot 1 is the leader robot.

5.5.2.1 Real Experiment 1

This experiment addresses the corridor problem in obstacle avoidance where the robots have to pass through a corridor changing the formation in order to avoid the obstacles (corridor's walls) in the same fashion as seen in the simulations 1. The objective here is to converge robot 2 towards the ball departing from the coordinates (-3,1.2) while robot 1 is placed near this target. In this case, robot 1 and the target are in the coordinates (-1,2.8) and (0,2.8), respectively. The obstacles' walls and the robots' movements are presented in Fig. 5.16 and in the video **C2T1Real** that shows that no collisions between robots or with an obstacle occurred.

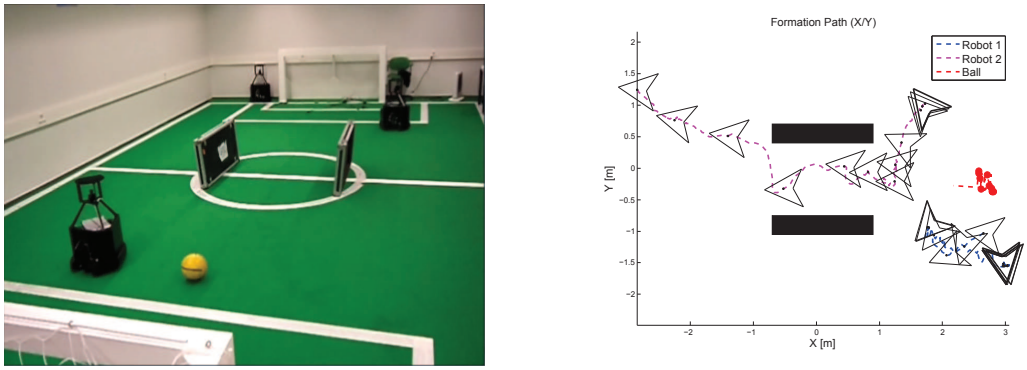

Figure 5.16: Real Experiment 1 - Environment and Plot XY

A graph with the distance between the robot and the ball and the minimization of the merged covariance's determinant can be seen in Fig. 5.17.

It is important to notice in this last graph that when robot 2 leaves the corridor its position is too close to the ball and too near both obstacles' walls. Nevertheless, as it is closer to the ball, the total merged covariance reaches a lower minimum. Then the controller puts robot 2 in a position more distant from the obstacles, at a ideal distance from the target in its perception, but it is a less minimized merged covariance. Moreover, the merged covariance can only be considered 3 seconds after the experiment had started. Both robots are able to see the target only after 3 seconds have passed.

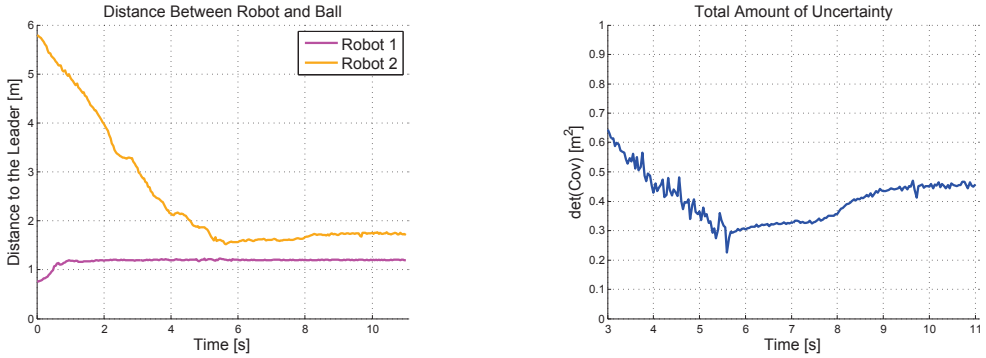

Figure 5.17: Real Experiment 1 - Distance between Robot and the Ball and Determinant of $\Sigma^{\perp}_{\text{Merged}}$

5.5.2.2 Real Experiment 2

This second experiment addresses the entrapment problem in obstacle avoidance where the robot has to avoid an U shape obstacle by changing the formation in order to avoid the obstacle. Similarly to simulation 2, the leader robot (robot 1) is near the ball, sending the information to the other robot (robot 2) of where the target is located. Therefore, the objective here is to converge to the ball departing from the coordinates (3,1.2), (-3,-0.2) and (3,0) for the robots 1, 2 and target, respectively. The NMPFC attracts the robot and unavoidably puts the robots in a local minima just as the potential function would behave. However, the A* path planner starts instantly when the robots enter this situation and places the robots outside the U shape obstacle. The walls and the robots' movements are presented in Fig. 5.18 and in the video **C2T2Real** that shows that robot 2 was successful in leaving the entrapment situation and that no collisions between robots occurred.

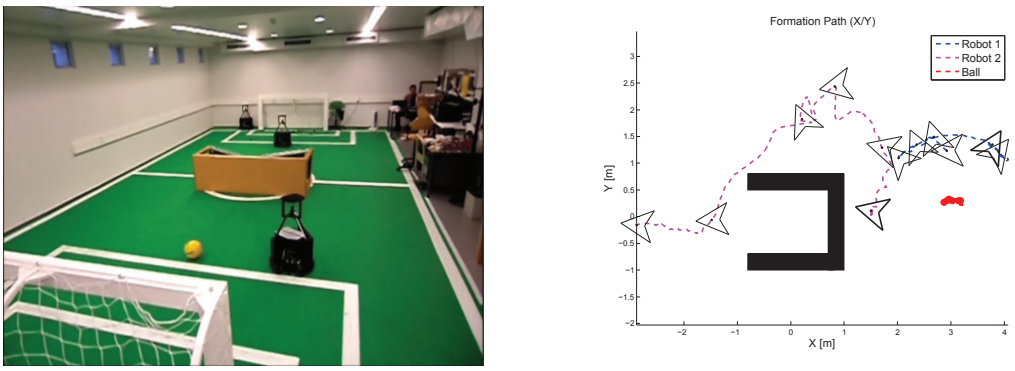

Figure 5.18: Real Experiment 2 - Environment and Plot XY

A graph with the distance between the robot and the ball and the minimization of the merged covariance's determinant can be seen in Fig. 5.11. It is noticeable here the same behavior from the NMPFC as in the previous experiment and due to the same reasons.

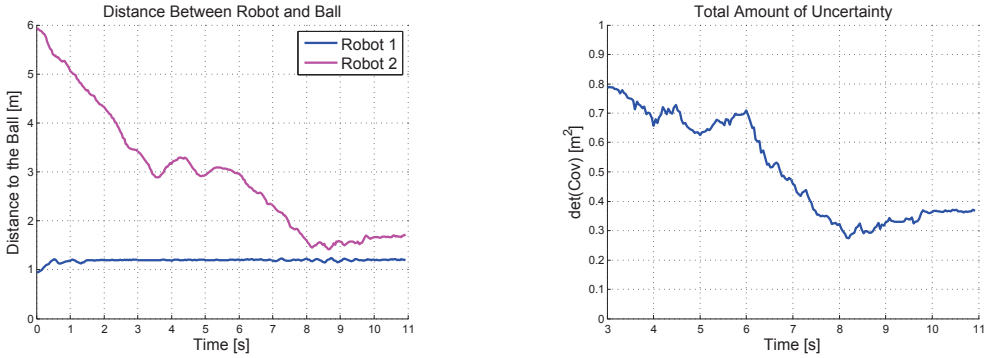

Figure 5.19: Real Experiment 2 - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$

Analyzing both video and the Plot XY presented in the Fig. 5.18 the path made by robot R_2 can be understood. During the convergence, robot R_2 gets trapped between the obstacle, the end of the field and robot R_1, which stood in front of robot R_2. Then, robot R_2 starts searching for a way through, always trying to converge to the target. As the minimization of the covariance is cooperative, robot R_1 moves itself into another pose allowing robot R_2 to pass and converge to the target. It can be seen through Fig. 5.18 that in the real experiment the convergence of both robots towards the target was accomplished.

In the convergence, it is common to see small errors in the distance between the robots and the target. These errors are explained by a few small vision problems that exist in the system (even after calibration) and which create errors in the localization of the target in the robot's frame. Furthermore, each robot has small localization errors, which, summed with the vision errors, generate the errors in the world frame and therefore each robot sees its ball in a different position. Finally, all the balls are then fused in the CTE and a fused target is generated and spread through RTDB, not eliminating the distance errors completely.

5.5.2.3 Real Experiment 3

This last experiment repeats the crowded environment in simulation 3. The objective is to converge robot 2 towards the ball. The robot leader (robot 1) is near the ball and it passes the target's coordinates to the other robot. The two persons passing through the robot's path are considered as mobile obstacles. The robot sees their legs and considered as black blobs avoiding them as any other obstacle. The path made by the robots can be seen in Fig. 5.20. The graph of the distance shows the robots' convergence towards the target. The covariance graph, starting from a distance where both robots see the ball (from 2s on) is also presented containing some noise due to the temporarily obstruction made by the obstacles in robot 2 target observation.

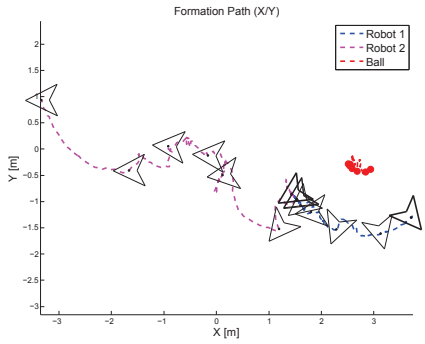

Figure 5.20: Real Experiment 3 - Plot XY

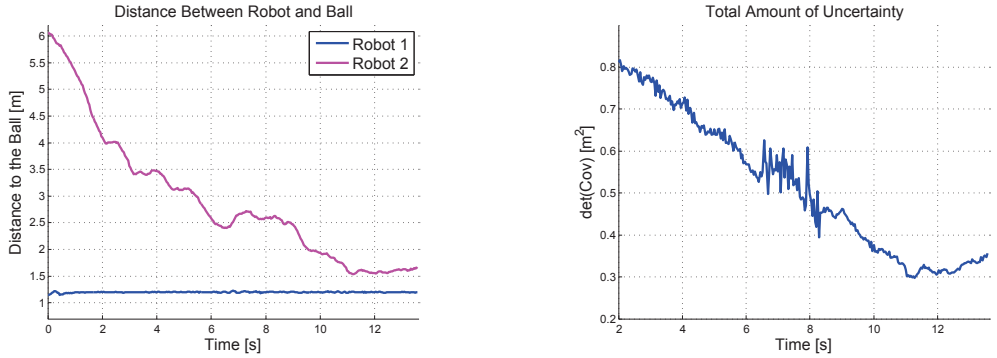

Figure 5.21: Real Experiment 3 - Distance between Robot and Determinant of $\Sigma^{\perp}_{\text{Merged}}$

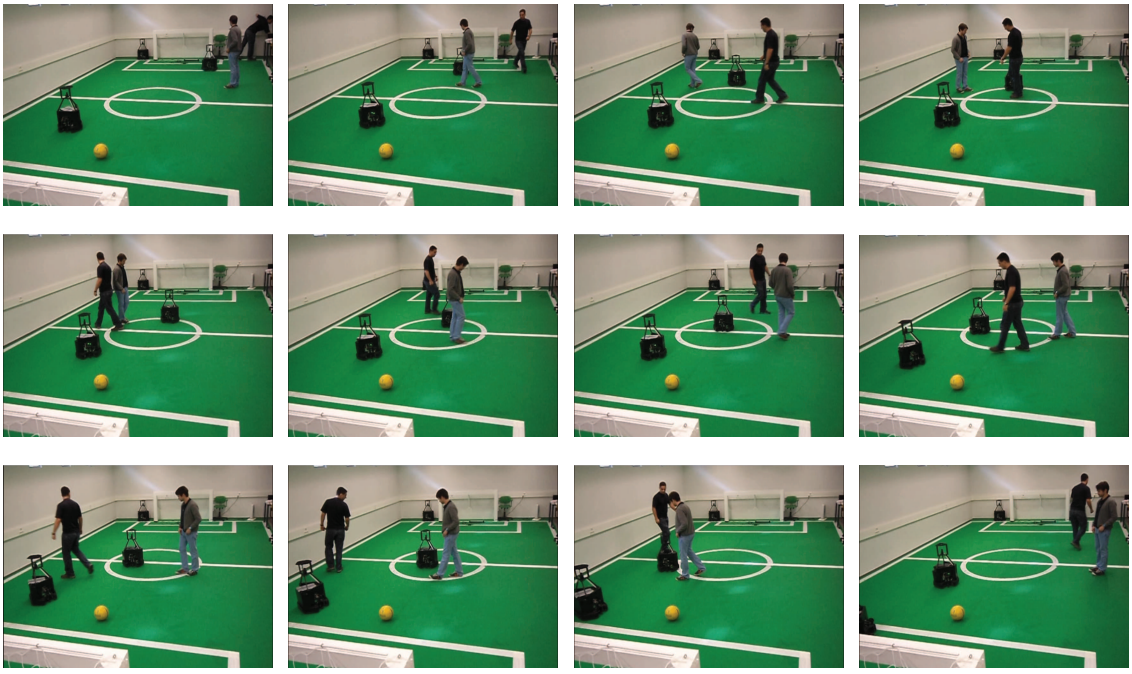

Figure 5.22: Real Experiment 3

A graph with the distance between the robot and the ball as well as the minimization of the merged covariance's determinant can be seen in Fig. 5.21. Furthermore, the screen shots of the experiments showing the robot's movement and the obstacles' movement are presented in Fig. 5.22. Moreover, the video **C2T3Real** also shows that no collisions happened. Note that in this experiment, as there is no obstacles pushing the robots afar, they put themselves in the ideal position for the merged covariance minimization.

5.6 Conclusion

This chapter had the main objective of better explaining the use of the approach while utilizing artificial potential fields (APF) considered as a term of the nonlinear model predictive formation control's cost function, in order to avoid static or moving obstacles. Regarding the singularity problems that are present when using the APF, the modified A* algorithm was considered as a complement for the potential field approach during these cases.

As the NMPFC already minimizes a cost function, the APF approach made feasible the solution for the obstacles and mates avoidance problem as a terms of the NMPFC cost function. Another contribution was introduced as a switching approach that, in case of singularities, uses a modified version of the A* path planner in order to exit the entrapment situation.

The results demonstrated the efficiency of this approach in formation control when embedded in the NMPFC cost function by successfully avoiding the obstacles in experiments both done in simulation and with real robots. In the corridor case, the formation of multiple robots could be seen avoiding two walls of (static) obstacles, breaking the desired formation by the obstacle avoidance penalization term and, after passing through the corridor, converging back in formation. The crowded environment demonstrated the same obstacle avoidance capability regarding mobile obstacles.

However, the APF approach also has its drawbacks regarding concave obstacles where local minima may occur. In such situations, the modified A* algorithm proved to be a feasible temporary alternative.

The main advantage of this approach is to consider in the same minimization problem both controller and obstacle/mate avoidance problem. This approach excludes the need of a path planner in the active target tracking problem. However, as disadvantage this approach needs to switch to a global path planner, in some special occasions (singularity), in order to leave an entrapment situation that may occur.

Finally, it can be concluded that despite the fact that in a single robot situation, a global path planner would have better results in obstacle avoidance problem, when considering a formation of a multi-robot system, this approach proved to be a better choice. This is

due to the fact that the formation uses only one cost function to minimize both formation behavior terms that perform the convergence towards a target and the obstacle/mate avoidance terms.

Chapter 6

Intelligent State Changing Applied to Multi-Robot Systems

The target searching problem is a situation where a formation of multi-robot systems is set to search for a target and converge towards it when it is found. This problem lies in the fact that the target is initially absent and the formation must search for it in the environment. During the target search, false targets may appear dragging the formation towards them. Therefore, in order to avoid the formation following a false target, this chapter presents a new methodology using the Takagi-Sugeno type fuzzy automaton (TS-TFA) in the area of formation control to solve the target searching problem. The TS fuzzy system is used to change the formation through the modifications in the states of the automaton. This change does not only switch the rules and therefore the state of each robot, but also the controllers and cost functions. This approach amplifies the versatility of the formation of mobile robots in the target searching problem. In this work, the TS-TFA is presented and its implications in the formation are explained. Simulations and results with real robot are presented where it can be noticed that the formation is broken to maximize the perception range based on each robot's observation of a possible target. Finally this work is concluded in the last section.

6.1 Introduction

Fuzzy systems and automatons are often applied in robotics. Several research proposals use fuzzy systems to control, navigate or evaluate robot systems and the approach most frequently used is the Mamdani fuzzy system [149], [150], [151] and [152]. Moreover, robotic soccer has become a good work-bench for testing artificial intelligence algorithms and it plays an important role in the progress of the intelligent control algorithm

field [153], [154], [155] and [156]. In order to avoid deviating the formation from following a false target, this chapter presents the use of a recently presented technique, the Takagi-Sugeno type fuzzy automaton (TS-TFA) [157]. The TS-TFA selects the roles (behaviors) applied to the formation control of three 5dpo soccer robots [123]. These robots, usually participants of RoboCup [158], are used here as a platform to demonstrate the performance of a formation during the search of a target when there is the commuting absence/presence of this target. The problem here lies in the fact that the target is initially absent. During the target search, false targets may appear dragging the formation towards them. Therefore, as a proposed solution for this problem, the fuzzy automaton contains three states which are sufficient in order to solve the problem in the proposed case of study. The fuzzy automaton does not only change the robots' role but it changes also the robots' cost function or controller (depending on the state change), while pursuing or searching for a target. A static leader approach was chosen for this work; nevertheless, the proposed contribution also works with a non-static leader approach. Simulations and experiments with real robots will also be presented in this chapter.

The positioning of robots in a dynamic environment is also another issue regarding changes of roles in multi-robot systems [159] and [160]. In the work of Akiyama et. al. [161], the authors proposed a novel agent positioning mechanism for the dynamic environments. They state that because the real-world problem is generally dynamic, suitable positions for each agent should be determined according to the current status of the environment. Therefore, the authors in [161] formalized this issue with a map from a focal point (like a ball position in a soccer field) to a desirable positioning of each player agent. Furthermore, they proposed a method to approximate this map using Delaunay Triangulation. The performance of the method was evaluated in RoboCup soccer simulation environment and compared to other function approximation methods such as the normalized gaussian network (NGN).

Furthermore, Stone and Veloso [162] introduced periodic team synchronization (PTS) domains as time-critical environments in which agents act autonomously with low communication, but in which they can periodically synchronize in a full-communication setting. The two main contributions of this article were a flexible team agent structure and a method for inter-agent communication in domains with unreliable, single-channel, low-bandwidth communication. In said paper, homogeneous agents can flexibly switch roles within formations, and agents can change formations dynamically, according to pre-defined triggers which are evaluated at run-time. This flexibility increases the performance of the overall team. Our teamwork structure further includes pre-planning for frequent situations. Second, the novel communication method is designed for use during the low-communication periods in PTS domains. Finally, they fully implemented both the flexible teamwork structure and the communication method in the domain of simulated

robotic soccer, and conducted controlled empirical experiments to verify their effectiveness.

Regarding the dynamic change of roles, the work of Reis et. al. [163], presents an interesting approach. In their paper, they proposed an approach for coordinating a team of homogeneous agents based on a flexible common team strategy as well as on the concepts of situation based strategic positioning and dynamic positioning and role exchange. The authors also introduced an agent architecture including a specific high-level decision module capable of implementing this strategy. Their proposal was based on the formalization of the idea of what a team strategy is for competing with an opponent team having opposite goals. Agent's reactivity was also introduced for appropriate response to the dynamics of the current situation. However, in their approach this was done in a way that preserves team coherence instead of permitting uncoordinated agent behavior.

The authors in [164] investigate the leader follower motion coordination of multiple nonholonomic mobile robots. A combination of the virtual vehicle and trajectory tracking approach was used to derive the formation architecture. A virtual vehicle was steered in such a way it stabilizes to a shifted reference position/heading defined by the leader, the velocity of the virtual vehicle was provided for further use in designing control law for the follower independent from the measurement of leader's velocity. Position tracking control was then constructed by the authors for the follower to track the virtual vehicle using the backstepping and Lyapunov direct design technique. Furthermore, and to ensure the safety of robots while moving in a dynamic environment, an obstacle avoidance scheme based on sensing the relative distance between follower robots and obstacles was introduced by [164] using fuzzy logic.

Changing roles in a multi-robot formation system requires a high level of navigation and control. Role assignment, according to the robot's features, is a crucial step in the coordination of multiple heterogeneous robots. The research of [165] presents a strategy for picking heterogeneous players and forming a soccer team in the RoboCup simulation environment, which is a multi-robot coordination research platform. Using fuzzy evaluation and fuzzy inference, the authors identify the most suitable role in a soccer team for a given heterogeneous robot. In the games, the team that possessed this strategy as its bases showed a significantly better performance when compared to a team based on the previous hand-tuned solution.

The authors in [166] presented a fuzzy logic based system for formation control of multiple mobile robots. Two main problems of formation control are investigated - maintaining correct formation position and inter-formation collision avoidance. A leader-follower approach with minimal communication between robots was presented here. Separate fuzzy logic controllers were developed by the authors for formation position control and internal collision avoidance with a higher level fuzzy coordinator used to fuse their outputs. A major issue in formation control is the presence of uncertainties in the real

world, in the form of noisy sensor data and delay in leader position transmission. Noise was added to the simulated data to prove that the system was capable of tolerating such disturbance.

In [167], the authors focus on the middle size soccer robot league (MSL) as well as on new hierarchical hybrid fuzzy methods for decision making and action selection of a robot. In article [167], the behavior of an agent was introduced, implemented and classified in two layers, the *Low Level Behavior* and the *High Level Behavior*. In phase one, the robot's situation is checked in order for a decision to be make on how to perform the required behavior. In the second phase, the team strategy, team formation, robot's role and the robot's positioning system are introduced. A fuzzy logic approach is adopted to provide the player with the best position to move based on the information given by the current state.

The inherent uncertainty present in robotics in general and RoboCup in particular demands the use of probabilistic methods. With its fuzzy constructs, Fuzzy Logic has been used as an approach to face the current problems of robotics including uncertainty. Extending the use of fuzzy logic with type-2 systems and high level world models should present new solutions to the robotics domain. In [168] the authors introduce an attempt to present a solution to the not fully resolved question of fusing actions together, using a fuzzy type-2 approach with the added flavor of semi qualitative world model.

An obstacle avoidance mechanism was developed by [169] for formation control of multiple mobile robots. This mechanism is designed based on the sequential situations when the robots likely have a chance to collide with each other. First, the obstacle avoidance is enabled if the distance between robot and obstacle is smaller than a designed distance and the obstacle is located on the way where the robot is moving to. Second, the time to turn on the mechanism for obstacle avoidance is decided by a predefined distance. Finally, an alternative path is generated for the robot to avoid obstacles by using a vector operation. Meanwhile, a robot behavior-based formation control was also constructed applying a three level architecture. The developed mechanism was then applied to the formation control of a group of omnidirectional robots.

The formation problem of a group of robots is one of challenging research directions of multi-robot system. Aiming at the formation control problem, [170] proposed a kind of hybrid architecture firstly, which combined the reactive architecture based on motor schema with the hierarchical architecture. Secondly four kinds of basic behaviors based on the reactive architecture had been designed, each behavior implemented by a fuzzy neural networks. Lastly, it was used nerve cell to fuse the outputs of each of behavior to obtain the final outputs of the system.

Usually, approaches to select roles use Mamdani fuzzy systems [171]. The paper written by [172] presented a new fuzzy-genetic analytical model for the problem of project team formation. It built on previous quantitative approaches, but added several modeling

enhancements like derivation of personnel attributes from dynamic quantitative data and complex attribute modeling. The authors improved the flexibility of requirement specification using a special format that expresses the required team capabilities using fuzzy descriptors. A single compound objective function was then defined, which incorporated multiple criteria that the solution should maximize. To optimize the selection of multiple project teams with possibly conflicting requirements, the authors proposed a special adaptation of island genetic algorithm with mixed crossover where the fitness of common solution was used to drive the selection within the islands.

In [173], an approach that gave the starting point for this contribution was presented. In [173], a new hierarchical hybrid fuzzy-crisp method for decision making and action selection of an agent in soccer simulation 3D environment was presented. First, the skills of an agent were introduced, implemented and classified in two layers, the basic-skills and the high-level skills. In the second layer, a two-phase mechanism for decision making was introduced. In phase one, some useful methods were implemented which check the agent's situation for performing required skills. In the next phase, the team strategy, team formation, agent's role and the agent's positioning system were introduced. A fuzzy logical approach was employed to recognize the team strategy and furthermore to tell the player the best position to move.

In all previously mentioned approaches it is assumed that the ball (target) is in the group observation range, where at least one robot is seeing the target. This situation does not occur in the target searching problem. To solve this problem, the TS-TFA was proposed and implemented. To describe the TS-TFA implemented here, this chapter is organized in the following order: The next section presents the state machine that represents the robot's possible behavior. Section 6.3 presents the TS type fuzzy automaton. The problem is formulated in the following section and in section 6.5 the results obtained both in simulation and with real robots are presented. Finally, the conclusion is presented in the last section.

6.2 The State Machine

The TS-TFA combines fuzzy sets and automaton theory. In one of the most popular applications of the automaton theory in robotics, a state machine is used to define a high level control (or navigation). This navigation system is based on the robot's change in behavior (e.g. changing from a standstill state to a moving state). In this chapter, the multi-robot system performs a search and track task of a target (ball). In this task, the ball can be within the robot's sensor range or not. Therefore, the multi-robot system changes its behavior from three states that can be seen in the state machine shown in Fig. 6.1.

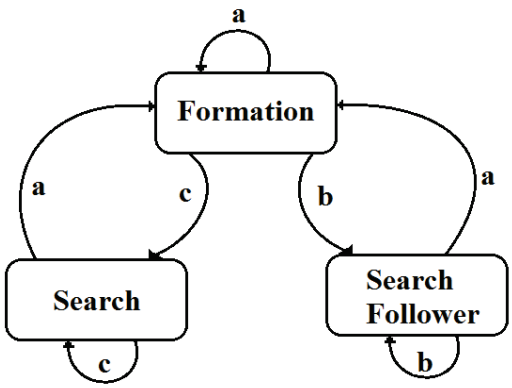

Figure 6.1: Robot State Machine

To better understand the TS-FSA implications, the state machine that governs the robots' behavior has to be explained first. Here, a deterministic finite automaton composed of three states governs the behavior of the robots in the target searching problem.

Therefore, consider the state machine $SM = (Q, \Sigma, \Phi, Q_0, F)$ which can be seen in Fig. 6.1, where Q is the set of all states, Σ is the input alphabet, Φ is the state-transition function that correlates the alphabet with the transition between states, Q_0 is the set of initial states and F is the set of final states. The following assumptions are made:

1. $Q := \{Formation, Search, SearchFollower\}$
2. $\Sigma := \{a, b, c\}$
3. $\Phi := Qx\Sigma \rightarrow Q$
4. $Q_0 := Q$
5. $F \subseteq Q$

Given Z as the evaluation of the situation generated by the fuzzy system, and L as the subscript identifying a robot n (where $1 \leq n \leq N$ and N is the total number of robots in formation) chosen to be the leader, the values for Σ are:

$$\begin{cases} & a, \quad If \quad Z = 1 \\ b, \quad If \quad Z = 0; & n, \forall\, n \neq L \\ c, \quad If \quad Z = 0; & n, \forall\, n = L \end{cases} \tag{6.1}$$

The *Formation* state is when the robot sees the target and starts moving towards it for improved observation and tracking. The *Search* state is when the leader robot ($n = L$) cannot see the ball and starts the procedure to search for a target, moving around the soccer field. The last state is the *SearchFollower* applied to any robot where $n \neq L$ that

can not see the target. When the *SearchFollower* is applied, the robot starts following robot *L*. In the *Formation* state, the robot uses the nonlinear model predictive formation controller (NMPFC) [133]. This controller minimizes the cost function that dictates the robot's behavior and penalizes it if the formation does not perform as expected. When the *Formation* state is activated, the robot visualizes the target as a leader and starts moving towards it. The final position of the formation (in a three robot case) should put all of the robots around the ball, forming a 120° (degree) angle between them when the ball is not moving. In the *SearchFollower* state, the robot also uses a NMPFC but with a different cost function that considers the leader robot and not the target, thus it moves towards the leader robot. Finally, in the *Search* state, the robot changes to a normal reactive controller [118] using the global path planner A* presented in chapter 5 to move around the field.

The aim of this work is to use the resulting output state value to change the state of the robot. The robot formation possesses three states, as previously seen. These three states are enough to solve the target searching problem where the change between absence/presence of a target becomes an issue. This is one of the main differences between this approach and the one in [157], where the authors use the output state as a linear function instead of a value. In this work the TS-fuzzy is simplified and instead of having an output fuzzy function, it has an output state. Therefore, if the resulting output state value is equal to 1, the resulting state is *Formation*(F), whereas, if the value is equal to 0 its resulting state is either *Search*(S) or *SearchFollower*(SF) depending on the value of Z from robot n, where n is the robot's number. If the robot's number is the chosen one to be the leader ($n = L$) then the final state will be *Search*(S). If $n \neq L$, then the final state will be *SearchFollower*(SF). This interaction can be seen in the look-up table below. The look-up table is a concise representation of the TS-TFA's behavior with respect to the state transitions defined through the dominant crisp states.

Table 6.1: Membership Functions-State Transitions. The values NV, VR and VM in the Quality (q) membership function means, respectively, *Not Visible*, *Visible Reasonably* and *Visible Much*, and HT, TR and ST in the Trust (t) membership function mean *High Trust*, *Trust Reasonably* and *Small Trust*, respectively.

q\ t	HT	TR	ST
NV	S/SF	S/SF	S/SF
VR	F	S/SF	S/SF
VM	F	F	S/SF

6.3 The Takagi-Sugeno Type Fuzzy Automaton

Tracking the status of an event-driven, large control system is a difficult problem. These systems often encounter unexpected roadblocks in an uncertain environment. The use of

a fuzzy automaton offers an effective approximation method to model continuous and discrete signals in a single theoretical framework using the combination of two techniques: the automaton theory and Takagi-Sugeno fuzzy systems. A Max-Min automaton can successfully model a cluster of relevant states when a decision must be made on the next state of a goal path at a supervisory level. However, to provide analytical proof for stability and other key properties of a fuzzy controller, a Takagi-Sugeno (TS) model is preferred. Finally, in [157] a TS-type fuzzy automaton is introduced.

The fuzzy automaton can remain in different fuzzy states simultaneously, to a certain degree in each. These degrees are defined by a state membership function. For each fuzzy state there is just one dominant (crisp) state for which the state membership is 1 (full membership). Each fuzzy state is associated with a linguistic label for inference. For each fuzzy state there is a fuzzy set that has a state membership degree greater than 0 in that fuzzy state [157].

The transitions between fuzzy states are based on the transitions defined between their dominant crisp states. There is an underlying Boolean finite state machine that implements the fuzzy automaton. The states of this Boolean automaton are the dominant crisp states of the fuzzy automaton. In the TS-TFA model, the Takagi-Sugeno version of the IF THEN rules are adopted [157]. Therefore,

$$R: If\, x_1\, is A_1, ..., x_k\, is A_k\, then Z = g_{Sk} \tag{6.2}$$

and g is a linear function adopted from [157] that, in this work, was defined to assume only two values

$$g_{Sk} = 0 \quad or \quad g_{Sk} = 1 \tag{6.3}$$

In each crisp state S_k the final output Z_{Sk} is calculated according to

$$[Z_{Sk} = Z^i_{Sk}] = (A^i_1(x^0_1) \wedge ... \wedge A^i_k(x^0_k)) \wedge [R^i] \tag{6.4}$$

$$Z_{Sk} = \frac{\sum [Z_{Sk} = Z^i_{Sk}] \times Z^i_{Sk}}{\sum [Z_{Sk} = Z^i_{Sk}]} \tag{6.5}$$

where $[\cdot]$ means the truth value of proposition $\cdot$, the $\wedge$ operator is implemented as min and $A(x^0)$ stands for the grade of membership of x^0 in fuzzy set A. For simplicity, $[R^i] = 1$ is assumed. Furthermore, $\times$ means a scalar multiplication. The notion of the composite output Z^* is introduced to reflect the contribution of the output values devised from the Takagi-Sugeno linguistic models that are attached to crisp states to the final output in a fuzzy state. Let the TS-TFA be in a fuzzy state S_{Fk}, then

$$Z^*_k = \frac{\beta^k_1 Z_{S1} + ... + \beta^k_p Z_{Sp}}{\sum_i \beta^k_i} \quad (i = 1, ..., p) \tag{6.6}$$

It is clear from equation 6.6 that only crisp states that have greater than 0 degrees of membership in fuzzy state S_{Fk} contribute to the final output. Therefore, a TS-TFA automaton with p states is defined as follows [157].

$$S_{Fk} : S_k, g_{Sk} \tag{6.7}$$

$$R_S = TS(X_F, Z_S) \tag{6.8}$$

$$G = \begin{bmatrix} \beta_1^1 & \cdots & \beta_p^1 \\ \vdots & \cdots & \vdots \\ \beta_1^p & \cdots & \beta_p^p \end{bmatrix} \tag{6.9}$$

$$Z^* = TS(X_F^0, R_S, G) \tag{6.10}$$

$$X_T \; is \; TRUE \; if \; X_A \geq X_{AT} \tag{6.11}$$

$$X_B = B(X_F) \tag{6.12}$$

$$Y_B = f_y(X_B, W_B, X_T, y_B) \tag{6.13}$$

$$U_B = f_u(X_B, W_B, X_T, y_B) \tag{6.14}$$

Where S_{Fk} stands for fuzzy state k, S_k represents crisp state k and g_{Sk} is the state membership function associated with $S_k (k = 1, ..., p)$. β_1^k,..., β_p^k stand for the degrees of state membership function g_{Sk}. G stands for the matrix of state membership functions. The computational algorithm for equation 6.10 is given by equations 6.4-6.6. Finally, X_F^0 stands for inputs values evaluated by the member function of the fuzzy sets (A). The variables W_B and X_A stand for two-valued (boolean) and analog inputs with associated X_{AT} threshold values, respectively. A threshold comparator module compares the value of each analog signal with its associated threshold value to set the corresponding X_T signal as true or false. U_B stands for two-valued (boolean) outputs. X_B, Y_B and y_B stand for two-valued boolean inputs, next states and the present states of the state variables, respectively.

6.4 Problem Formulation

In order to avoid deviating the formation from following a false target, we present the use of the TS-TFA technique to select roles applied to formation control of a multi-robot system composed of three 5dpo soccer robots [123]. The fuzzy automaton will not only choose the robot's role, it will also select the robot's cost function or controller (depending on the state change), while pursuing or searching for a target. In the 5dpo robot soccer team, the architecture of each robot is composed of a Coach (a state machine that changes the robot's behavior) inside each computer that runs parallel with the Dec (controller) and the HAL (vision system). These systems are explained in more detail in [123]. The

state machine used that governs the robot's behavior and a brief explanation of the TS-TFA theory have been presented to outline the proposed solution. In this work, intelligent state changing is applied in the formation control of multi-robot systems to solve the target searching problem. To accomplish this task, a TS-TFA was applied using the state machine previously explained in this chapter. This state machine is governed by a set of rules extracted from the TS fuzzy system. The switch between states performed by the fuzzy rules assumes that each state in the state machine is a possible scenario that the robot would be in each instant of time. In turn, each scenario is the result of a fuzzy rule. Finally, each fuzzy rule performs a weighted calculation based on the measurements from the robots distance to the target and the robot's quality of the target perception measurements. In this TS model the implications (TS fuzzy rules) are as follows:

$$\begin{aligned}
R_1 &: If\ x_1\ is\ NV\ and\ x_2\ is\ HT\ then\ Z = 0 \\
R_2 &: If\ x_1\ is\ NV\ and\ x_2\ is\ TR\ then\ Z = 0 \\
R_3 &: If\ x_1\ is\ NV\ and\ x_2\ is\ ST\ then\ Z = 0 \\
R_4 &: If\ x_1\ is\ VR\ and\ x_2\ is\ HT\ then\ Z = 1 \\
R_5 &: If\ x_1\ is\ VR\ and\ x_2\ is\ TR\ then\ Z = 0 \\
R_6 &: If\ x_1\ is\ VR\ and\ x_2\ is\ ST\ then\ Z = 0 \\
R_7 &: If\ x_1\ is\ VM\ and\ x_2\ is\ HT\ then\ Z = 1 \\
R_8 &: If\ x_1\ is\ VM\ and\ x_2\ is\ TR\ then\ Z = 1 \\
R_9 &: If\ x_1\ is\ VM\ and\ x_2\ is\ ST\ then\ Z = 0
\end{aligned} \tag{6.15}$$

where x_1 is the fuzzy input that represents the quality of the robot camera's visual on the target. A camera perception quality decay range that stretches from -100 to 1000 was created for the 5dpo robot soccer team based on the number of pixels present in the target's visualization that represent the target [123]. Moreover, the input x_2 is the confidence in which this observation is made with respect to the distance between the robot and the estimated target position. If the robot is near the target, it means that its observation has high confidence and if it is far, it has low confidence. The visibility distance usually varies from 0 to 10 meters (maximum) and this is the range of confidence in estimation.

Each robot in the formation measures the position and speed of the ball (the state of the ball) in local frame. This information is sent to the Coach software that is located in a central computer and merges the information originated from the robots into a single ball state in the world frame. Afterwards, that is shared among the robots in the formation through wireless communication. All robots' poses are also shared throughout the formation. It is assumed that the merging of the ball position is performed using Smith and Cheeseman's formulation [132] and it provides the fused data of the ball speed and position, as well as the quality of this data. In each rule, the weight given to each measurement (distance of quality of perception) can be seen through a set of membership functions, which in turn can be seen in 6.2.

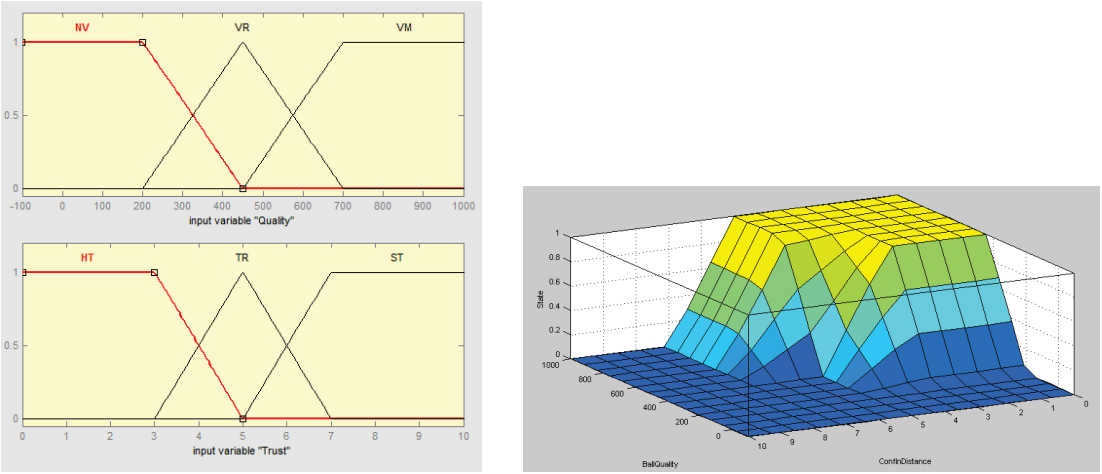

Figure 6.2: Membership Functions and Surface

6.4.1 Membership Functions

The membership functions can be seen in Fig. 6.2. These membership functions are designed by considering the quality decay of target perception from the robot's vision software (HAL). It is important to notice that the present work does not focus on the vision system. The vision system delivers the blob of the target as a set of pixels with a center point that in turn is translated into a quality decay range. This range is used to create the membership functions to evaluate the quality of the observed target [174], [175].

The Trust (confidence) function represents the reliability in the quality of the target's observation based on the distance between the robot and this observed target. Therefore, the Trust (confidence) function represents the reliability of the vision based on the distance to the target. If the distance is too far, the trust (confidence) is low but if it is close, the trust (confidence) is high. This membership function is rated in meters of distance. The membership functions are adjusted to the maximum performance by setting the functions proportionally, as per the implementation in real robots. In Fig. 6.2 for example, by analyzing the Trust function based on the robot's vision, it is noticeable that the value that separates the state of seeing and not seeing the target is 5. In the Quality function of target perception, the value is 450. In the Trust function, between 3 and 7 meters the confidence in the visual is regular, for less than 3 meters the confidence is strong and any measurement above 7 meters is not considered. The same situation happens with the quality function of target perception.

6.4.2 The Problem Example

The example below was created to better understand the contribution of this chapter and only to illustrate a situation where the existence of a false target would influence the for-

mation's behavior. Here, robot 1 is the leader robot, while robots 2 and 3 are the followers. Robots 1, 2 and 3 depart from coordinates (4.5,-3), (4.5,-1) and (6.5,-3), respectively. To simulate the appearance of a false target during the search process, the ball was initially placed in the coordinates (2,2.9). After a while during the simulation the ball is changed to the coordinates (-4.5,6) where the "true" ball would be located.

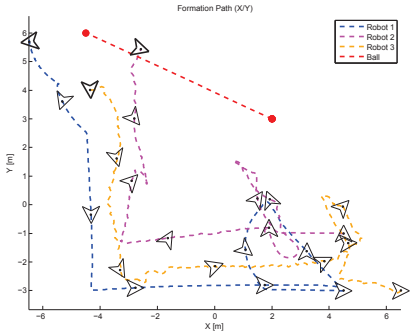

Figure 6.3: The Problem Simulation: Formation Behavior without the TS-TFA approach

This simulates a situation where the robot or the whole formation sees a false ball and, getting closer to the observed object, discards it as the target (either by the object format or any other vision process that occurs with the color segmentations process used to recognize the ball). In this case, without the proposed approach, the formation would be as shown in Fig. 6.3.

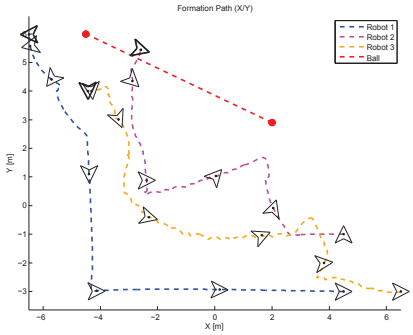

Figure 6.4: The Solution Simulation: Formation Behavior with the TS-TFA approach

Note that in this example, one of the robots sees the false target and sends the information to the other robots. Then, all robots in formation try to converge towards this false target. When the formation approaches the observed object, the robots consider this as a false target and have to restart the search once more. However, when applying the proposed technique, the behavior of the formation is improved. Instead of all the formation converging to a false target, only the robot that sees it weights between the quality of its observation and the distance that it is from the observed object. Then, this robot breaks itself from the formation trying to converge towards the possible target while the

other robots keep searching. By analyzing Fig. 6.4, it can be seen that when the observed object is considered as a false target, the robot converges back to the the formation which was still searching for a "true" target. Once the formation finds the "true" target, it converges successfully.

6.5 Results

The results are presented in two sections. The first outlines the simulation results using the SimTwo simulator [117]. In the simulations the robots (three omnidirectional 5dpo soccer robots) should achieve a final formation configuration around the ball, forming a 120° (degree) angle between them. The three simulations presented here are three different starting situations: *Simulation 1: Robot 3 is relatively near to the target and robots 1 and 2 are relatively far from the target*, *Simulation 2: Robot 2 is relatively near to the target and robots 1 and 3 are very far from the target* and *Simulation 3: Robots 1, 2 and 3 are very far from the target*. Furthermore, the subsection 6.5.2 presents the results of the experiment with real robots, where the real 5dpo soccer robots were used. The experiment was only conducted with the last simulation case to validate this theory. As a safety measure, the maximum robot velocity allowed was 0.7m/s.

6.5.1 Simulations

This section presents all three simulations. It is important to remember that here the ball is always stopped and robot 1 is the robot leader. All the setup made in chapter 4 for the simulation environment was also made here.

6.5.1.1 Simulation 1

In this simulation, the ball was placed at the coordinates (2,3.5) and robots 1, 2 and 3 were placed at the coordinates (4.3,-3.1), (4.3,-1.8) and (4.3,1.6), respectively. All robots have an initial $\theta = 270^o$ in world frame. Fig. 6.5 shows that robot 3 maintained its initial state *Formation* and moved towards the ball using the NMPFC, without ever changing its state. Then, robot 1 receives *Search* as its initial state with a modified A^* path planner and a reactive controller and it kept this state until it reached the coordinates (-3.2,2.8) where its confidence was high enough to change its state to *Formation*.

Finally, robot 2 receives *SearchFollower* as its initial state with a NMPFC controller with the cost function to follow a leader. Meanwhile, as the formation moves, robot 2 passes through coordinates (3.5,-2) where it was close enough to the target to change its state to *Formation*. If other robots see a target, each robot weighs up the global quality of the ball perception and its distance from the target. If the robot is too far from the target,

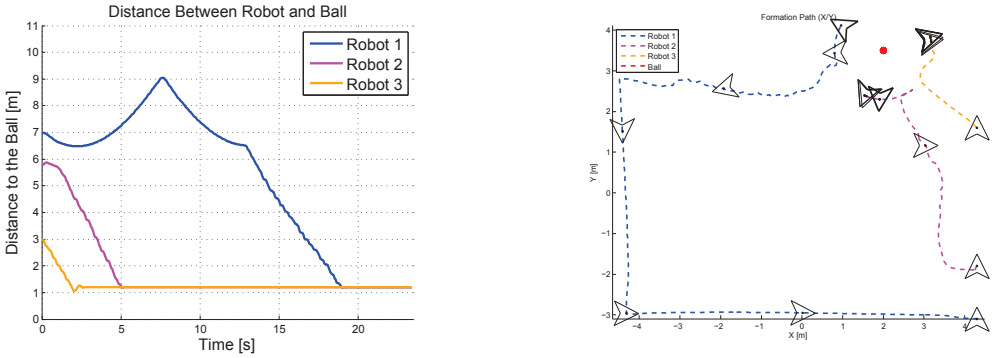

Figure 6.5: Simulation 1: Distance Robot-Ball and Plot XY of the Robots Path

it will not change its state, as shown in table 6.1. Here, the change in state corresponds to a change in the controller.

By analyzing Fig. 6.5, it can be seen that there is no disturbance in the robots' trajectories or any presence of instability in the state changes. The controller is the same but the cost functions are different and once again, no instability could be noted. The distance to the target and their path in an XY plot can be seen in Fig. 6.5, respectively. The behavior of the robots in this situation can be seen in Fig. 6.6 where it is possible to see the change in states during each instant taken from the Coach's view of the simulations.

6.5.1.2 Simulation 2

Here, Fig. 6.7 shows, through the Plot XY, that the ball was placed at the point with coordinates (-5.3,0) and robots 1, 2 and 3 were placed at the coordinates (4.3,-3.1), (-6.3,0) and (4.3,3.1), respectively. Furthermore, all robots have an initial $\theta = 90^\circ$ in world frame. Note in the XY Plot from Fig. 6.7 that there is no disturbance in the robot's trajectory and there is no presence of instability in the state change. In the distance graph from Fig. 6.7, it can be seen that the robot converged successfully towards the target.

The XY Plot from Fig. 6.7 demonstrates that robot 2 maintained its initial state *Formation* and moved towards the ball using the nonlinear model predictive formation controller (NMPFC). Furthermore, robot 1 received *Search* as its initial state with an A^* path planer and a reactive controller and it maintained this state until it reached a position where its confidence was high enough to change its state to *Formation* (coordinates (1.1,-3)). It is important to note that the change in state is also a change in the controller and no instability could be seen in the robot's path. However, the "irregularity"' seen in the path of robot 1 is due to the change in its orientation around 5 seconds after the start of the experiment, as it can be seen in Fig. 6.8. The rotation is decided and governed by the low-level control system.

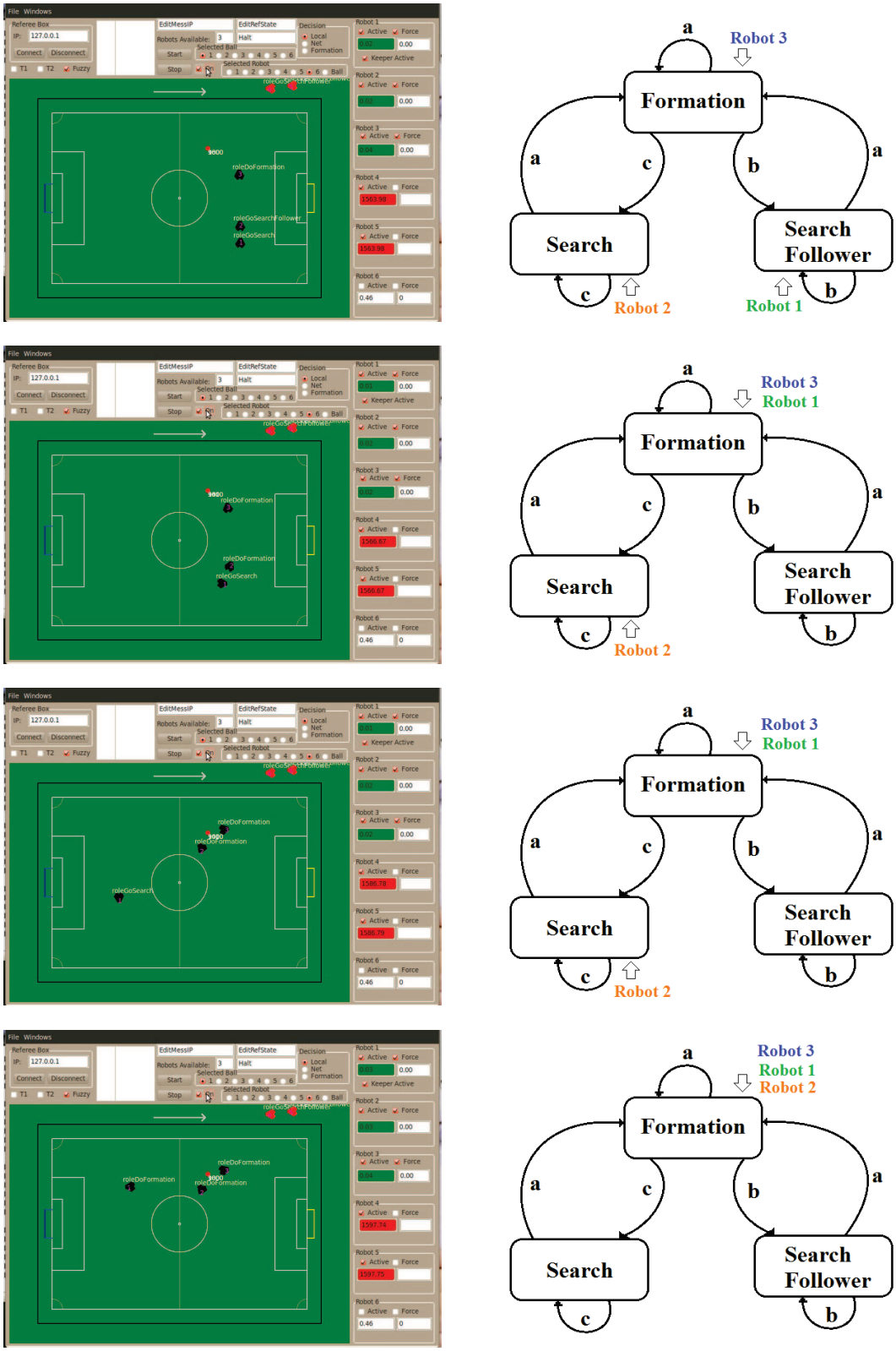

Figure 6.6: Simulation 1

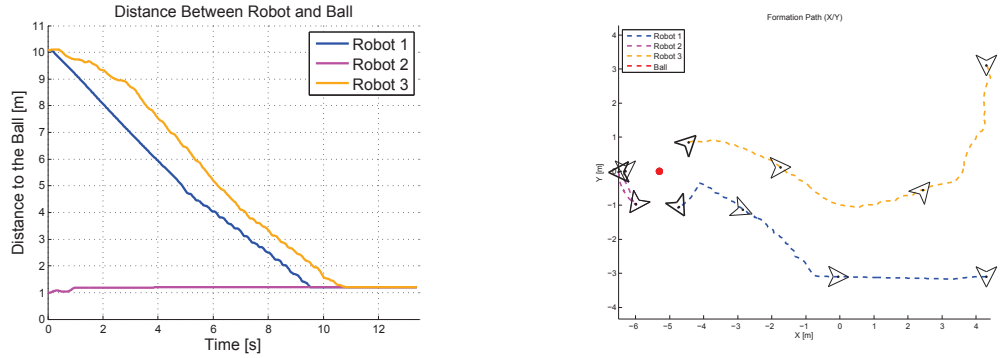

Figure 6.7: Simulation 2: Distance Robot-Ball and Plot XY of the Robots Path

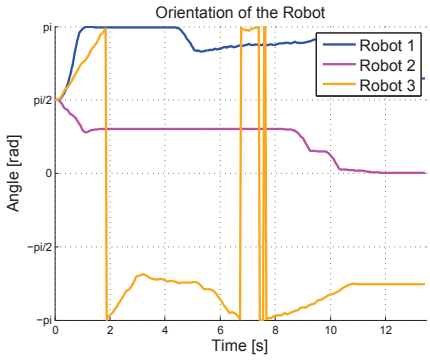

Figure 6.8: Simulation 2: Orientation of the robots

Finally, robot 3 receives *SearchFollower* as its initial state and it starts moving towards the robot leader using a NMPFC with a cost function to follow the leader. As it can be seen in Fig. 6.7, when robot 3 reached the point (0,-1), it places itself with enough distance to the target in order to change its state to *Formation*. Here, the controller is the same but the cost functions are different and once again, no instability can be noticed.

6.5.1.3 Simulation 3

The last simulation case started with robots 1, 2 and 3 on one side of the field placed at coordinates (4.3,-2), (4.3,0) and (4.3,2), respectively. Robot 1 is the leader robot here and the ball was positioned at coordinates (-3,1). Furthermore, all robots have an initial $\theta = 270^\circ$ in world frame. Note in Fig. 6.9 that there is no disturbance in the robot's trajectory and there is no presence of instability in the state change.

By analyzing Fig. 6.9 this experiment shows that the robots started performing the search for the target by moving around the field. When robots 1 and 3 reached the coordinates (-0.4,-3.1) and (-0.1,-1.4) respectively, they began to see the target, which in turn changed the state of each robot. Therefore, these two robots started the process of pursuing the target. At this instant robot 2 could not see the target yet. Meanwhile, robot

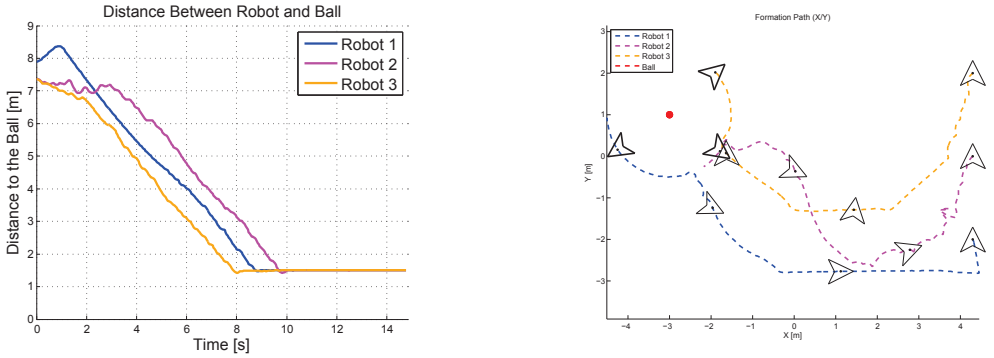

Figure 6.9: Simulation 3: Distance Robot-Ball and Plot XY of the Robots Path

2 was a little bit behind and continued looking for the target and following robot 1 until the group gained more confidence in the observation while in turn robot 2 saw the target as well.

6.5.2 Results of the Experiments with Real Robots

This case was similar to the last simulation case where robot 1 was the leader and the search should be performed by departing from an target absent state.

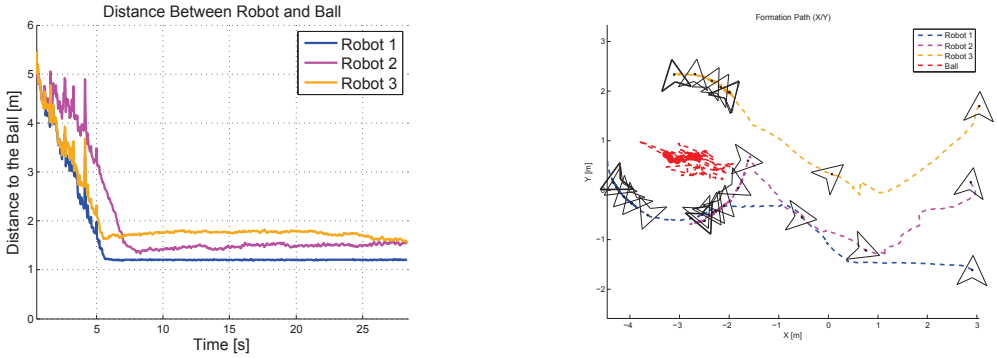

Figure 6.10: Real Experiment: Distance Robot-Ball and Plot XY of the Robots Path

However, as the real field of experiments is small and the fact that the search routine makes the robot leader perform a square path in the field independent from its size the initial coordinates for robots 1, 2 and 3 were respectively (3,-2), (3,0) and (3,2). The small difference between this experiment and the last simulation is that the robots depart with a greater distance in the X axis which in turn does not represent any significant difference for the comparison between simulation and real experiment. Moreover, Fig. 6.10 shows the distance from the robot to the ball during both states *Search/SearchFollower* and *Formation*.

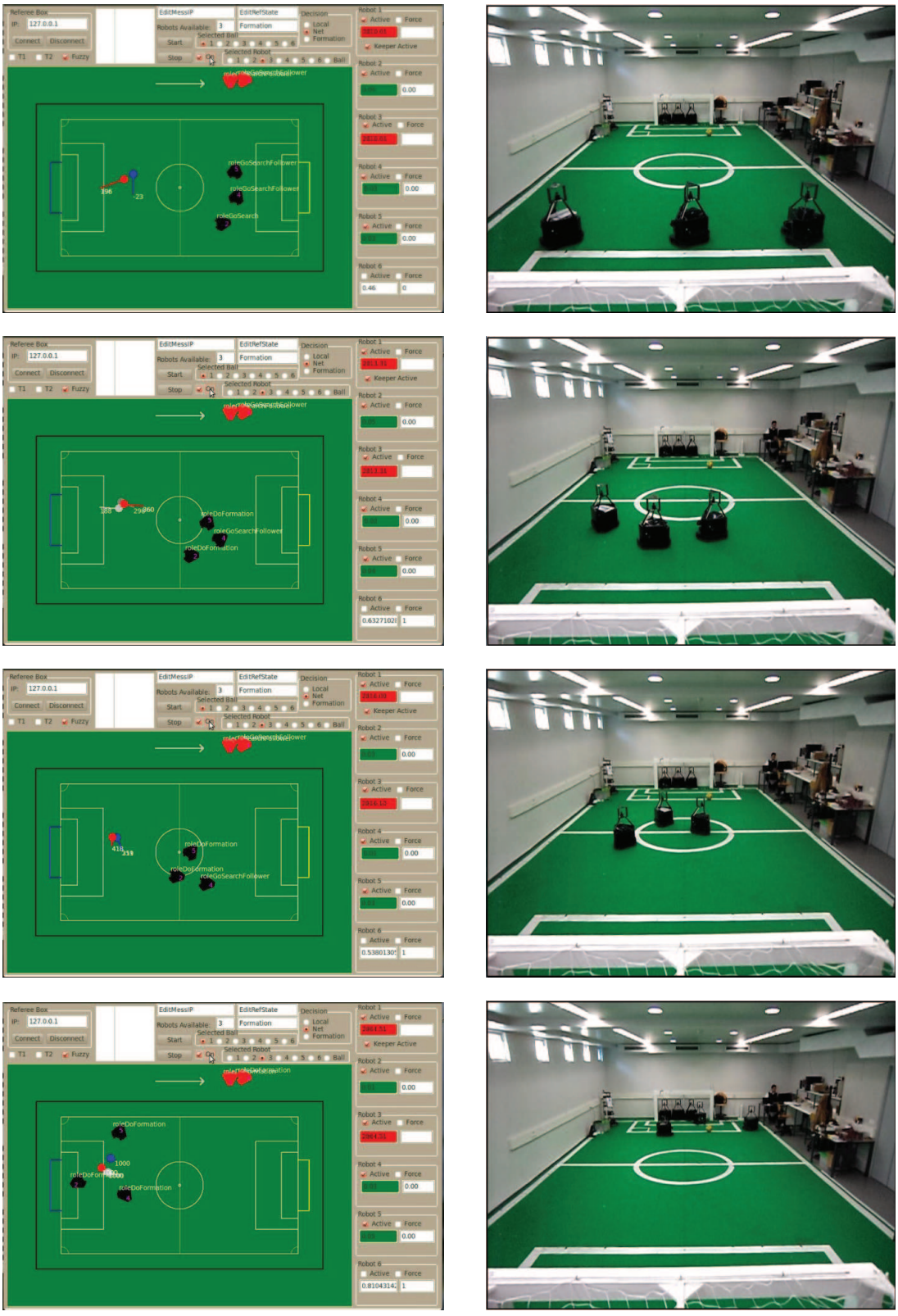

Figure 6.11: Real Experiment: Coach View

It is important to note that before changing to state *Formation*, the quality of the target's perception is affected by various vision noises due to the large distance between the robot and its target and the camera precision. This fact is demonstrated by the XY Plot and by the distance graph from Fig. 6.10, and it makes the importance of this contribution more notable. In this experiment, the robots started performing the search for the target by moving around the field. When the robots reached a certain point in the field (coordinates (0.2,-1.5) for robot 1 and (0.5,0.2) for robot 3 which were the firsts to see the target as seen in Fig. 6.10), the robots that reached this specific distance changed their state and started to pursue the target. Meanwhile, robot 2 was a little bit (1.2m) behind and continued looking for the target and following robot 1 until it saw the target as well. By analyzing the distance graph in Fig. 6.10, the main difference between the simulation and the real robot experiment are noticeable. Here, the variation in the perception of the target can be seen while the robots are searching and coincidentally approaching the target. It is the variation in the quality of the perception that makes the fuzzy system discard the ball positions. The point where the target is reasonably visible is reached in approximately four seconds when it can also be noticed that the perception quality has improved. The system works rapidly and smoothly, preventing the robots from colliding, which is possible at high velocity. However this is not the case when vision perception is not of a high quality. The XY plot of the movement of the robots is shown in Fig. 6.10. In this figure, the positions of the ball do not signify that the target moved, they only mean that the position of the target measured by the robots was not precise until the robot approached the target. This lack of precision in measurements makes the ball perception move in this plot. The behavior of the robots in this situation can be seen in Fig. 6.11.

6.6 Conclusion

This chapter presented a new methodology, the Takagi-Sugeno type fuzzy automaton (TS-TFA), applied to the formation control of multi-robot systems to solve the target searching problem where the formation is put to search for a target and they converge to it when it is found. This problem lies in the fact that the target is initially absent and the formation of a multi-robot system must search for this target in the environment. During the target search, false targets may appear. Therefore, in order to avoid the formation following a false target, this chapter presented the use of a recently presented technique, the TS-TFA. This technique selects one of the three roles applied to formation regarding the target searching problem. The fuzzy automaton not only changed the robots' roles (automaton states) but it also changed the robots' cost function or controller (depending on the state change), while searching for a target.

The TS-TFA was modified slightly in order to adapt to the formation control theory with the leader-following approach. From the simulations and the results with real robots

it can be noted that the formation is broken to maximize the perception range based on each robot's observation of a possible target. It can also be noted that the change in the controller and the change in the cost function do not make the system unstable in the representative cases presented. Finally, this application can be generalized and the formation can consider all sorts of environment applications, including an outdoor environment where in many circumstances, a false target can be found.

Chapter 7

Conclusions and Future Work

This chapter concludes the work of this thesis. A final observation and commentaries are made throughout the conclusion. Some suggestions for future works are also made in this chapter in order to give continuity to the research in formation control of multi-robot systems.

7.1 Conclusion

The main objective of this thesis was the formation control of a group of mobile robots; said control minimizes the total amount of uncertainty of a target's perception converging towards a desired position while avoiding collisions with the mates, obstacles and the target. The thesis was focused on the formation with holonomic and nonholonomic mobile robots and followed a research line that modeled the formation system, selected and conceived a formation controller and approached the addressing roles for breaking formation. This work was structured to present a summary in the beginning of each chapter and the obtained results and conclusions in detailed fashion in the end of each chapter. Therefore, the summary of this work and its main conclusions are as follows:

- The first part of this work modeled the 5dpo middle size league omnidirectional mobile soccer robot in the SimTwo environment. This task was important to tune and validate the theory proposed in this work. The modeling and validation were proven accurate;
- A nonlinear model predictive formation control (NMPFC) was conceived to perform a formation control on a group of mobile robots as the following step. The distributive configuration was created using wireless communication, where each robot received data from the other robots and processed its own task in the formation without the need of any kind of supervisor.

- Two cost functions were created for this work. The first cost function considers the target as a leader in formation and tries to minimize the total amount of uncertainty in perception of this target, while converges the robot towards this element in a desired pose, avoids collisions with mates and obstacles and penalizes oscillations in the control effort. The second cost function uses a robot as the leader, when the target is absent, following the leader robot in search for the target.

- The NMPFC also showed that the vision system can influence in the controller's efficiency. Small errors in the ball detection and localization are reflected in the calculation of the distance between the robot and the target and in the covariance minimization. Nevertheless, the controller was successful in converging the robots to the desired pose, minimizing the total amount of uncertainty while avoiding mates and setting the correct pose for all robots. Both convergence and keeping the formation were such that the total amount of uncertainty in the target's perception was minimized. Regarding the experiments without a target, the follower robots were successful in converging towards the leader and keeping the formation while the leader robot was moving despite the weights not being tuned with a criteria that contemplated the internal product term.

- A methodology was also proposed to search for a set of weights for the NMPFC's cost function. This methodology proved to be effective. Nevertheless, a tuning aiming at the situation regarding the following leader was not performed showing both the controller's robustness and a non optimal convergence of the internal product term (position term) of the cost function.

- The NMPFC was also proved capable of being applied in holonomic and nonholonomic robots with different sizes when joining a multi-robot system in formation. The models were implemented and used to predict the robot's velocities. These are simplified first order models used to calculate the robot's velocity dynamics. As the controller actuates directly on the robot's velocities rather then the motor's velocities, it enabled the controller to be generalized;

- The NMPFC was proven to be able to make the robot avoid collisions with static obstacles, moving obstacles and other robots in the formation by using a artificial potential field approach through repulsive potential functions embedded in the controller's cost function. To avoid entrapment situations (local minima problem), a modified version of the A* global path planner algorithm was used to lead the robot out of the entrapment situation. This same algorithm was used as a way for the leader robot to search for a target;

- A TS (Takashi-Sugeno) type fuzzy automaton was used to change cost functions or controllers accordingly to the presence or absence of the target as well as to evaluate

the change in the role given to the robot in order to break or not the formation in search for a possible target. The TS-TFA is present in the Coach and its function is to assign the roles of each robot separately. If a robot sees a target and it is in disagreement with the formation (the other robots do not see it), despite sharing the information, the robot weights between the quality of its perception and its distance from what it sees. If needed, the robot moves towards the possible target. The formation keeps searching for another target if disagrees with the robot that is seeing a possible target. This weighting between the quality of perception of each robot and their distance from an observable possible target is continually performed online.

7.2 Future Work

Many variations of this work can be executed in order to experiment and try to improve the formation control system. Some considerations are presented in this section.

7.2.1 Changeable Leader Robot

When considering multi-robot systems in formation one has to considered either a static or a changeable leader. In this thesis it was chosen a static leader as a matter of simplicity. Nevertheless, an changeable leader approach can improve the formation in many situations such as outdoor environments or soccer games. In target pursue problems where the target is a non-controllable object, to consider this kind of approach (e.g. tracking a missile by heat sensors) can improve the system by maximizing the coverage area of pursuit.

7.2.2 Model Considerations

Another consideration to be made is the possibility of using the NMPFC controller with different models. For instance, the model of the robot used in [130] could be tested.

7.2.3 Parameters Tuning

The tuning methodology proposed in this thesis was performed using as testbed a static ball case where the criteria of tuning did not consider the position term with respect to the velocity of the target nor the case of following a robot leader. Therefore, in such cases and as future work, it is suggested the tuning of the cost function accordingly to a criterion that considers these cases.

7.2.4 Heterogeneity Considerations

Another consideration about the multi-robot system is the use of aerial and marine robots. This NMPFC is prepared to control quadrotors, but it could also potentially control marine robots due to its capacity to control the formation without considering the actuators, only the kinematic model of the robot. Therefore, it is a possibility of future works to use the NMPFC in a multi-robot system with aerial and marine robots in the heterogeneous group in formation.

7.2.5 Target Observation

Finally, the solution in the active target tracking problem would be improved if the merged target spread trough the robots would also contain its merged covariance. This would give a more accurate measurement of the target instead of simulating all the covariances of the robots in formation.

7.3 Final Considerations

The final objective of this thesis was the formation control of a group of mobile robots which minimizes the total amount of uncertainty of perception of a target. This work was planned in four main steps. The first step was to validate the simulation model in SimTwo environment of the 5dpo mobile robots so that the simulations would represent accurately the robots' behavior. The second step conceived the nonlinear model predictive formation control for the formation with a visible target (ball or robot leader). The third phase of this work took into account the obstacle and mate avoidance problem. For that step, a hybrid approach was introduced between artificial potential fields used in the NMPFC and the modified A* path planning algorithm to be used in the leader robot and in local minima problems of the APF. Finally, the last step of this thesis considered the intelligent change of role applied when the formation does not see the target and suddenly starts seeing it. This algorithm avoided the formation to pursue a false target breaking the formation to verify its veracity.

Finally, the success in each step of this research was fundamental to accomplish the next step. The last step of this thesis was executed during an internship in Italy which provided even more knowledge to this work. All phases of this work were tested with real robots and simulation environments proving the viability of this work's contribution. Therefore, this work provides an important contribution that, through a set of tested and implemented techniques and algorithms, can be used in problems related to robotics and control theory as well.

Appendix A

The Covariance Model

Here it is presented the statistical development from equation (4.17) to equation (4.18) presented in the chapter 4.

A.1 Transformation in the Canonical Form

Let's analyze the figure below.

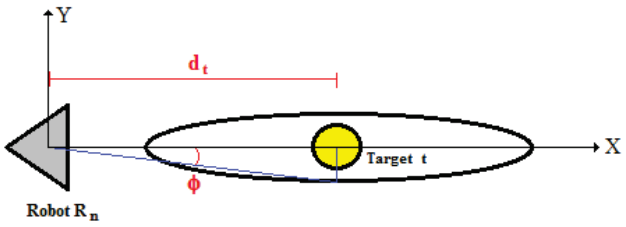

Figure A.1: Model of Observation

Based on Fig. A.1, it can be assumed that in a general fashion, to a generic omnidirectional camera:

$$\Sigma_{R_n} = \begin{bmatrix} \sigma_{d_t}^2 & \rho\sigma_{d_t}\sigma_{\phi} \\ \rho\sigma_{\phi}\sigma_{d_t} & \sigma_{\phi}^2 \end{bmatrix} \tag{A.1}$$

where σ_{d_t} is the variance of the target's distance measurement d_t, σ_{ϕ} is the variance of the target's bearing measurement and ρ is the correlation coefficient. It is assumed that both the measurements are uncorrelated and $\rho = 0$.

Therefore,

$$\begin{cases} \sigma_{d_t}^2 = K_1 {d_t}^2 \\ \sigma_{\phi}^2 = k_a \frac{1}{d_t} \end{cases} \tag{A.2}$$

Nevertheless, assuming that

$$\begin{cases} E(\tilde{d}_t) = d_t \\ E(\phi) = 0 \end{cases} \tag{A.3}$$

where $\tilde{d}_t$ is a random variable. Then

$$\begin{cases} E(x) = E(\tilde{d}_t.cos(\phi)) \\ \quad = E(\tilde{d}_t).E(cos(\phi)) \\ \quad = E(\tilde{d}_t).cos(E(\phi)) \\ \quad = d_t.1 \\ E(x) = d_t \end{cases} \tag{A.4}$$

and

$$\begin{cases} E(y) = E(\tilde{d}_t.sin(\phi)) \\ \quad = E(\tilde{d}_t).E(sin(\phi)) \\ \quad = E(\tilde{d}_t).sin(E(\phi)) \\ \quad = d_t.0 \\ E(y) = 0 \end{cases} \tag{A.5}$$

Knowing that

$$\sigma_{a.b}^2 = E(a)^2.\sigma_b^2 + E(b)^2.\sigma_a^2 + \sigma_a^2.\sigma_b^2 \tag{A.6}$$

and that

$$\sigma_{f(a)}^2 = (f'(E(a)))^2.\sigma_a^2 \tag{A.7}$$

we can use equation A.6 as follows

$$\begin{cases} \sigma_x^2 = \sigma_{d_t.cos(\phi)}^2 \\ \sigma_{d_t.cos(\phi)}^2 = E(\tilde{d}_t)^2.\sigma_{cos(\phi)}^2 + E(cos(\phi))^2.\sigma_{d_t}^2 + \sigma_{d_t}^2.\sigma_{cos(\phi)}^2 \end{cases} \tag{A.8}$$

and equation A.7 allows us to conclude that

$$\begin{cases} \sigma^2_{cos(\phi)} = (-sin(E(\phi)))^2.\sigma^2_{\phi} \\ \sigma^2_{cos(\phi)} = (-sin(0))^2.\sigma^2_{\phi} \\ \sigma^2_{cos(\phi)} = 0 \end{cases} \tag{A.9}$$

Then, coming back to equation A.8 we have

$$\begin{cases} \sigma^2_{d_t.cos(\phi)} = E(\tilde{d_t})^2.\sigma^2_{cos(\phi)} + E(cos(\phi))^2.\sigma^2_{d_t} + \sigma^2_{d_t}.\sigma^2_{cos(\phi)} \\ \sigma^2_{d_t.cos(\phi)} = E(\tilde{d_t})^2.0 + E(cos(\phi))^2.\sigma^2_{d_t} + \sigma^2_{d_t}.0 \\ \sigma^2_{d_t.cos(\phi)} = cos(E(\phi))^2.\sigma^2_{d_t} \\ \sigma^2_{d_t.cos(\phi)} = cos(0)^2.\sigma^2_{d_t} \\ \sigma^2_{d_t.cos(\phi)} = \sigma^2_{d_t} \\ \sigma^2_{d_t.cos(\phi)} = \sigma^2_{x} = \sigma^2_{d_t} \\ \sigma^2_{x} = k_1.{d_t}^2 \end{cases} \tag{A.10}$$

To calculate σ^2_y, it is necessary to consider once more equation A.6 as follows

$$\begin{cases} \sigma^2_{y} = \sigma^2_{d_t.sin(\phi)} \\ \sigma^2_{d_t.sin(\phi)} = E(\tilde{d_t})^2.\sigma^2_{sin(\phi)} + E(sin(\phi))^2.\sigma^2_{d_t} + \sigma^2_{d_t}.\sigma^2_{sin(\phi)} \end{cases} \tag{A.11}$$

and with equation A.7 it can be concluded that

$$\begin{cases} \sigma^2_{sin(\phi)} = (cos(E(\phi)))^2.\sigma^2_{\phi} \\ \sigma^2_{sin(\phi)} = (cos(0))^2.\sigma^2_{\phi} \\ \sigma^2_{sin(\phi)} = 1.\sigma^2_{\phi} \\ \sigma^2_{sin(\phi)} = \sigma^2_{\phi} \end{cases} \tag{A.12}$$

then, coming back to equation A.11 we have

$$\begin{cases} \sigma_y^2 = \sigma_{d_t.sin(\phi)}^2 = E(\tilde{d}_t)^2.\sigma_{sin(\phi)}^2 + E(sin(\phi))^2.\sigma_{d_t}^2 + \sigma_{d_t}^2.\sigma_{sin(\phi)}^2 \\ \sigma_y^2 = E(\tilde{d}_t)^2.\sigma_\phi^2 + sin(E(\phi))^2.\sigma_{d_t}^2 + \sigma_{d_t}^2.\sigma_\phi^2 \\ \sigma_y^2 = (E(\tilde{d}_t)^2 + \sigma_{d_t}^2).\sigma_\phi^2 \\ \sigma_y^2 = ({d_t}^2 + k_1.{d_t}^2).\frac{k_a}{d_t} \\ \sigma_y^2 = (1+k_1).{d_t}^2.\frac{k_a}{d_t} \\ \sigma_y^2 = (1+k_1).k_a.d_t \\ \sigma_y^2 = k_2.d_t \end{cases} \tag{A.13}$$

Finally,

$$\begin{cases} \sigma_x^2 = k_1.{d_t}^2 \\ \sigma_y^2 = k_2.d_t \end{cases} \tag{A.14}$$

References

[1] W. B. Dunbar and R. M. Murray. Model predictive control of coordinated multi-vehicle formations. In *Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada - USA*, pages 4631–4636, 2002.

[2] H.L. Sng, G. Sen Gupta, and C.H. Messom. Strategy for collaboration in robot soccer. In *Proceedings First IEEE International Workshop on Electronic Design, Test and Applications '2002, Christchurch, UK*, pages 347–351, 2002.

[3] Petter Ögren and Naomi Ehrich Leonard. A Convergent Dynamic Window Approach to Obstacle Avoidance. *IEEE Transactions on Robotics*, 21(2):188–195, 2005.

[4] Lovekesh Vig and J.A. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649, 2006.

[5] J Shao, G Xie, and L Wang. Leader-following formation control of multiple mobile vehicles. *IET Control Theory & Applications*, 1(2):545–552, 2007.

[6] Yunfei Zou, Prabhakar R. Pagilla, and Ryan T. Ratliff. Distributed Formation Control of Multiple Aircraft Using Constraint Forces. In *2008 American Control Conference, Seattle, WA, USA*, pages 644–649, 2008.

[7] Nicolas Lechevin, C A Rabbath, and Marc Lauzon. A Decision Policy for the Routing and Munitions Management of Multiformations of Unmanned Combat Vehicles in Adversarial Urban Environments. *IEEE Transactions on Control Systems Technology*, 17(3):505–519, 2009.

[8] Gianluca Antonelli, Filippo Arrichiello, Stefano Chiaverini, and Studi Cassino. Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control. *IEEE Transactions on Control Systems Technology*, 17(5):1173–1182, 2009.

[9] Laura E Barnes, Mary Anne Fields, and Kimon P Valavanis. Swarm Formation Control Utilizing Elliptical Surfaces and Limiting Functions. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 39(6):1434–1445, 2009.

[10] K. D. Do. Output-feedback formation tracking control of unicycle-type mobile robots with limited sensing ranges. *Robotics and Autonomous Systems*, 57(1):34–47, 2009.

[11] Luca Consolini, Fabio Morbidi, Domenico Prattichizzo, and Mario Tosques. Stabilization of a Hierarchical Formation of Unicycle Robots with Velocity and Curvature Constraints. *IEEE Transactions on Robotics*, 25(5):1176–1184, 2009.

[12] He Bai and John T Wen. Cooperative Load Transport : A Formation-Control Perspective. *IEEE Transactions on Robotics*, 26(4):742–750, 2010.

[13] H. Lim, Y. Kang, J. Kim, and C. Kim. Formation control of leader following unmanned ground vehicles using nonlinear model predictive control. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore*, pages 945–950, 2009.

[14] Yangmin Li and Xin Chen. Leader-formation Navigation with Sensor Constraints. In *Proceedings of the 2005 IEEE International Conference on Information Acquisition, Hong Kong and Macau, China*, pages 554–559, 2005.

[15] Yongxing Hao and Sunil K. Agrawal. Planning and control of UGV formations in a dynamic environment : A practical framework with experiments. *Robotics and Autonomous Systems*, 51(1):101–110, 2005.

[16] H. Fukushima, K. Kon, and F. Matsuno. Distributed Model Predictive Control for Multi-Vehicle Formation with Collision Avoidance Constraints. In *Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain*, pages 5480–5485, 2005.

[17] Yu Gu, Brad Seanor, Giampiero Campa, MArcello R. Napolitano, Larry Rowe, Srikanth Gururajan, and Sheng Wan. Design and Flight Testing Evaluation of Formation Control Laws. *IEEE Transactions on Control Systems Technology*, 108(6):1105–1112, 2006.

[18] Liu Shi-cai, Tan Da-long, and Liu Guang-Jun. Formation Control of Mobile Robots with Active Obstacle Avoidance. *Automatica*, 33(5):529–535, 2007.

[19] Bo Liu, Tianguang Chu, Long Wang, and Guangming Xie. Controllability of a LeaderâĂŞFollower Dynamic Network With Switching Topology. *IEEE Transactions on Automatic Control*, 53(4):1009–1013, 2008.

[20] João Rodrigo Alvelos Ferreira. *Controlo Coordenado de Equipas de Robots Móveis*. Master, Porto University, 2010.

[21] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

[22] T. H. Chung, J. W. Burdick, and R. M. Murray. A decentralized motion coordination strategy for dynamic target tracking. In *Proceedings 2006 IEEE International Conference Robotics and Automation, Pasadena - USA*, pages 2416–2422, 2006.

[23] R. Olfati-Saber. Distributed tracking for mobile sensor networks with information-driven mobility. In *Proceedings of American Control Conference, Hanover, Germany*, pages 4606–4612, 2007.

[24] P. Yang, R. A. Freeman, and K. M. Lynch. Distributed cooperative active sensing using consensus filters. In *Proceedings 2007 IEEE International Conference Robotics and Automation, Evanston - USA*, pages 405–410, 2007.

[25] K. Zhou and S. I. Roumeliotis. Optimal motion strategies for rangeonly constrained multisensor target tracking. *IEEE Transactions on Robotics*, 24(5):1168–1185, 2008.

[26] D. Uciński. *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, 2005.

[27] Fabio Morbidi and Gian Luca Mariottini. On Active Target Tracking and Cooperative Localization for Multiple Aerial Vehicles. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco - USA*, pages 2229–2234, 2011.

[28] Fabio Morbidi, Christopher Ray, and Gian Luca Mariottini. Cooperative active target tracking for heterogeneous robots with application to gait monitorin. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco - USA*, pages 3608–3613, 2011.

[29] Ke X. Zhou and Stergios I. Roumeliotis. Multi-robot Active Target Tracking with Distance and Bearing Observations. In *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, Saint Louis - USA*, pages 2209–2216, 2009.

[30] Ke Zhou and Stergios I. Roumeliotis. Multirobot Active Target Tracking With Combinations of Relative Observations. *IEEE TRANSACTIONS ON ROBOTICS*, 27(4):678–695, 2011.

[31] E. Bicho, S. Diegues, M. Carvalheira, A. Moreira, and S. Monteiro. Airship Formation Control. In *Proceedings of the 3rd Int. Conf. on Informatics in Control, Automation and Robotics, in Workshop Multi-Agent Robotic Systems (MARS 2006), Setúbal, Portugal*, pages 22–33, 2006.

[32] Richard M. Murray. Recent Research in Cooperative Control of Multivehicle Systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(3):571–583, 2007.

[33] S. Monteiro and E. Bicho. Robot formations: robots allocation and leader-follower pairs. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA2008),Pasadena, CA, USA*, pages 3769–3775, 2008.

[34] S. Monteiro and E. Bicho. Attractor dynamics approach to robot formations: theory and implementation. *Autonomous Robots*, 29(3):331–355, 2010.

[35] D. H. A. Maithripala, D. H. S. Maithripala, and S. Jayasuriya. A Geometric Approach to Dynamically Feasible, Real-Time Formation Control. *Journal of Dynamic Systems, Measurement, and Control*, 133(2):1–14, 2011.

[36] Huizhen Yang and Fumin Zhang. Robust Control of Formation Dynamics for Autonomous Underwater Vehicles in Horizontal Plane. *Journal of Dynamic Systems, Measurement, and Control*, 134(2):1–7, 2012.

[37] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated Multi-Robot Exploration. *IEEE Transactions on Robotics*, 21(3):376–386, 2005.

[38] Kiattisin Kanjanawanishkul. *Coordinated Path Following Control and Formation Control of Mobile Robots*. Phd, Eberhard-Karls-Universität Tübingen, 2010.

[39] Matthew J Daigle, Xenofon D Koutsoukos, and Gautam Biswas. Distributed Diagnosis in Formations of Mobile Robots. *IEEE Transactions on Robotics*, 23(2):353–369, 2007.

[40] Noritaka Sato, Kazuyuki Kon, Hiroaki Fukushima, and Fumitoshi Matsuno. Map-based Navigation Interface for Multiple Rescue Robots. In *Proceedings of the 2008 IEEE International Workshop on Safety, Security and Rescue Robotics, Sendai, Japan*, pages 152–157, 2008.

[41] Dongbing Gu. A Differential Game Approach to Formation Control. *IEEE Transactions on Control Systems Technology*, 16(1):85–93, 2008.

[42] Jing Guo, Gangfeng Yan, and Zhiyun Lin. Local control strategy for moving-target-enclosing under dynamically changing network topology. *Systems & Control Letters*, 59(10):654–661, 2010.

[43] Yingying Ding and Yan He. Flexible Leadership in Obstacle Environment. In *International Conference on Intelligent Control and Information Processing, China*, pages 788–791, 2010.

[44] Yangquan Chen and Zhongmin Wang. Formation Control : A Review and A New Consideration. In *IEEE/RSJ International Conference onIntelligent Robots and Systems, 2005. (IROS 2005). Vilamoura, Portugal*, pages 3181 – 3186, 2005.

[45] Paulo Tabuada, G.J. Pap, and Pedro Lima. Motion feasibility of multi-agent formations. *IEEE Transactions on Robotics*, 21(3):387–392, 2005.

[46] Wei Ren. Consensus strategies for cooperative control of vehicle formations. *IET Control Theory & Applications*, 1(2):505–512, 2007.

[47] Wei Zhang and Jianghai Hu. Optimal Multi-Agent Coordination Under Tree Formation Constraints. *IEEE Transactions on Automatic Control*, 53(3):692–705, 2008.

[48] Kar-Han Tan and M. A. Lewis. Virtual structures for high-precision cooperative mobile robotic control. In *IEEE International Conference Intelligent Robots and Systems, Osaka, Japan*, pages 132–139, 1996.

[49] Wenjie Dong. Cooperative Control of Multiple Nonholonomic Mobile Agents. *IEEE Transactions on Automatic Control*, 16(6):1434–1448, 2008.

[50] Dong Sun, Can Wang, Wen Shang, and Gang Feng. A Synchronization Approach to Trajectory Tracking of Multiple Mobile Robots While Maintaining Time-Varying Formations. *IEEE Transactions on Robotics*, 25(5):1074–1086, 2009.

[51] C. A. Kitts. Cluster Space Specification and Control of Mobile Multirobot Systems. *IEEE/ASME Transactions on Mechatronics*, 1(2):207–218, 2009.

[52] Vinicius T. L. Rampinelli, Alexandre S. Brand, Felipe N. Martins, and Ricardo Carelli. A Multi-Layer Control Scheme for Multi-Robot Formations with Obstacle Avoidance. In *Proceedings on the International Conference on Advanced Robotics. ICAR 2009, Bangkok, Thailand*, pages 1–6, 2009.

[53] Wei Ren. Consensus Tracking Under Directed Interaction Topologies: Algorithms and Experiments. *IEEE Transactions on Control Systems Technology*, 56(1):230–237, 2010.

[54] S. J. Yoo, J. B. Park, and Y. H. Choi. Adaptive formation tracking control of electrically driven multiple mobile robots. *IET Control Theory & Applications*, 4(8):1489–1500, 2010.

[55] Jawhar Ghommam, Hasan Mehrjerdi, Maarouf Saad, and Faiçal Mnif. Formation path following control of unicycle-type mobile robots. *Robotics and Autonomous Systems*, 58(5):727–736, 2010.

[56] Erick J. Rodriguez-Seda, James J. Troy, Charles A. Erignac, Paul Murray, Dušan M. Stipanovic, and Mark W. Spong. Bilateral Teleoperation of Multiple Mobile Agents: Coordinated Motion and Collision Avoidance. *IEEE Transactions on Control Systems Technology*, 18(4):984–992, 2010.

[57] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. The Entrapment-Escorting Mission. *IEEE Robotics & Automation Magazine*, 15(1):22–29, 2008.

[58] Brian Anderson, Changbin Yu, B.ÿ Fidan, and Julien Hendrickx. Rigid graph control architectures for autonomous formations. *IEEE Control Systems Magazine*, 28(6):48–63, 2008.

[59] João Sequeira and M. Isabel Ribeiro. Robot team control : A geometric approach. *Robotics and Autonomous Systems*, 53(1):59–71, 2005.

[60] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. Formation Control and Collision Avoidance for Multi-Agent Systems and a Connection between Formation Infeasibility and Flocking Behavior. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, Seville, Spain*, pages 84–89, 2005.

[61] Nelson André Saldanha Filipe. *Funcionalidades de Decisão Individual e Coordenação na Equipa CAMBADA*. Master, Universidade de Aveiro, 2008.

[62] Antidio Viguria and Ayanna M. Howard. An Integrated Approach for Achieving Multirobot Task Formations. *IEEE/ASME Transactions on Mechatronics*, 14(2):176–186, 2009.

[63] Anjan Kumar Ray, Patrick Benavidez, Laxmidhar Behera, and Mo M. Jamshidi. Decentralized Motion Coordination for a Formation of Rovers. *IEEE Systems Journal*, 3(3):369–381, 2009.

[64] Xiannuan Liang and Yang Xiao. Studying bio-inspired coalition formation of robots for detecting intrusions using game theory. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 40(3):683–693, 2010.

[65] Tamás Keviczky, Francesco Borrelli, Kingsley Fregere, Datta Godbole, and Gary J. Balas. Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations. *IEEE Transactions on Control Systems Technology*, 16(1):19–33, 2008.

[66] Wei Ni and Daizhan Cheng. Leader-following consensus of multi-agent systems under fixed and switching topologies. *Systems & Control Letters*, 59(3-4):209–217, 2010.

[67] Jiangyang Huang, S. M. Farritor, Ala Qadi, and Steve Goddard. Localization and follow-the-leader control of a heterogeneous group of mobile robots. *IEEE/ASME Transactions on Mechatronics*, 11(2):205–215, 2006.

[68] S. Carpin and L. E. Parker. Cooperative leader following in a distributed multi-robot system. In *Proceedings of IEEE lntemational Conference on Robotics and Automation, Washington, DC, USA*, pages 2994–3001, 2002.

[69] Liu Shi-cai, Tan Da-long, and Lui Guang-Jun. Robust Leader-follower Formation Control of Mobile Robots Based on a Second Order Kinematics Model. *Automatica*, 33(9):947–955, 2007.

[70] Luca Consolini, Fabio Morbidi, Domenico Prattichizzo, and Mario Tosques. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44(5):1343–1349, 2008.

[71] Tove Gustavi and Xiaoming Hu. Observer-Based Leader-Following Formation Control Using Onboard Sensor Information. *IEEE Transactions on Robotics*, 24(6):1457–1462, 2008.

[72] Michael Defoort, Thierry Floquet, Annemarie Kokosy, and Wilfrid Perruquetti. Sliding-Mode Formation Control for Cooperative Autonomous Mobile Robots. *IEEE Transactions on Industrial Electronics*, 55(11):3944–3953, 2008.

[73] Daisuke Kurabayashi, Tatsuki Choh, Jia Cheng, and Teteuro Funato. Adaptive Formation Transition among a Mobile Robot Group based on Phase Gradient. In *Proceedings of the 2008 IEEE International Conference on Robotics and Biomedics, Bangkok, Thailand*, pages 2001–2006, 2009.

[74] J. Guo, Z. Lin, M. Cao, and G. Yan. Adaptive control schemes for mobile robot formations with triangularised structures. *IET Control Theory & Applications*, 4(9):1817–1827, 2010.

[75] Geunho Lee and Nak Young Chong. Decentralized formation control for small-scale robot teams with anonymity. *Mechatronics*, 19(1):85–105, 2009.

[76] Q. Wang, M. Wu, Y. Huang, and L. Wang. Formation control of heterogeneous multi-robot systems. In *Proceedings of the 17th World Congress The International Federation of Automatic Control, Seoul, Korea*, pages 6596–6601, 2008.

[77] G. W. Gamage and G. Mann. Formation control of multiple nonholonomic mobile robots via dynamic feedback linearization. In *Robotics, 2009. ICAR 2009, Munich, Germany*, pages 1–6, 2009.

[78] Jiangping Hu and Gang Feng. Distributed tracking control of leader-follower multi-agent systems under noisy measurement. *Automatica*, 46(8):1382–1387, 2010.

[79] Veysel Gazi. Swarm aggregations using artificial potentials and sliding-mode control. *IEEE Transactions on Robotics*, 21(6):1208–1214, 2005.

[80] Shuzhi Sam Ge and Cheng-Heng Fua. Queues and artificial potential trenches for multirobot formations. *IEEE Transactions on Robotics*, 21(4):646–656, 2005.

[81] Tian-Tian Yang. Formation Control and Obstacle Avoidance for Multiple Mobile Robots. *Acta Automatica Sinica*, 34(5):588–592, 2008.

[82] Frank E Schneider and Dennis Wildermuth. Experimental Comparison of a Directed and a Non-Directed Potential Field Approach to Formation Navigation. In *Proceedings 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Espoo, Finland*, pages 21–26, 2005.

[83] G. Pereira, V. Kumar, and M. Campos. Closed loop motion planning of cooperating mobile robots using graph connectivity. *Robotics and Autonomous Systems*, 56(4):373–384, 2008.

[84] S. K. Yadlapalli, S. Darbha, and K. R. Rajagopal. Information Flow and Its Relation to Stability of the Motion of Vehicles in a Rigid Formation. *IEEE Transactions on Automatic Control*, 51(8):1315–1319, 2006.

[85] Travis Dierks and S. Jagannathan. Neural network control of mobile robot formations using RISE feedback. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 39(2):332–347, 2009.

[86] Chen Xin, Wu Min, and Li Yangmin. Formation Control Based on Adaptive NN with Time-Varying Interaction among Robots. In *Proceedings of the 27th Chinese Control Conference, Kunming, Yunnan, China*, pages 341–345, 2008.

[87] Travis Dierks and Sarangapani Jagannathan. Neural network output feedback control of robot formations. *IEEE Transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 40(2):383–399, 2010.

[88] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, London, England, 2004.

[89] Rolf Findeisen and Frank Allgöwer. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control, Veldhoven, The Netherlands*, pages 1–23, 2002.

[90] Dongbing Gu and Erfu Yang. A Suboptimal Model Predictive Formation Control. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, USA*, pages 3646–3651, 2005.

[91] William Dunbar and Richard Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[92] Dongbing Gu and Huosheng Hu. Receding horizon tracking control of wheeled mobile robots. *IEEE Transactions on Control Systems Technology*, 62(4):743–749, 2006.

[93] Feng Xie and Rafael Fierro. On Motion Coordination of Multiple Vehicles with Nonholonomic Constraints. In *2007 American Control Conference, New York - USA*, pages 1888–1893, 2007.

[94] Kiattisin Kanjanawanishkul and Andreas Zell. A model-predictive approach to formation control of omnidirectional mobile robots. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France*, pages 2771–2776, 2008.

[95] Kiattisin Kanjanawanishkul and Andreas Zell. Distributed model predictive control for coordinated path following control of omnidirectional mobile robots. In *2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore*, pages 3120–3125, 2008.

[96] Kiattisin Kanjanawanishkul. Formation Control of Omnidirectional Mobile Robots using Distributed Model Predictive Control. In *Proceedings of the 2nd International Conference on Robotic Communication and Coordination, Gent, Belgium*, pages 1–7, 2009.

[97] Massimo Vaccarini and Sauro Longhi. Formation Control of Marine Veihicles via Real-Time Networked Decentralized MPC. In *17th Mediterranean COnference on Control & Automation, Thessaloniki, Greece*, pages 428–433, 2009.

[98] D. M. Fontes, F. A. Fontes, and A. Caldeira. Model Predictive Control of Vehicle Formations. In *Optimization and Cooperative Control Strategies, M.J. Hirsch, C.W. Commander, P. Pardalos, and R. Murphey editors, Lecture Notes in Control and Information Sciences, Springer Verlag*, pages 371–384, 2009.

[99] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks, San Francisco, CA, USA*, pages 586–591, 1993.

[100] N. Trawny and T. Barfoot. Optimized motion strategies for cooperative localization of mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA - USA*, pages 1027–1032, 2004.

[101] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7):650–668, 1999.

[102] V. N. Christopoulos and S. I. Roumeliotis. Adaptive sensing for instantaneous gas release parameter estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain*, pages 4461–4467, 2005.

[103] A. I. Mourikis and S. I. Roumeliotis. Optimal sensor scheduling for resource-constrained localization of mobile robot formations. *IEEE Transactions on Robotics*, 22(5):917–931, 2006.

[104] E. W. Frew. *Trajectory design for target motion estimation using monocular vision*. Phd, Stanford University, Stanford - USA, 2003.

[105] A. W. Stroupe and T. Balch. Value-based action selection for observation with robot teams using probabilistic techniques. *Robotics and Autonomous Systems*, 50(2-3):85–97, 2005.

[106] R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press, London, England, 2004.

[107] Hugo S. Pinto Alves. *Navegação e Controlo de um Veículo Móvel*. Master, Porto University, 2010.

[108] J. Gonçalves, J. Lima, and P. Costa. Real-time localization of an omnidirectional mobile robot resorting to odometry and global vision data fusion: an EKF approach. In *IEEE International Symposium on Industrial Electronics, Cambridge, England*, pages 1275–1280, 2008.

[109] J. Gonçalves, J. Lima, and P. Costa. Real-time tracking of an omnidirectional robot - An Extended Kalman Filter approach and global vision data fusion: an EKF approach. In *International Conference on Informatics in Control, Automation and Robotics, Funchal, Portugal*, pages 5–10, 2008.

[110] Pedro Costa. *Planeamento Cooperativo de Tarefas e Trajectórias em Múltiplos Robôs*. Phd, Porto University, 2011.

[111] André Gustavo Scolari Conceição, António Paulo Gomes Mendes Moreira, and Paulo Costa. A Nonlinear Mobile Robot Modeling Applied to a Model Predictive Controller. In *Proceedings of the 2009 ACM symposium on Applied Computing, Honolulu, Hawaii - USA*, pages 1186–1187, 2009.

[112] H. P. Oliveira, A. J. Sousa, A. P. Moreira, and P. G. Costa. *Modeling and Assessing of Omni-directional Robots with Three and Four Wheels*. CONTEMPORARY ROBOTICS - Challenges and Solutions, A D Rodi(Ed.), INTECH, 2009.

[113] Tiago Pereira Nascimento, André Gustavo Scolari Conceição, and António Paulo Gomes Mendes Moreira. Omnidirectional Mobile Robot's Multivariable Trajectory Tracking Control: A Robustness Analysis. In *9th Portuguese Conference on Automatic Control, Coimbra, Portugal*, pages 63–68, 2010.

[114] João Rodrigo Ferreira and António Paulo Gomes Mendes Moreira. Non-linear Model Predictive Controller for Trajectory Tracking of an Omni-directional Robot Using a Simplified Model. In *9th Portuguese Conference on Automatic Control, Coimbra, Portugal*, pages 57–62, 2010.

[115] Luis Oliveira, Luis Almeida, and Frederico Santos. A Loose Synchronisation Protocol for Managing RF Ranging in Mobile Ad-Hoc Networks. In *RoboCup 2011: Robot Soccer World Cup XV, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Istanbul, Turkey*, pages 574–585, 2012.

[116] Paulo Costa and Paulo Malheiros José Gonçalves, José Limab. SimTwo Realistic Simulator: A Tool for the Development and Validation of Robot Software. *Theory and Applications of Mathematics and Computer Scien*, 1(1):17–33, 2011.

[117] Paulo J. Costa. Simtwo, 2010. Available from http://paginas.fe.up.pt/ paco/wiki/index.php?n= Main.SimTwo.

[118] Tiago Pereira Nascimento. *Controle de Trajetória de Robôs Móveis Omnidirecionais: Uma Abordagem Multivariável*. Master, Federal University of Bahia, 2009.

[119] Tiago Pereira Nascimento, Pedro Costa, Paulo Costa, André Gustavo Scolari Conceição, and António Paulo Gomes Mendes Moreira. Modeling Omnidirectional Mobile Robots: An Approach Using SimTwo. In *Proceedings of the 10th Portuguese Conference on Automatic Control, Funchal, Portugal*, pages 117–122, 2012.

[120] Maxon DC Motor. Datasheet re 40 d40 mm, graphite brushes, 150 watt, 2009. Page 82.

[121] Renato Miguel dos Santos Caldas. *Modelação e Simulação de um Robot Omnidireccional de 3 rodas*. Master, Porto University, 2009.

[122] Hélder P. Oliveira, Armando J. Sousa, A. Paulo Moreira, and Paulo J. Costa. Precise modeling of a four wheeled omni-directional robot. In *Robótica Scientific Meeting 2008, Aveiro, Portugal*, pages 57–62, 2008.

[123] T. P. Nascimento, M. A. Pinto, H. M. Sobreira, F. Guedes, A. Castro, P. Malheiros, A. Pinto, H. P. Alves, M. Ferreira, P. Costa, P. G. Costa, A. Souza, L. Almeida, L. P. Reis, and A. P. Moreira. 5dpo Robot Soccer Team Description Paper, 2011. Available from http://paginas.fe.up.pt/ robosoc/en/doku.php.

[124] Fumin Zhang and N.E. Leonard. Cooperative Filters and Control for Cooperative Exploration. *IEEE Transactions on Automatic Control*, 55(3):650–663, 2010.

[125] K. Kanjanawanishkul and A. Zell. Path following for an omnidirectional mobile robot based on model predictive control. In *IEEE International Conference on Robotics and Automation, ICRA 2009. Kobe, Japan*, pages 3341–3346, 2009.

[126] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[127] Fernando A. C. C. Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42(2):127–143, 2001.

[128] J. M. Maestre, D. M. de la Pena, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2):153–176, 2011.

[129] Aamir Ahmad and Pedro Lima. Multi-Robot Cooperative Object Tracking Based on Particle Filters. In *5th European Conference on Mobile Robots, Örebro, Sweden*, pages 1–6, 2011.

[130] André Gustavo Scolari Conceição. *Controlo e Cooperação de Robôs Móveis Autónomos Omnidireccionais*. Phd, Porto University, 2007.

[131] Kaustubh Pathak and Sunil K. Agrawal. An Integrated Path-Planning and Control Approach for Nonholonomic Unicycles Using Switched Local Potentials. *IEEE Transactions on Robotics*, 21(6):1201–1208, 2005.

[132] R. C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.

[133] Tiago Pereira Nascimento, André Gustavo Scolari Conceição, Fernando A. Fontes, and António Paulo Gomes Mendes Moreira. Leader Following Formation Control for Omnidirectional Mobile Robots: The Target Chasing Problem. In *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Noordwijkerhout, Netherlands*, pages 102–111, 2011.

[134] Tiago Pereira Nascimento, André Gustavo Scolari Conceição, and António Paulo Gomes Mendes Moreira. A Modified A* Application to a Highly Dynamic Unstructured Environment. In *Proceedings of the 11th International Conference on Mobile Robots and Competitions, Lisbon, Portugal*, pages 9–14, 2011.

[135] D. Alejo, R. Conde, J. A. Cobano, and A. Ollero. Multi-UAV collision avoidance with separation assurance under uncertainties. In *2009 IEEE International Conference on Mechatronics, Malaga, Spain*, pages 1–6, 2009.

[136] Pedro Costa, A Paulo Moreira, and Paulo Costa. Real-time path planning using a modified A* algorithm. In *ROBOTICA 2009 - 9th Conference on Mobile Robots and Competitions, Castelo Branco, Portugal*, pages 141–146, 2009.

[137] Khammapun Khantanapoka and Krisana Chinnasarn. Pathfinding of 2D & 3D game real-time strategy with depth direction A* algorithm for multi-layer. In *2009 Eighth International Symposium on Natural Language Processing, Seattle, WA - USA*, pages 1249–1254, 2009.

[138] S. X. Yang. Real-time torque control of nonholonomic mobile robots with obstacle avoidance. In *Proceedings of the IEEE Internatinal Symposium on Intelligent Control, Vancouver, Canada*, pages 81–86, 2002.

[139] Kenzo Kurihara, Nobuyuki Nishiuchi, Jun Hasegawa, and Kazuaki Masuda. Mobile Robots Path Planning Method with the Existence of Moving Obstacles. In *2005 IEEE Conference on Emerging Technologies and Factory Automation, Catania, Italy*, pages 195–202, 2005.

[140] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[141] Priyadarshi Bhattacharya and Marina Gavrilova. Roadmap-Based Path Planning - Using the Voronoi Diagram for a Clearance-Based Shortest Path. *IEEE Robotics & Automation Magazine*, 15(2):58–66, 2008.

[142] Mike Peasgood, Christopher Michael Clark, and John McPhee. A Complete and Scalable Strategy for Coordinating Multiple Robots Within Roadmaps. *IEEE Transactions on Robotics*, 24(2):283–292, 2008.

[143] Junfeng Yao, Chao Lin, Xiaobiao Xie, Andy JuAn Wang, and Chih-Cheng Hung. Path Planning for Virtual Human Motion Using Improved A* Star Algorithm. In *2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV - USA*, pages 1154–1158, 2010.

[144] Chenghui Cai and Silvia Ferrari. Information-driven sensor path planning by approximate cell decomposition. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 39(3):672–689, 2009.

[145] Dave Ferguson, Maxim Likhachev, and Anthony Stentz. A Guide to Heuristic-based Path Planning. In *Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems. International Conference on Automated Planning and Scheduling (ICAPS), Monterey, CA, U.S.A*, pages 1–10, 2005.

[146] J. R. Bruce and M. Veloso. Safe Multirobot Navigation Within Dynamics Constraints. *Proceedings of the IEEE*, 94(7):1398–1411, 2006.

[147] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *Int. Journal of Robotics Research*, 17(7):760–772, 1998.

[148] D. Wilkie, J. Berg, and D. Manocha. Generalized velocity obstacles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, New York, N.Y., USA*, pages 5573–5578, 2009.

[149] Samir Shaker, Jean J. Saade, and Daniel Asmar. Fuzzy Inference-Based Person-Following Robot. *International Journal of Systems Applications, Engineering and Development*, 2(1):29–34, 2008.

[150] Hani Hagras, Victor Callaghan, and Martin Colley. Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online Fuzzy Genetic system. *Fuzzy Sets and Systems*, 141(1):107–160, 2004.

[151] Jemej Virant and Niko Zimic. Fuzzy Automata with Fuzzy Relief. *IEEE Trasactions on Fuzzy Systems*, 3(1):69–74, 1995.

[152] Andrea Bonarini, Giovanni Invernizzi, Thomas Halva Labella, and Matteo Matteucci. An architecture to coordinate fuzzy behaviours to control an autonomous robot. *Fuzzy Sets and Systems*, 134(1):101–115, 2003.

[153] C. McMillen and Manuela Veloso. *Distributed, Play-Based Coordination for Robot Teams in Dynamic Environments*, volume 4434. Gerhard Lakemeyer, Elizabeth Sklar, Domenico Sorrenti and Tomoichi Takahashi, RoboCup 2006, Robot Soccer World Cup X, Springer LNCS, Berlin, 2007.

[154] Nuno Lau and Luis Paulo Reis. *FC Portugal - High-level Coordination Methodologies in Soccer Robotics*. Robotic Soccer, Book edited by Pedro Lima, Itech Education and Publishing, Vienna, Austria, 2007.

[155] Nuno Lau, Luís S. Lopes, Nelson Filipe, and Gustavo A. Corrente. Roles, Positionings and Set Plays to Coordinate a RoboCup MSL Team. In *Proceedings of EPIA, Aveiro, Portugal*, pages 323–337, 2009.

[156] Vadim Kyrylov and Serguei Razykov. *Pareto-Optimal Offensive Positioning in Simulated Soccer*, volume 5001. U. Visserm, F. Ribeiro, T. Ohashi and F. Dellaert (eds), RoboCup 2007: Robot Soccer World Cup XI, Springer LNAI, Berlin, 2008.

[157] J. L. Grantner and G. A. Fodor. Takagi-Sugeno type fuzzy automaton model. In *IEEE International Conference on Fuzzy Systems, Barcelona, Spain*, pages 1–6, 2010.

[158] Robocup 2011. Robocup 2011, istanbul, turkey, 2011. Available from http://www.robocup2011.org/en/.

[159] H. T. Dashti, N. Aghaeepour, S. Asadi, M. Bastani, Z. Delafkar, F. M. Disfani, S. M. Ghaderi, S. Kamali, S. Pashami, and A. F. Siahpirani. *Dynamic Positioning based on Voronoi Cells (DPVC)*, volume 4020. Itsuki Noda, Adam Jaco, Ansgar Bredenfeld and Yasutake Takahashi, RoboCup 2005, Robot Soccer World Cup IX, Springer LNAI, Berlin, 2006.

[160] Ryota Nakanishi, Kazuhito Murakami, and Tadashi Naruse. *Dynamic Positioning Method Based on Dominant Region Diagram to Realize Successful Cooperative Play*, volume 5001. U. Visserm, F. Ribeiro, T. Ohashi and F. Dellaert (eds), RoboCup 2007: Robot Soccer World Cup XI, Springer LNAI, Berlin, 2008.

[161] H. Akiyama and I. Noda. *Multi-Agent Positioning Mechanism in the Dynamic Environment*, volume 5001. U. Visserm, F. Ribeiro, T. Ohashi and F. Dellaert (eds), RoboCup 2007: Robot Soccer World Cup XI, Springer LNAI, Berlin, 2008.

[162] Peter Stone and Manuela Veloso. Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. *Artificial Intelligence*, 110(2):241–273, 1999.

[163] Luis Paulo Reis, Nuno Lau, and Eugénio C. Oliveira. *Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents*, volume 2103. Springer LNAI, Berlinm 2001, 2001.

[164] Jawhar Ghommam, Hasan Mehrjerdi, and Maarouf Saad. Leader-Follower Formation Control of Nonholonomic Robots with Fuzzy Logic Based Approach for Obstacle Avoidance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA*, pages 2340–2345, 2011.

[165] Kehua Miao and MaoQing Li. Role Assignment Strategy for Heterogeneous Robots Based on Fuzzy Evaluation And Fuzzy Inference. In *9th International Conference on Control, Automation, Robotics and Vision, Singapore*, pages 1–5, 2006.

[166] Marianne Sisto and Dongbing Gu. A Fuzzy Leader-Follower Approach to Formation Control of Multiple Mobile Robots. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Shanghai, China*, pages 2515–2520, 2006.

[167] S. Hamidreza Mohades Kasaei, S. Mohammadreza Mohades Kasaei, S. Alireza Mohades Kasaei, and Mohsen Taheri. Design of an Action Selection Mechanism for Cooperative Soccer Robots Based on Fuzzy Decision

Making Algorithm. *Broad Research in Artificial Intelligence and Neuroscience*, 1(3):5–18, 2010.

[168] Can Kavaklioglu and Okyay Kaynak. A Type-2 Fuzzy Planner with Semi Qualitative World Model for Robocup Domain. In *Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering, Scottsdale, AZ, USA*, pages 795–799, 2007.

[169] Yu-Cheng Chen and Yin-Tien Wang. Obstacle Avoidance and Role Assignment Algorithms for Robot Formation Control. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA*, pages 4201–4206, 2007.

[170] Yanmin Lei, Qidan Zhu, Xin Leng, Zhibin Feng, and Jihong Song. The Application of Fuzzy Neural Networks in Formation Control for Multi-robot System. In *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, China*, pages 69–73, 2008.

[171] A. P. Gerdelan. *Artificial Intelligence in Robot Soccer*. Bachelor, Massey University, 2006.

[172] D. Strnad and N. Guid. A fuzzy-genetic decision support system for project team formation. *Applied Soft Computing Journal*, 10(4):1178–1187, 2020.

[173] K. N. Maleki, M. H. Valipour, R. Y. Ashrafi, S. Mokari, M. R. Jamali, and C. Lucas. A simple method for decision making in RoboCup soccer simulation 3D environment. *Revista Avances en Sistemas e Informática*, 5(3):109–116, 2008.

[174] Armando Jorge Miranda Sousa. *Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através de Visão*. Phd, Porto University, 2004.

[175] Armando Sousa, Paulo Costa, and António Moreira. Vision Based Real-Time Localization Of Multiple Mobile Robots. In *3rd Int. Conf. on Field and Service Robotics, Helsinki, Finland*, pages 103–106, 2001.