FORMATION CONTROL OF MULTIPLE MOBILE ROBOTS USING PARAMETRIC AND IMPLICIT REPRESENTATIONS

by

Yeşim Hümay Esin

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabanci University
Spring 2008

APPROVED BY

Assoc. Prof. Dr. Mustafa Ünel .....
(Thesis Supervisor)

Assist. Prof. Dr. Kemalettin Erbatur .....
(Thesis Co-Supervisor)

Assist. Prof. Dr. Ahmet Onat .....

Assist. Prof. Dr. Volkan Patoğlu .....

Assist. Prof. Dr. Mehmet Yıldız .....

DATE OF APPROVAL: .....

to my parents

Acknowledgments

I would like to express my deepest gratitude to my thesis advisor Assoc. Prof. Dr. Mustafa Ünel for his valuable support, guidance and giving me the chance to do this interesting work.

I am grateful to my co-advisor Asist. Prof. Dr. Kemalettin Erbatur for his support and understanding. I also would like to thank Ahmet Onat, who has always been enthusiastic to answer my questions, Volkan Patoğlu and Mehmet Yıldız for spending their valuable time as my jurors.

I would like to express my thanks to Berk Çallı for his endless understanding and support for many years. I would like to thank Lale Tunçyürek for always being with me and helping me in my most miserable times. I would like to thank all my friends from Mechatronics graduate program for their valuable friendship. I send my many thanks to all my precious friends who were always next to me. I also thank anyone who contributed in anyway to this thesis.

Finally, I am deeply grateful to my beloved parents, Tülay Esin and Emin Murat Esin for raising me with great love. Their priceless trust in me and their support in every instant of my life cannot be compared to anything else.

FORMATION CONTROL OF MULTIPLE MOBILE ROBOTS USING PARAMETRIC AND IMPLICIT REPRESENTATIONS

Yeşim Hümay Esin
Mechatronics, MS Thesis, 2008
Thesis Supervisor: Assoc. Prof. Dr. Mustafa ÜNEL
Thesis Co-Supervisor: Assist. Prof. Dr. Kemalettin ERBATUR
Keywords: Formation Control, Robot Coordination, Autonomous Mobile Robots

Abstract

Coordination of autonomous robot groups is an active research area and much recent work has focused on modeling and control issues related to coordination. Robot groups can coordinate in many different ways. Some robot groups may execute coordination in which group members move in a scattered manner like the bees of a beehive or coordination of the group may require a more strict formation like the swallows. The shape formation is very important for the coordination of autonomous mobile robot groups because it increases the capability of a robot group by increasing the competence and the security of the group. The shape formation is applicable in many tasks like formation flight, flocking and schooling, transportation systems, search-and-rescue operations, competitive games, reconnaissance and surveillance.

This thesis develops a flexible shape formation control method for autonomous mobile robots. There are different approaches in the literature for shape formation of mobile robots. Proposed method is different from these

existing approaches by being applicable to complex formation curves as well as different number of robots and heterogeneous groups. It consists of two phases. In the first phase, shape formation is controlled by using potential fields generated from implicit polynomial representations and in the second phase, the control for keeping the desired shape is designed using elliptical Fourier descriptors. In this shape formation method, coordination between the robots is modeled using virtual linear springs between each robot and its nearest two neighbors. The success of the proposed method is shown through simulations on groups of different numbers of point-particle robots. Proposed method is then extended to non-holonomic mobile robots by using the desired positions in point particle model as references for the non-holonomic robots. The method is also implemented with real non-holonomic robots with a bird-eye-view camera.

MOBİL ROBOTLARIN ELİPTİK FOURIER TANIMLAYICILAR VE ÖRTÜK POLİNOMLAR KULLANILARAK FORMASYON KONTROLÜ

Yeşim Hümay Esin
Mekatronik Mühendisliği, Yüksek Lisans Tezi, 2008
Tez Danışmanı: Doç. Dr. Mustafa ÜNEL
Tez Eşdanışmanı: Yrd. Doç. Dr. Kemalettin ERBATUR
Anahtar Kelimeler: Formasyon Kontrolü, Robot Koordinasyonu, Otonom Mobil Robotlar

Ozet

Otonom robotların koordinasyonu aktif bir araştırma konusudur. Son yıllarda bu alanda modelleme ve kontrol konularında bir çok araştırma yapılmaktadır. Robot gruplarının koordinasyonu bir çok farklı şekilde sağlanabilir. Bazı robot gruplarında arılar gibi dağınık bir formasyonda koordinasyon sağlanırken bazı gruplarda kırlangıçlar gibi daha katı formasyonlar oluşturulabilir. Koordineli hareket eden robot gruplarının belli bir formasyon halinde iş yapmaları bu robot grubunun iş yapma kapasitesini arttırır. Ayrıca birçok koordinasyon görevi, belli bir formasyonda hareket etmeyi gerektirir. Şekil formasyonu, arama ve kurtarma, gözetleme, büyük cisimlerin taşınması, bir grup hava aracının düzenli uçuş yapması gibi bir çok alanda kullanılabilir.

Bu tezde, otonom mobil robotların formasyonu için esnek bir yöntem sunulmaktadır. Literatürde bu konuda bir çok yaklaşım bulunmaktadır. Bu tezde geliştirilen yöntem, karmaşık formasyon şekillerine ve çeşitli büyüklükteki robot gruplarına uygulanabilir olmasıyla diğer yöntemlerden ayrılır. Su-

nulan formasyon yöntemi iki aşamadan oluşmaktadır. Birinci aşamada formasyon, örtük polinom tanımı kullanılarak oluşturulan potansiyel alanlarla sağlanır. İkinci aşamada oluşturulan şeklin korunumu için eliptik Fourier betimleyiciler kullanılmıştır. Bu formasyon kontrolünde, robotlar arasındaki koordinasyon her bir robot ve en yakın iki komşusu arasındaki lineer yaylarla modellenmiştir. Sunulan yöntemin başarısı çeşitli büyüklükteki robot gruplarıyla ve farklı formasyon şekilleriyle yapılan benzetimlerle gösterilmiştir. Bu yöntem daha sonra holonomik olmayan robotlar için de genişletilmiştir. Yöntem ayrıca holonomik olmayan gerçek robotlar üzerinde uygulanmıştır.

Table of Contents

Acknowledgments **v**

Abstract **vii**

Ozet **ix**

1 Introduction **1**

2 Literature Survey **6**
2.1 Shape Formation Control in Mobile Robots 7
2.2 Sensors in Shape Formation Control 13
2.2.1 Ultrasonic Range Sensors 14
2.2.2 Vision in Mobile Robotics 15

3 Representation of Closed Curves **19**
3.1 Elliptic Fourier Descriptors (EFD) 19
3.2 Implicit Polynomial Representation 23
3.3 Examples for Closed-Bounded Curve Representations 27
3.3.1 Representation of a Quadrangle Using 3 Harmonics . . 27
3.3.2 Representation of a Star-Like Shape Using 6 Harmonics 28

4 Control of Non-Holonomic Mobile Robots **31**
4.1 Modeling of Non-Holonomic Robots 32
4.2 Trajectory Control of Non-Holonomic Mobile Robots 34

5 Shape Formation Control of Mobile Robots **37**
5.1 Coordination Control 39

5.2 Formation Control Using Implicit Polynomial Potential Functions . . . 42
5.3 Keeping Formation Using Elliptic Fourier Descriptors . . . 44
5.4 Shape Formation Control of Nonholonomic Robots . . . 47

6 Simulation Results and Discussions **49**
6.1 Simulations with Point Particle Model . . . 50
6.1.1 Simulation Results for a Single Robot . . . 50
6.1.2 Simulation Results for 5 Robots . . . 51
6.1.3 Simulation Results for 6 Robots . . . 53
6.2 Simulations with Non-Holonomic Model . . . 55
6.2.1 Simulation Results for a Single Robot . . . 56
6.2.2 Simulation Results for 5 Robots . . . 57
6.2.3 Simulation Results for 6 Robots . . . 57

7 Experimental Results **60**
7.1 Assumptions . . . 60
7.2 The System . . . 61
7.2.1 Our Robots . . . 61
7.2.2 Vision System . . . 63
7.3 Experiments . . . 65
7.3.1 Experiment with One Single Robot . . . 65
7.3.2 Experiment with a Robot Group . . . 66

8 Conclusion **68**

List of Figures

2.1 An example of ultrasonic sensor usage in formation control . . 15
2.2 An example of robots used in vision-based formation control . 17

3.1 Basic idea of EFD 20
3.2 Closed curve constructed from three component ellipses 22
3.3 Representation of a quadrangle by EFD and Implicit polynomial 29
3.4 Representation of a star-like shape by EFD and Implicit polynomial 30

4.1 A Unicycle robot 33

5.1 Modeling of coordination control 40
5.2 Shape formation control for nonholonomic robot 47

6.1 Route of a point particle mobile robot 51
6.2 Desired formation (ellipse) with 5 point particle robots 52
6.3 Desired formation (star shape) with 6 point particle robots . . 54
6.4 Route of non-holonomic mobile robot 56
6.5 Desired formation (ellipse) with 5 non-holonomic robots 58
6.6 Desired formation (star shape) with 6 non-holonomic robots . 59

7.1 The robot used in the shape formation implementations 61
7.2 The powerful PC on implementation robots 62
7.3 RS-232 connection between PC and the processor of Trilobot . 65
7.4 Implementation of shape formation with a single robot 66
7.5 Implementation of shape formation with 3 robots 67

Chapter 1

Introduction

The role of autonomous robots in our lives is increasing in many fields. The robots are desired in many tasks for their high speed, precision and repeatability. The robots are also being employed in the areas which are hazardous, dangerous or boring for humans. The working areas of robots is enlarging from idealized areas, like industrial plants, to work in natural environments or to serve humans in their complicate homes. New working areas bring new problems for researchers. By the increasing demands for robots in different areas, the robots need to be more adaptive to changing or unknown environmental conditions in the workplace and they should be more intelligent to be able to make their own decisions in these conditions.

Robots can adapt to complex environments and perform tasks more intelligently by working in groups. Robot groups may be composed of many different kinds of robots like ground vehicles, aerial vehicles, underwater vehicles or spacecrafts. A robot group may be homogenous; each member in the group may be identical, or it can be heterogeneous; the group may include

different kinds of robots. Using a team of simple robots is advantageous than using a single but more complicated robot in many ways. Robot's working in groups brings flexibility in a given task. If the robots of a group is doing a task together, the robots can learn about the environmental conditions more quickly by gathering sensor information from a variety of sensors of each member. Besides, if one of the robots gets hurt during the task, the remaining ones can finish the task. This makes the robot group systems more fault tolerant than single robot systems. Since using a group of robots brings the possibility of parallel processing, the time required for the completion of the task decreases, especially when it is a distributed task, like search and rescue or mapping of unknown areas.

Robot groups can coordinate in many ways. Some robot groups may execute coordination in which the robots move in a scattered manner like the bees of a beehive or the control of the robot group may require a more strict formation like the swallows. The shape formation is very important for coordination of mobile robot groups because it increases the capability of a robot group by increasing the competence and the security of the group. The shape formation is applicable in many tasks like formation flight, flocking and schooling, transportation systems, search-and-rescue operations, competitive games, reconnaissance and surveillance.

The shape formation in mobile robots is a challenging topic and there are many researches on that subject, as it will be mentioned in detail in Chaper 2. For robot groups coordinating with shape formation, the flexibility of the shape formation is very important. With the increasing demand for autonomous robots in different fields, many different kinds of formation

shapes are required. In non-idealized environments, forming many of the simple shapes may not be feasible. Besides, many different task definitions may require very complicated formation shapes. Another important issue of shape formation is the fault-tolerance. The shape formation algorithm should guarantee the completion of the task even if some of the group members are hurt. Since different tasks require different types of robot groups, a formation shape algorithm should also be flexible in the number and the heterogeneity of the team members.

Control of a robot group can be centralized or decentralized. In the centralized control, the data is collected in a central control unit and the control commands are sent from that unit to the robots. This central unit can be an independent computer or can be one of the members of the robot group which has a higher computational capacity. The central control unit receives a collection of the data from the robot group and the decision for each member is done according to this knowledge.

In the decentralized control, each member in the robot group gathers data using its own sensors and decides about its move according to its role definition in the desired task. In some cases, there are also some local communications among the group members.

In decentralized control, the members have a local sense of the group because the knowledge is limited by the sensor angle and occlusions. On the other hand, since in the centralized control all the data are collected by the central unit, the effects of the view angle limitation and the occlusions can be compensated. The central unit has an overall view of the robot group condition. This leads to a better decision. In the central control, complete

solution and global optimum is more likely to be achieved.

One of the limitations of the centralized control is the communication. In the centralized control, the moves of agents in the group are decided by the central unit and these commands are sent to each agent. As the number of the agents increases, the communication load of the central unit increases. This can be seen as a bottleneck for centralized control but there are studies which solves this problem by decreasing the communication load on the central unit.

In robot coordination, the robustness of the algorithm to robot failures is very important. In centralized control, the detection of agent failure is available. In such a case, the central unit can decide for a better strategy of the robot group for the task to be executed in the best way available. On the other hand, in centralized controls, the failure of the central unit is a major problem to cause task failure.

This thesis provides a new method for a shape formation method which brings flexibility to formation shape and is applicable to groups of different sizes and heterogeneous systems. The achievement of the proposed method is based on the flexible representation of the desired formation curves using implicit polynomials and elliptic Fourier descriptors. The success of the method for point particle and non-holonomic mobile robot models is demonstrated through simulations. The method is also implemented using real non-holonomic robots. For the implementations, a centralized control is prefered by considering the requirements of the task.

Chapter 2 gives a brief survey on formation control. Representation of complex closed curves is presented in Chapter 3. Chapter 4 is on the mod-

eling and trajectory tracking control for non-holonomic mobile robots. The proposed formation method is presented in Chapter 5. The simulation and implementation results are given in Chapter 6 and Chapter 7, respectively. Finally Chapter 8 concludes the thesis and indicates possible future directions.

Chapter 2

Literature Survey

In the recent years the coordination of multi-robot systems has been subjected to considerable research efforts. The main motivation is that in many tasks a group of robot can perform more efficiency than a single one and can accomplish tasks not executable by a single robot. Multi-robot systems have advantages like increasing tolerance to possible vehicle fault, providing flexibility to the task execution or taking advantage of distributed sensing and actuation [10]. Each animal in a herd, for instance, benefits by minimizing its encounters with predators [50]. Arkin and Balch [39] argued that two or more robots can be better than one for several reasons:

- Many robots can be in many places at the same time (distributed action).
- Many robots can do many, perhaps different things at the same time (inherent parallelism).
- Often each agent in a team of robots can be simpler than a more com-

prehensive single robot solution (simpler is better).

Among the tasks that are done with a robot group, operating in a special formation increases the capability of the robot team in many ways. Shape formation during the operation of a task enhances the system performance by increasing instrument resolution and cost reduction. In [11], it is stated that global security and efficiency of the team can be enhanced by a proper configuration for the formation. Formations allow individual team members to concentrate their sensors across a portion of the environment while their partners cover the rest. In [6], it is stated that air force fighter pilots for instance direct their visual and radar search responsibilities depending on their position in a formation.

Formation in a proper configuration is one of the ways to get the maximum efficiency from a robot team. There are many tasks that the shape formation of autonomous robots can be used. Examples in the literature include box pushing [26], load transportation [19] , dispersing a swarm [47] [2],moving in formation [6], covering areas while maintaining constraints [34], perform shepherding behaviors [34] and enclosing an invader [54].

2.1 Shape Formation Control in Mobile Robots

Shape formation of multiple mobile robots is a challenging subject. This subject includes many sub-problems like decision of the feasible formation shape, getting into formation, maintenance of the formation shape and switching between the formations.

Shape formation and maintenance of the formation is one of the important

problems in the shape formation on which much research has been done. There are many different approaches to modeling and solving these problems, ranging from paradigms based on combining reactive behaviors [4], to those based on leader-follower graphs [17] and potential field methods [44].

One of the common methods is to determine the desired position of each member within the group to control each robot to these specified positions. This methods works fine when the number of the group is small. When the number of robots increases, it becomes difficult and inefficient to manually determine the position of each and every agent within the formation.

There are some approaches for formation control which are inspired by biological systems. Biologists who study animal aggregations such as swarms, flocks, schools, and herds have observed the individual-level behaviors which produce the group-level behaviors [30] [33]. In some studies this observation are applied on robot groups and the animal behaviors are mimicked by the robots.

One of the well-known applications in this field is by Reynolds [37]. He developed simple egocentric behavior model for the individuals of the simulated group of birds or so-called "Boids" . In this model, the basic flocking model consists of three simple steering behaviors which describe how an individual Boid maneuvers based on the positions and velocities its nearby flockmates. First behavior is separation which is steering to avoid crowding local flockmates. The other behavior is alignment which is steering towards the average heading of local flockmates and the last is cohesion; steering to move toward the average position of local flockmates. Reynolds showed that Boids behave just like real birds.

Vicsek et al. reported the group behavior of real bacteria by simple model [48]. The simple "nearest neighbors" method is proposed in order to investigate the emergence of autonomous motions in systems of particles with biologically motivated interaction. In this method, particles are driven with a constant absolute velocity and they choose the average direction of motion of the particles in their neighborhood with some random perturbation added. The developed model showed a good approximation to the motion of bacteria that exhibit coordination motion in order to survive under unfavorable conditions. This idea has then been widely used in the literature to attack the problem of modeling the coordinated motion of a group of autonomous mobile robots [53], [3], [28], [49].

Leader follower method is one of the most common approaches for formation control. In the leader following method one or more robots are assigned as leaders and responsible for guiding the formation. The other robots are required to follow the leader according to predefined behaviors. Examples include papers by Wang [52], presented some simple strategies for a fleet of autonomous robots to navigate in formation and studied the interaction dynamics of these robots with the presented navigation strategies. In this study, several strategies which are based on leader following and neighbor following are presented. The presented strategies include "Nearest-Neighbor Tracking" in which each robot is assigned to maintain its desired position according to its nearest neighbor. Another method presented is "Multi-Neighbor Tracking" in which several robots are assigned as leaders or the guardians of the fleet. [35] and [16] are some more recent examples of the formation control using the leader-follower strategy.

Behavior based approach is used in many studies for shape formation. In this approach, shape formation of the whole group is achieved through of the individual agents by using the weighted sum of some basic and intuitive behaviors. We can see successful applications of this idea in the subsumption architecture [27], [24], [7].

Balch and Arkin presented a behavior-based approach to robot formation keeping [5]. In this study, new reactive behaviors for implementing formations in robot groups are presented and evaluated. In this study, several motor schemas, move-to-goal, avoid-static-obstacle, avoid-robot and maintain-formation are introduced. Each schema represents a vector representing the desired behavioral response to the current situation of the robot and the group. A gain value is indicated representing the importance of individual behaviors. The high-level combined behavior is generated by multiplying the outputs of each primitive behavior by its gain, summing and normalizing the result. This method makes the robot group to be able to move to a goal location while keeping in formation, avoiding obstacles and collision with other robots. In [6], this approached is extended by an additional motor schema which is based on a potential field method.

In [10], a novel behavior based approach is introduced for a platoon of mobile robots to shape formation while avoiding collision with themselves and external obstacles. It uses a hierarchy-based approach so called Null-Space-based Behavioral (NSB) control. This control uses the null-space projection to obtain the final motion command from outputs of multiple conflicting tasks.

Potential function approaches to robot navigation provide an elegant

paradigm for expressing multiple constraints and goals in mobile robot navigation problems [21]. One of the first work applying artificial potentials to agent coordination is [36]. In this approach a distributed control for very large scale robotic (VLSR) systems is presented. Simple artificial force laws between pairs of robots or robot groups are introduced. This force laws are inverse-power force laws which incorporates both attraction and repulsion. These forces are used to reflect "social relations" among robots to a degree and therefore this method is called "Social Potential Fields". In this method, each robot senses the resultant potential from components like other robots, obstacles, objectives etc. and acts under the resultant force. In this approach the parameters can be chosen arbitrarily to reflect the relationship between the robots whether they should stay close together or far apart to form the desired formation shape.

Yamaguchi and Arai [55] define a potential field on the space according to the relative distances between neighbors. In this study, the shape-generation problem is approached using systems of linear equations. Each robot, starting at some initial location, changes its position according to a linear function of its neighbors' positions and some fixed constant. Simulations of the method show that a group of initially collinear robots will converge into the shape of an arc.

Song and Kumar [44] introduced a framework for control a group of robots for cooperative manipulation task. In this framework, the trajectory generation problem for cooperative manipulation task is addressed. This framework allows the robots to approach the target object, organize themselves into a formation that will trap the object and then transport the

object to the desired destination. The robots in the group can also avoid static obstacles. The controllers are derived from simple potential fields and the hierarchical composition of the potential fields.

In [6], an approach which is inspired by the way molecules "snap" into place as they form crystals; the robots are drawn to particular "attachment sites" positioned with respect to other robots. Using this approach, a new class of potential functions is developed for shape formation control of multiple robots homogeneous largescale robot teams while navigating to a goal location through an obstacle field.

In [23] a shape formation method is presented for a heterogeneous robot group. In this method, the robots are controlled to reach the goals while controlling the group geometry, individual member spacing and obstacle avoidance is managed. Bivariate normal probability density functions (pdfs) are presented to construct the surface which swarm members move on to generate potential fields. Limiting functions are also introduced to provide tighter swarm control by modifying and adjusting a set of control variables forcing the swarm according to set constraints. In this method, the swarm member orientation and the swarm movement as a whole is controlled by the combination of limiting functions and bivariate normal functions.

In [25], the potential field approach is combined with virtual leaders proposed in [20]. A virtual leader is a moving reference point that affects the robots in the group by means of artificial potentials. Virtual leaders are used to maintain group geometry and direct the motion of the group. In this approach, the potential produced from functions of relative distance between a pair of neighbors. The control force for an individual is derived as the minus

gradient of the sum of all potentials affecting that individual. This leads the individuals are driven to the minimum of the total potential. The desired group is achieved by designing local potentials with some predescribed inter-vehicle spacing associated with virtual leaders which are moving reference points.

In this study, a novel potential function approach for shape formation of autonomous robot groups is developed. In this method implicit polynomial representations and elliptic Fourier descriptors are used for describing the formation shape. The implicit polinomial representation is used for producing potential fields to make the robots reach to the desired formation curve. When the robots reach the formation curve, the elliptic Fourier representation of the curve is used to define a trajectory for each robot to make the group travel around the curve. For coordination of the robot group, a coordination control component is applied with the shape formation control component. The coordination component is modeled by linear springs between each robot and its nearest two neighbours. In this method, the implicit polynomials and elliptic Fourier descriptors introduces flexibility for the formation shape. The method is applicable for heteregenous groups with different number of robots.

2.2 Sensors in Shape Formation Control

The formation control of a robot team needs a good pose estimate of the robot members, obstacles and targets. That is why; the sensor choice has an important role for the success of the formation control implementation.

Although numerous types of sensors exist in the market, in the formation control of robot groups mainly ultrasonic range sensors and vision sensors are used.

2.2.1 Ultrasonic Range Sensors

Ultrasonic rage sensors on robots are used for having distance measurement of other robots and obstacles around. These sensors are practical because valuable data is gathered with low computational costs. They also are robust against changes in environmental factors such as temperature, color, etc. compared to other sensing methods.

In formation control implementations, ultrasonic sensors are mostly used as ultrasonic sensor rings attached to the robot base. Sensor rings helps the robots to get multiple and more correct measurements and increases the sensing range of the robot. This provides better tracking of changes around the robot such as the movement of the robots and obstacles in a wider range. In [4], ultrasonic range sensors are used this way. Two autonomous robots equipped with 16 range sensors are used for hazard detection in formation control implementations. The robots can be seen in Figure 2.1. In this study, the experiments are run in a test area measuring approximately $10m \times 5m$. The robots are dictated to travel in formation in two environmental conditions; with and without obstacles. The robots estimate their positions using shaft encoders and each robot reports its position to the other one using wireless communication.

Figure 2.1: An example of ultrasonic sensor usage in formation control

2.2.2 Vision in Mobile Robotics

The pose estimate of the robot members, obstacles and targets is important in formation control of a robot team. The main advantage of the cameras in these applications is the richness of the provided data. Since the cameras are getting cheaper and reaching higher data speeds, they are becoming more advantageous. Processing the camera data may cost more than the other sensors but the recent developments in the processor technology increases the processing speed, decreasing the size and the costs of the processors which make vision-based systems become more available for mobile robots.

Although cameras provide rich data, there are some limitations on this data because of occlusion and view angle. The camera data cannot give information about the environment which is beyond its view-angle or is occluded by an obstacle, another robot etc. Omnidirectional cameras can be used for

solving the view angle limitation of the cameras. These cameras are widely used in shape formation implementations. In [8], a cooperation strategy based on omnidirectional vision is presented. This strategy is designed to be applied on a heterogeneous robot group which is formed by small and simple robots and a bigger leader robot with high computational power. In this group, the leader robot has an omnidirectional camera and can see the small robots. The formation strategy is based on non-linear control techniques and the stability is proven using Lyapunov method. This is a centralized control method that the leader robot uses its omnidirectional camera system to find the positions of smaller robots and sends commands to them. The smaller ones have their own controllers to maintain the commanded linear and angular velocities. Each follower robot is identified by a colorful rectangle on its platform. The poses of the robots are estimated through color segmentation and Kalman filtering.

The best information about the condition of the whole robot group, the obstacles and the targets can be achieved by sharing the information of the data from cameras with different positions and orientations if multiple cameras are available. In [1], omnidirectional cameras are used as vision sensors. Estimators that abstract sensory information at different levels enabling a decentralized control are introduced. Some logical sensors using omnidirectional images are developed. These sensors are an obstacle detector, a collision detector, a decentralized state observer, and a centralized state observer. The obstacle and collision detectors rely on edges on the images and the remaining sensors uses color segmentation. Each robot is equipped with colorful cylinders for this process. A blob extractor is used to the color

Figure 2.2: An example of robots used in vision-based formation control

segmented image for each robot to isolate teammates within its own image.

On the other hand, the communication for data sharing brings extra computational cost. That is why; there are also some researches for efficient shape formation without the need of communication. In [31], a vision-based control strategy for decentralized stabilization of autonomous robot formation is presented. This algorithm uses leader-follower relative distance and bearing. The approach is based on an output feedback controller that uses a high-gain observer to estimate robots' relative positions. A pan-controlled camera on-board the follower robot is used for data measurements. In [32], the authors present a vision-based architecture for mobile robot detection and tracking from single frames using off-the-shelf on-board cameras and fiducial markers. The method aims to eliminate the need of inter-vehicle communication. In the proposed approach, markers are distributed on the back of each robot on truncated octagon shaped structures. Each face of these shapes have a code

that identifies the face and its position on the robot. These robots with the identifications can be seen in Figure 2.2. Model-based pose estimation can be stated as the nonlinear optimization problem. Relative position, bearing, heading angles, and leader's velocities are estimated by a dual unscented Kalman filter proposed in this study.

Chapter 3

Representation of Closed Curves

In this study, the desired formation shape for the robot group is represented using Elliptic Fourier Descriptors (EFDs) and Implicit Polynomials (IP). These representations bring a high flexibility for the formation shape because using these representations virtually any closed curve can be represented [15]. In this thesis, the implicit functions which are good for producing potential functions are used for shape formation and the parametric functions are employed for keeping the formation.

3.1 Elliptic Fourier Descriptors (EFD)

In shape formation of mobile robots, modeling of formation shape with finite set of measures is one of the main problems. In general, any closed curve can be described in terms of a set of Fourier series whose coefficients are called

Fourier descriptors (FD). Fourier Descriptors are convenient in describing 2-D and 3-D closed curves, as well as 3-D surfaces. Usage of FDs is advantageous because the shape information is concentrated in the low frequency parts [13] [51] [22] [56].

Granlund [13] proposed a method for representing closed shapes by using Elliptic Fourier Descriptors (EFDs). The basic idea of the elliptic Fourier descriptors is representing x and y coordinates of a point on the closed curve by a Fourier series.

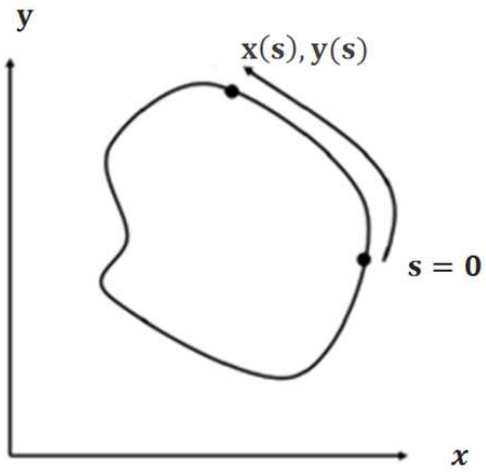

Figure 3.1: Basic idea of EFD

As it can be seen from Figure 3.1, a closed curve satisfies

$$\begin{aligned} x(s) &= x(s+L) \\ y(s) &= y(s+L) \end{aligned} \tag{3.1}$$

where L is the total length of the curve, $x(s)$ and $y(s)$ are periodic functions of the arclength, s. Using

$$t = \frac{s}{L} 2\pi \tag{3.2}$$

substitution these coordinates can be made 2π periodic functions of t, namely

$$s \in [0, L) \Rightarrow t \in [0, 2\pi) \tag{3.3}$$

$x(t)$ and $y(t)$ will have the following Fourier expansions:

$$\begin{aligned} x(t) &= a_0 + \sum_{k=1}^{n} a_k cos(kt) + b_k sin(kt) \\ y(t) &= c_0 + \sum_{k=1}^{n} c_k cos(kt) + d_k sin(kt) \end{aligned} \tag{3.4}$$

In this expression, a_0, c_0 are the mean values of $x(t)$ and $y(t)$, the coordinates of the points on the closed-bounded curve, a_k, b_k, c_k, d_k are elliptic Fourier coefficients which are used to model the closed-bounded curve and n is a positive number which represents the number of the harmonics used to represent the closed-bounded curve.

In [46], it is mentioned that each term of the summation in Equation 3.4 is the parametric form of an ellipse. The resulting contour can be viewed as a composition of rotation phasors, each individually represent an ellipse and rotating with a speed proportional to their harmonic number k. This can be seen in Figure 3.2, where the contour is constructed using three ellipses.

In Figure 3.2, C_o is the center of the first ellipse and each of other point is the center of the next higher ellipse. The straight lines represent the phasors for each ellipse shown at three different times. The point C_{ij} traces out the i^{th} ellipse at time j. The points C_{31}, C_{32}, C_{33} are three point on the defined closed curve.

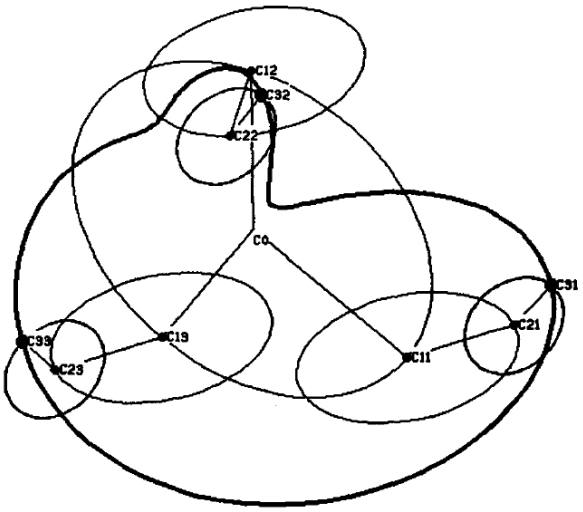

Figure 3.2: Closed curve constructed from three component ellipses

The computation of elliptic Fourier coefficients is considered in [22]. In the cases where we have an array of m points of the closed contour, the Fourier descriptors can be computed by a discrete approximation obtained by the Reiman summation as

$$\begin{aligned} a_0 &= \frac{1}{m}\sum_{i=1}^{m} x_i \qquad c_0 = \frac{1}{m}\sum_{i=1}^{m} y_i \\ a_k &= \frac{2}{m}\sum_{i=1}^{m} x_i cos(kt) \qquad b_k = \frac{2}{m}\sum_{i=1}^{m} x_i sin(kt) \\ c_k &= \frac{2}{m}\sum_{i=1}^{m} y_i cos(kt) \qquad d_k = \frac{2}{m}\sum_{i=1}^{m} y_i sin(kt) \end{aligned} \tag{3.5}$$

where m is the number of points describing the closed contour and x_i, y_i are the coordinates of each point.

3.2 Implicit Polynomial Representation

Virtually any closed-bounded curve can be represented by implicit polynomial equations. Although elliptic Fourier descriptors are also successful for representing complex closed-bounded curves, implicit polynomials allow a better treatment of several problems. For example, using implicit polynomials, it can be determined whether a point is on the curve and a measure for the distance of a point to the closed curve, an algebraic distance, can be defined. Using this property, the implicit function is used for producing potential fields for shape formation.

The representation of the implicit curve has the following form

$$F(x,y) = \sum_{0 \leq i+j \leq d} a_{ij} x^i y^j = 0 \tag{3.6}$$

where x, y are point positions on the curve, a_{ij} is the coefficients and d is the degree of the algebraic equation.

The implicit polynomial that represents the closed-bounded curve is not easily determined directly from points. However, using the method developed by [15], implicit polynomial representation for any closed-bounded curve can be found through the elliptic Fourier description of this curve. In this method, an n-harmonic elliptic representation of any 2-D closed curve is considered as:

$$\begin{aligned} x(t) &= a_0 + \sum_{k=1}^{n} a_k cos(kt) + b_k sin(kt) \\ y(t) &= c_0 + \sum_{k=1}^{n} c_k cos(kt) + d_k sin(kt) \end{aligned} \tag{3.7}$$

where $k = 1, \ldots, n$, (a_0, c_0) is the center of the curve and a_k, b_k, c_k, d_k are elliptic Fourier coefficients of the curve up to n Fourier harmonics. The well

known relation

$$cos(kt) = \frac{e^{jkt} + e^{-jkt}}{2} \qquad sin(kt) = \frac{e^{jkt} - e^{-jkt}}{2} \tag{3.8}$$

is used for substituting for $cos(kt)$ and $sin(kt)$ to obtain a complex exponential form of the elliptic Fourier descriptors in Equation 3.7:

$$\begin{aligned} x(t) &= A_0 + \sum_{k=1}^{n} A_k e^{jkt} + B_k e^{-jkt} \\ y(t) &= C_0 + \sum_{k=1}^{n} C_k e^{jkt} + D_k e^{-jkt} \end{aligned} \tag{3.9}$$

where

$$\begin{aligned} A_k &= \frac{a_k - jb_k}{2} \qquad B_k = \frac{a_k + jb_k}{2} \\ C_k &= \frac{c_k - jd_k}{2} \qquad D_k = \frac{c_k + jd_k}{2} \end{aligned} \tag{3.10}$$

for $k = 1, ..., n$ with $A_0 = a_0$ and $C_0 = c_0$. By substituting z for e^{jt},

$$\begin{aligned} x(z) &= A_0 + \sum_{k=1}^{n} A_k z^k + B_k z^{-k} \equiv \sum_{k=-n}^{n} g[k] z^k \\ y(z) &= C_0 + \sum_{k=1}^{n} C_k z^k + D_k z^{-k} \equiv \sum_{k=-n}^{n} h[k] z^k \end{aligned} \tag{3.11}$$

where

$$g[k] = \begin{cases} A_k & \text{if } k > 0 \\ A_0 & \text{if } k = 0 \\ B_k & \text{if } k < 0 \end{cases} \qquad h[k] = \begin{cases} C_k & \text{if } k > 0 \\ C_0 & \text{if } k = 0 \\ D_k & \text{if } k < 0 \end{cases} \tag{3.12}$$

If the g and h sequences are written as vectors

$$\begin{aligned} g &= \begin{bmatrix} B_n \dots B_1 & A_0 & A_1 \dots A_n \end{bmatrix} \\ h &= \begin{bmatrix} D_n \dots D_1 & C_0 & C_1 \dots C_n \end{bmatrix} \end{aligned} \tag{3.13}$$

then Equation 3.11 can be rewritten as

$$x(z) = g \cdot \vec{z} \qquad y(z) = h \cdot \vec{z} \tag{3.14}$$

where

$$\vec{z}^T = \begin{bmatrix} z^{-n} \dots z^{-l} & 1 & z \dots z^n \end{bmatrix} \tag{3.15}$$

A well-known time convolution property of the z-transform states that:

$$g[k] \Leftrightarrow x(z) \text{ and } h[k] \Leftrightarrow y(z) \Longrightarrow g[k] * h[k] \Leftrightarrow x(z)y(z) \tag{3.16}$$

Noting that convolution in discrete time domain corresponds to multiplication in the z domain. For example,

$$\begin{aligned} x^2 &= x(z)x(z) = Z\{g[k] * g[k]\}, \\ xy &= x(z)y(z) = Z\{g[k] * h[k]\}, \\ y^2 &= y(z)y(z) = Z\{h[k] * h[k]\}, \end{aligned} \tag{3.17}$$

The monomials for $x^p\ y^q$ can be found similarly by writing the following equation.

$$\underbrace{\begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \\ x^3 \\ \vdots \\ xy^{d-1} \\ y^d \end{bmatrix}}_{\Gamma} = \underbrace{\begin{bmatrix} 0\ldots0\,1\,0\ldots0 \\ g \\ h \\ g*g \\ g*h \\ h*h \\ g*g*g \\ \vdots \\ g*\underbrace{h*\ldots*h}_{n-1} \\ \underbrace{h*h*\ldots*h}_{n} \end{bmatrix}}_{P:ConvolutionMatrix} \underbrace{\begin{bmatrix} z^{-nd} \\ \vdots \\ z^{-1} \\ 1 \\ z^1 \\ \vdots \\ z^{nd} \end{bmatrix}}_{\vec{z}} \tag{3.18}$$

or simply $\Gamma = P\vec{z}$ for some complex matrix P of the size $(d+1)(d+2)/2 \times (2dn+1)$. P can be rewritten as

$$P = \hat{P} \begin{bmatrix} 1\,0\,0\ldots0\,0\,0 \\ i\,0\,0\ldots0\,0\,0 \\ 0\,1\,0\ldots0\,0\,0 \\ 0\,i\,0\ldots0\,0\,0 \\ \vdots\,\vdots\,\ddots\,\vdots\,\vdots\,\vdots \\ 0\,0\,\ldots1\ldots0\,0 \\ \vdots\,\vdots\,\vdots\,\ddots\,\vdots\,\vdots \\ 0\,0\,0\ldots0\,1\,0 \\ 0\,0\,0\ldots0\,i\,0 \\ 0\,0\,0\ldots0\,0\,1 \\ 0\,0\,0\ldots0\,0\,i \end{bmatrix} \tag{3.19}$$

for some unique, real $(d+1)(d+2)/2 \times (2d^2+1)$ matrix $\hat{P}$. Then the "largest" $(d+1)(d+2)/2 - 1 = d(d+3)/2$ columns of $\hat{P}$ are defined by $QR = \hat{P}E$ using an orthogonal-triangular decomposition. In this equation Q is an unitary matrix, R is an upper triangular matrix with diagonal elements in order by decreasing absolute values, and E is a permutation matrix which orders the coulumns of $\hat{P}E$ in correspondance with those of QR. The first $d(d+3)/2$ columns of $\hat{P}E$ is defines as $\tilde{P}$. Then vector v that annihilates $\tilde{P}$ can be found from the yielding implicit polynomial function as

$$v\Gamma = f_d(x, y) = 0 \tag{3.20}$$

3.3 Examples for Closed-Bounded Curve Representations

3.3.1 Representation of a Quadrangle Using 3 Harmonics

In this example, a quadratic shape will be represented using the methods presented in Sections 3.1 and 3.2. The complex curve in Figure 3.3 has been represented by elliptic Fourier descriptors using 3 harmonics and with a corresponding implicit polynomial of degree 6. The Fourier coefficients are below:

$$a_0 = 0.0200 \qquad c_0 = 0.0300$$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0.1034 \\ 0.0009 \\ -0.1668 \end{bmatrix} \quad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} -1.3901 \\ -0.0003 \\ -0.1675 \end{bmatrix}$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} -1.3916 \\ -0.0004 \\ 0.1670 \end{bmatrix} \quad \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} -0.1018 \\ 0.0005 \\ -0.1671 \end{bmatrix}$$

The implicit polynomial representation of the closed curve is found as follows:

$$\begin{aligned} F(x,y) = 1.0000x^6 - 0.0148x^5y + 3.0032x^4y^2 - 0.0296x^3y^3 + 3.0063x^2y^4 - 0.0148xy^5 \\ +1.0031y^6 - 0.1292x^5 - 0.1650x^4y - 0.1993x^3y^2 - 0.3409x^2y^3 - 0.0699xy^4 \\ -0.1756y^5 14.0521x^4 + 77.7449x^3y - 70.3379x^2y^2 - 77.5463xy^3 + 14.1695y^4 \\ -3.3998x^3 - 0.5792x^2y 9.3872xy^2 - 0.2937y^3 + 135.7740x^2 - 0.0244xy \\ +134.6475y^2 - 5.1995x - 7.6102y - 228.2236 = 0 \end{aligned}$$

These representations can be seen in Figure 3.3. This figure is plotted according to the found EFD and IP representations of the curve. It is seen that a complex curve is represented successfully using both elliptic Fourier descriptor and implicit polynomial. It is also seen that the Fourier reprepresentation and implicit function representation matches well.

3.3.2 Representation of a Star-Like Shape Using 6 Harmonics

A more complex shape is represented by using 6 harmonics with EFD and with a corresponding implicit function of degree 12 using the implicitization

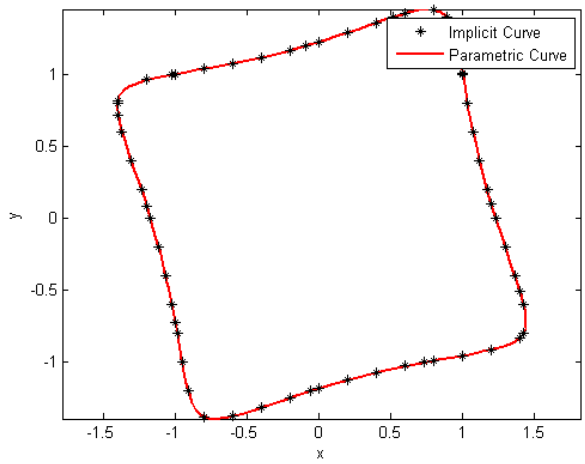

Figure 3.3: Representation of a quadrangle by EFD and Implicit polynomial method presented in [15]. The Fourier coefficients are found as:

$$a_0 = -0.1000 \qquad c_0 = 0.1000$$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} 0.1097 \\ -0.0024 \\ 0.0021 \\ -0.1389 \\ -0.0006 \\ 0.0658 \end{bmatrix} \qquad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} -1.3958 \\ -0.0008 \\ -0.0026 \\ -0.0974 \\ -0.0009 \\ 0.0710 \end{bmatrix}$$

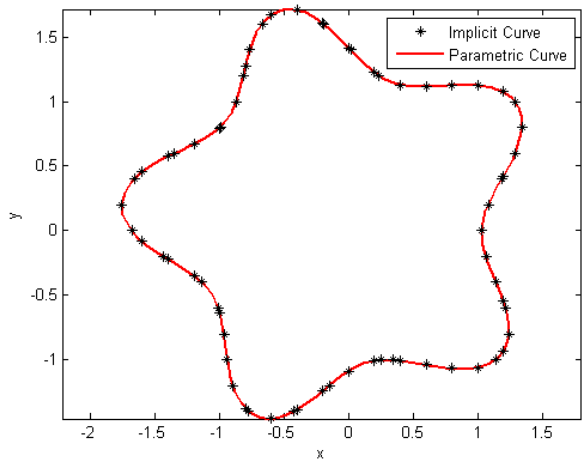

Figure 3.4: Representation of a star-like shape by EFD and Implicit polynomial

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} -1.3929 \\ -0.0014 \\ -0.0056 \\ 0.0930 \\ -0.0006 \\ 0.0748 \end{bmatrix} \quad \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix} = \begin{bmatrix} -0.1018 \\ 0.0011 \\ -0.0009 \\ -0.1284 \\ 0.0002 \\ -0.0684 \end{bmatrix}$$

Resulting 12^{th} degree polynomial has 91 coefficients which will not be shown here. Figure 3.4 plots both EFD and implicit polynomial curve.

Chapter 4

Control of Non-Holonomic Mobile Robots

A robot is nonholonomic if it cannot instantaneously move in all available directions and has some non-integrable velocity constraints on its movements. For example, car-like vehicles are non-holonomic because they cannot move sideways. The control of non-holonomic mobile robots is complicated because their controllable degrees of freedom (DOF) is less than effective DOF. In the recent years, there has been a significant research interest on the control of non-holonomic systems. Some of the successful studies can be found in [43], [42], [29], [18], [40], [12], [41].

In this work, non-holonomic mobile robots are used in the implementations. These mobile robots have two actuated wheels in the front and two spherical wheels at the back which moves freely according to the leading of the ones in the front. This kind of robots is referred as "Unicycle" robots. In the following sections, the modeling and trajectory control of this type of

non-holonomic mobile robots will be presented.

4.1 Modeling of Non-Holonomic Robots

The modeling of non-holonomic mobile robots is difficult because of the constraints. Simple dynamic and kinematic equations are not sufficient for modeling these robots. For example, consider a 2-wheeled non-holonomic mobile robot moving in 2-D, the orientation of this robot has an effect on its movements in x and y directions. That is why; the orientation of the mobile robot should also be considered in the kinematic model of this robot. There is a widely known kinematic model for non-holonomic unicycle robots which is given as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} u_1 cos\theta \\ u_1 sin\theta \\ u_2 \end{pmatrix} \tag{4.1}$$

where x and y are the Cartesian coordinates of the robot, θ is its orientation angle with respect to the x axis, u_1 and u_2 are respectively its linear and angular velocities.

The pose of the robot is represented by its position (x, y) and its orientation θ. In the mentioned non-holonomic model, these pose variables are considered to be outputs and the velocity variables are the inputs. The linear and angular velocities should be designed for the robot to achieve the desired pose. The mathematical equations of the non-holonomic model can be rewritten considering the two inputs of the system as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos\theta \\ sin\theta \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2 \tag{4.2}$$

The velocities u_1 and u_2 in the above equations are related to the linear velocities of the right and left wheels, u_R and u_L as:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} (u_R + u_L)/2 \\ (u_R - u_L)/(2\lambda) \end{pmatrix} \tag{4.3}$$

where λ is the half length of the wheel axis as shown in Figure 4.1.

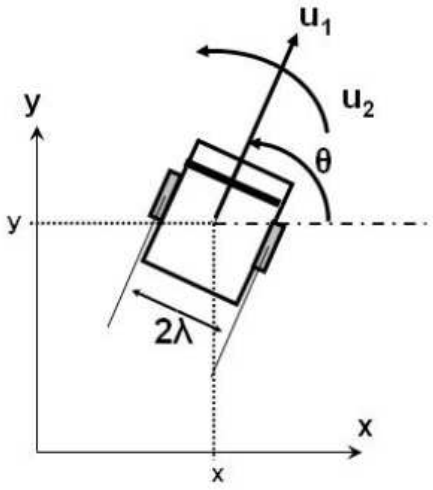

Figure 4.1: A Unicycle robot

4.2 Trajectory Control of Non-Holonomic Mobile Robots

The control of non-holonomic mobile robots is difficult since the non-holonomic mobile robots have more DOFs than controllable ones. As it can be seen from Equation 4.2, only two controls, the linear and angular velocities of the robot, are used to control three outputs for the pose of the robot.

Although non-holonomic mobile robots are completely controllable in their configuration space, they cannot be stabilized to a desired pose by using smooth state-feedback control [43]. However, the feedback stabilization of a point on a non-holonomic mobile robot was shown to be possible in [42]. In that work, C. Samson and K. Ait-Abderrahim proved that feedback stabilization of the robot's pose around the pose of a "virtual reference robot" is possible provided that the reference robot keeps moving. This control problem has also been considered by Morin and Samson [29] and tracking of time-variant reference trajectories are presented.

For time-variant reference trajectory tracking, the reference trajectory should satisfy the nonholonomic constraint. This is ensured by defining the trajectory using a virtual reference robot which moves according to the model:

$$\begin{pmatrix} \dot{x_r} \\ \dot{y_r} \\ \dot{\theta_r} \end{pmatrix} = \begin{pmatrix} cos\theta_r \\ sin\theta_r \\ 0 \end{pmatrix} u_{1r} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_{2r} \tag{4.4}$$

where (x_r, y_r) is position of the virtual reference robot according to the Carte-

sian coordinates and θ_r is its orientation. u_{1r} and u_{2r} are respectively its linear and angular velocities.

If x_r, y_r and θ_r are continuously differentiable and bounded as $t \rightarrow \infty$ it can be shown that:

$$\begin{pmatrix} u_{1r} \\ u_{2r} \\ \theta_r \end{pmatrix} = \begin{pmatrix} \dot{x_r}cos\theta_r + \dot{y_r}sin\theta_r \\ (\ddot{y_r}\dot{x_r} - \ddot{x_r}\dot{y_r})/(\dot{x_r}^2 + \dot{y_r}^2) \\ arctan(\dot{y_r}/\dot{x_r}) \end{pmatrix} \tag{4.5}$$

The tracking errors $\tilde{x}$, $\tilde{y}$ and $\tilde{\theta}$ are defined as the difference between the pose of actual robot and the virtual reference robot as follows:

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} - \begin{pmatrix} x_r \\ y_r \\ \theta_r \end{pmatrix} \tag{4.6}$$

For simplifying the control problem, new definitions for tracking errors, $e1$, $e2$ and $e3$, are obtained based on the kinematic model of non-holonomic robot in Equation 4.1 as:

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{pmatrix} \tag{4.7}$$

Considering the inverse transformation of Equation 4.7 it can be shown that as $t \rightarrow \infty$

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{pmatrix} \rightarrow 0 \qquad \text{if} \quad \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} \rightarrow 0 \tag{4.8}$$

In [41], it is shown that with a proper selection of constant control gains, $k_1 > 0$ and $k_2 > 0$, all tracking errors can be regulated to zero using the following controls, u_1 and u_2, for time-variant reference trajectories:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -k_1 e_1 + u_{1r} cose_3 \\ -u_{1r}(sine_3/e_3) - k_2 e_3 + u_{2r} \end{pmatrix} \tag{4.9}$$

Chapter 5

Shape Formation Control of Mobile Robots

Shape formation is an important problem in mobile robot coordination because many coordination tasks need the robots to maintain a desired formation. It has a key role in mobile robotic tasks like navigation, carrying large objects, search-and-rescue and hunting behaviors.

The desired formation shape, the number and type of mobile robots used in formation change according to different task definitions. For example, for carrying a large, complex shaped object, the mobile robots should form this shape. This may also be required for search tasks, where an area of a complex shape is searched; the robots should be able to form the shape of this area. There may be constraints for the formation shape because of some environmental conditions like obstacles or restricted areas for robots which prevent formation of simple shapes. Besides, different number and types of robots may be required in a task. Some tasks require very crowded

groups of robots and some tasks require different types of robots with different capabilities and sizes. Designing a shape formation control which is flexible in the means of these factors presents a challenge in robot coordination.

In this study, a flexible shape formation method for maintaining and keeping complex closed curve shapes is presented. The proposed method is applicable for complex formation shapes and robot groups of different number of robots and heterogenous groups.

In the presented method, two control phases are introduced for shape formation. In the beginning of the first phase, the robots are randomly positioned on the predefined workplace. The aim of this control is to reach the desired formation shape while maintaining a coordination. The aim of this coordination is to keep a predefied distance between each robot and its neigbours. This coordination prevents collisions between robots and provides that the group members stay together. The second phase starts when the desired shape is reached. The aim of control in the second phase is to keep the formation shape while allowing the robots travel around the contour shape. While keeping the desired shape, the coordination should again be considered.

For the design of the shape formation control, the robots are modeled as point particles which have the kinematic model for i^{th} robot as:

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \end{pmatrix} = \begin{pmatrix} u^i_{total} \\ v^i_{total} \end{pmatrix} \tag{5.1}$$

where $\dot{x}_i$ and $\dot{y}_i$ are the velocities of the particle in the x and y directions respectively.

Then the formation control design is extended to non-holonomic mobile

robots. In this method, the desired positions found for the point-particle robots are used as reference for non-holonomic robots and these robots are controlled using trajectory tracking method mentioned in Section 4.2.

In both phases of the shape formation, the final control input for the robots $\begin{pmatrix} u^i_{total} & v^i_{total} \end{pmatrix}^T$ is calculated as the sum of coordination control and formation control components, namely

$$\begin{pmatrix} u^i_{total} \\ v^i_{total} \end{pmatrix} = \begin{pmatrix} u^i_{coord} \\ v^i_{coord} \end{pmatrix} + \begin{pmatrix} u^i_{formation} \\ v^i_{formation} \end{pmatrix} \tag{5.2}$$

where $\begin{pmatrix} u^i_{coord} & v^i_{coord} \end{pmatrix}^T$ is the coordination component of the control and $\begin{pmatrix} u^i_{formation} & v^i_{formation} \end{pmatrix}^T$ is formation component of the control. The design of these components is presented in the following sections.

5.1 Coordination Control

The robots are assigned to be in coordination while forming the desired shape during the both two phases of the formation control. The coordination avoids collisions between the robots. It can also provide optimum usage of the group energy. For example, for a task that the robots are assigned for searching an item in the defined field, it increases the efficiency of the work if the robots keep a distance between them according to the range of their sensors. Also in the case that the robot group carries a load around the defined curve, keeping the desired distance between the neighbours will be extremely important.

In the proposed coordination, each robot keeps a predefined desired distance between itself and its two nearest neighbours. This is a reasonable

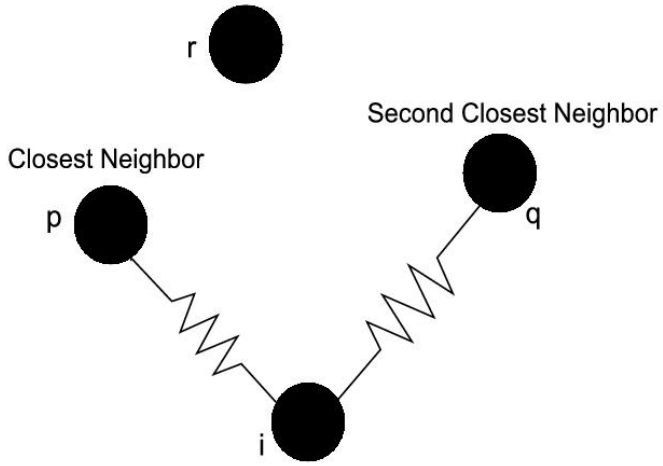

Figure 5.1: Modeling of coordination control

coordination because while travelling on a line curve, there are two neighbours which should be considered; the robot at the back of the robot and the one in the front. Since these two robots will consider their own neighbors at the back and in the front, as a result of this coordination, the robot group is expected to line up around the curve with the desired distance kept between each robot.

The coordination between each robot and its two nearest neighbours is modeled with virtual linear springs. These springs are placed between each robot and its two nearest neighbours, as it can be seen in Figure 5.1. Each of these springs produces a virtual attraction or repulsion force according to the distance between the robots. The coordination component of the control is found with the sum of these spring forces.

The proposed virtual springs have a normal length which is equal to the desired distance between the neighbours. The virtual force produced by the

springs is linearly proportional to the difference between the actual distance and the desired distance of the neighbouring robots. The force of the spring is on the direction of the vector from the i^{th} robot to its neighbor. The control component of coordination for each robot is the sum of these forces from its two nearest neighbours, namely:

$$\begin{pmatrix} u^i_{coord} \\ v^i_{coord} \end{pmatrix} = k(d_{desired} - d^{ip}_{actual}) \begin{pmatrix} x_i - x_p \\ y_i - y_p \end{pmatrix} / \left\| \begin{matrix} x_i - x_p \\ y_i - y_p \end{matrix} \right\| + k(d_{desired} - d^{iq}_{actual}) \begin{pmatrix} x_i - x_q \\ y_i - y_q \end{pmatrix} / \left\| \begin{matrix} x_i - x_q \\ y_i - y_q \end{matrix} \right\| \quad (5.3)$$

where k is positive number which is an adaptable spring constant. The unit of k is $1/seconds$. p and q are the indices for the robots that are the nearest two neighbors of i^{th} robot. d^{ip}_{actual} and d^{iq}_{actual} are the actual distances of the robot i from the robots p and q respectively. (x_p, y_p) and (x_q, y_q) are the x and y position coordinates of the robots p and q.

The spring constant in the Equation 5.3 is different when the robot is in shape formation and keeping formation phases. A larger spring constant is used in the first phase than in the second one. The reason is that the robots are desired to keep a more strict coordination while they are reaching the desired curve. But when they reache the curve, the spring constant is decreased to make the robots be able to keep the shape of the complicated closed curve. [14] is a similar study for coordinating the robots by modeling with virtual springs and adapting the spring constants during the formation process. The strictness of the coordination can be changed with the spring constant for any specific task. The effect of the coordination control can be increased by increasing the spring constant when a task requires the desired

distance between the robots to be kept more strictly.

5.2 Formation Control Using Implicit Polynomial Potential Functions

According to the proposed method, in the first phase robots are controlled to achive the desired formation shape. This control is used for the time interval from the beginning, when the mobile robots are randomly positioned in the predefined workplace, until the desired formation shape is achieved by the robots.

The implicit polynomial representation of the curve given as $F_d(x, y) = 0$ is used for the design of the formation component of control input. The position error of the i^{th} robot according to the curve is given by the algebraic distance function using the implicit equation as

$$e^i_{form} = F(x_i, y_i) \tag{5.4}$$

where e^i_{form} is the position error of i^{th} robot with respect to the desired curve and x_i and y_i are positions of this robot. Since the aim of the shape formation control is to make this position error decrease to zero for the shape formation to be achieved, it will be designed to force the error to decrease exponentially, namely

$$\dot{e}^i_{form} = -\lambda e^i_{form} \tag{5.5}$$

where λ is a positive number. Substituting $e^i_{form} = F(x_i, y_i)$ into Equation 5.5 yields

$$\dot{F}(x_i, y_i) = -\lambda F(x_i, y_i) \tag{5.6}$$

By using chain rule of differentiation, Equation 5.6 can be rewritten as:

$$F_x(x_i, y_i)\dot{x}_i + F_y(x_i, y_i)\dot{y}_i = -\lambda F(x_i, y_i) \tag{5.7}$$

where $F_x(x_i, y_i)$ and $F_y(x_i, y_i)$ are the partial derivatives of function $F(x, y)$ in the x and y directions at the point (x_i, y_i). Equation 5.7 can be rewritten in the vector form as:

$$\begin{pmatrix} F_x(x_i, y_i) & F_y(x_i, y_i) \end{pmatrix} \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \end{pmatrix} = -\lambda F(x_i, y_i) \tag{5.8}$$

which in turn implies that

$$\begin{pmatrix} F_x(x_i, y_i) & F_y(x_i, y_i) \end{pmatrix} \begin{pmatrix} u^i_{formation} \\ v^i_{formation} \end{pmatrix} = -\lambda F(x_i, y_i) \tag{5.9}$$

Equation 5.9 will be used for designing the two control inputs $u^i_{formation}$ and $v^i_{formation}$. It can be seen that this is an underdetermined case because there are two unknowns, $u^i_{formation}$ and $v^i_{formation}$ but we have one equation. In this case, infinitely many solutions can be found. In this study, the optimum solution, which requires less power and time from the system, will be used as the control input. Considering an equation in the form

$$AX = Y \tag{5.10}$$

where Y is a $m \times 1$ vector, X is a $n \times 1$ vector and A is a $m \times n$ matrix. If $m < n$, this equation is an underdetermined equation as in the case of Equation 5.9. It is known that the optimum solution of X satisfying the equation can be found using the Pseudo inverse of A matrix as:

$$X = A^T(AA^T)^{-1}Y \tag{5.11}$$

When the same approach is applied to Equation 5.9, assuming that

$$A = \begin{pmatrix} F_x(x_i, y_i) & F_y(x_i, y_i) \end{pmatrix} = \nabla F(x_i, y_i)^T$$
$$X = \begin{pmatrix} u^i_{formation} & v^i_{formation} \end{pmatrix}^T$$
$$Y = -\lambda F(x_i, y_i)$$

where X is a 2×1 vector, Y is a scalar and A is 1×2. The optimum solution is found as

$$\begin{pmatrix} u^i_{formation} \\ v^i_{formation} \end{pmatrix} = -\lambda \nabla F(x_i, y_i)(\nabla F(x_i, y_i)^T \nabla F(x_i, y_i))^{-1} F(x_i, y_i) \quad (5.12)$$

Since $(\nabla F(x_i, y_i)^T \nabla F(x_i, y_i))$ is a scalar, Equation 5.12 can be rewritten as:

$$\begin{pmatrix} u^i_{formation} \\ v^i_{formation} \end{pmatrix} = -\lambda \frac{1}{||\nabla F(x_i, y_i)| \, |^2} F(x_i, y_i) \begin{pmatrix} F_x(x_i, y_i) \\ F_y(x_i, y_i) \end{pmatrix} \quad (5.13)$$

5.3 Keeping Formation Using Elliptic Fourier Descriptors

In the second phase of the formation control, after the robots reach the desired formation, a new control is proposed. The aim of this control is to allow the robots to travel around the predefined formation curve while still keeping the formation and coordination with the other robots. The potential application areas of this control include the tasks in which the robots are assigned to search a substance or carry some load around the defined curve.

For the design of the formation keeping control, the parametric representation of the desired formation is used. This representation is a function of time t. That's why; the parametric representation is good for trajectory generation. The desired position $(x^*(t), y^*(t))$ at time t can be found using the Equation 3.4, as:

$$x^*(t) = a_0 + \sum_{k=1}^{n} a_k cos(kt) + b_k sin(kt)$$
$$y^*(t) = c_0 + \sum_{k=1}^{n} c_k cos(kt) + d_k sin(kt) \tag{5.14}$$

This definition of reference gives a trajectory which moves around the desired closed curve. This trajectory representation starts from a specific point $(x(0), y(0))$. In the application of this method, it is more sufficient for a robot to start traveling from the point that it reached the formation curve than starting from the point $(x(0), y(0))$. That is why, the trajectory is shifted for each robot to make it start with the point they reach the curve according to the equation below:

$$x_i^*(t) = x^*(t + t_i)$$
$$y_i^*(t) = y^*(t + t_i) \tag{5.15}$$

where $(x_i^*(t), y_i^*(t))$ is the desired position for robot i at time t and $x_i^*(t+t_i)$, $y_i^*(t+t_i)$ are found according to Equation 5.14. In this equation, the shift in the time, t_i, satisfies the equation below:

$$x^*(0 + t_i) = x_i^*(0)$$
$$y^*(0 + t_i) = y_i^*(0) \tag{5.16}$$

where $(x_i^*(0), y_i^*(0))$ is the position that the i^{th} robot reaches the desired curve.

The formation keeping control is designed according to this trajectory definition. For simplicity, the general trajectory presentation $(x_i^*(t), y_i^*(t))$ will be used, without loss of generality, while presenting the design of formation control.

The position error e^i_{form} of the i^{th} robot is defined as the difference between the desired position and the actual position of this robot at time t, namely:

$$e^i_{form} = \begin{pmatrix} x_i^*(t) \\ y_i^*(t) \end{pmatrix} - \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} \tag{5.17}$$

where $x_i^*(t)$ and $y_i^*(t)$ are the desired positions and $x_i(t)$ and $y_i(t)$ are the actual positions.

The aim of the control is to decrease this position error to zero. That is why; the control is designed to make the error decrease exponentially as in the following equation

$$\dot{e}^i_{form} = -\lambda e^i_{form} \tag{5.18}$$

The time derivative of the position error, $\dot{e}^i_{form}$ is found by taking the time derivatives of both sides in the Equation 5.17, namely:

$$\dot{e}^i_{form} = \begin{pmatrix} \dot{x}_i^*(t) \\ \dot{y}_i^*(t) \end{pmatrix} - \begin{pmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \end{pmatrix} \tag{5.19}$$

which in turn implies that

$$\dot{e}^i_{form} = \begin{pmatrix} \dot{x}_i^*(t) \\ \dot{y}_i^*(t) \end{pmatrix} - \begin{pmatrix} u^i_{formation} \\ v^i_{formation} \end{pmatrix} \tag{5.20}$$

Then the formation keeping control $\begin{pmatrix} u^i_{formation} & v^i_{formation} \end{pmatrix}^T$ is found as:

$$\begin{pmatrix} u^i_{formation} \\ v^i_{formation} \end{pmatrix} = \lambda e^i_{form} + \begin{pmatrix} \dot{x}^*_i(t_i) \\ \dot{y}^*_i(t_i) \end{pmatrix} \tag{5.21}$$

5.4 Shape Formation Control of Nonholonomic Robots

The shape formation method presented in the previous sections was designed for point-particle model. This method is extended for the formation control of non-holonomic mobile robots by using the desired positions found from this model as references for the nonholonomic robots. A block diagram explaining the shape formation control method for a non-holonomic mobile robot can be seen in Figure 5.2.

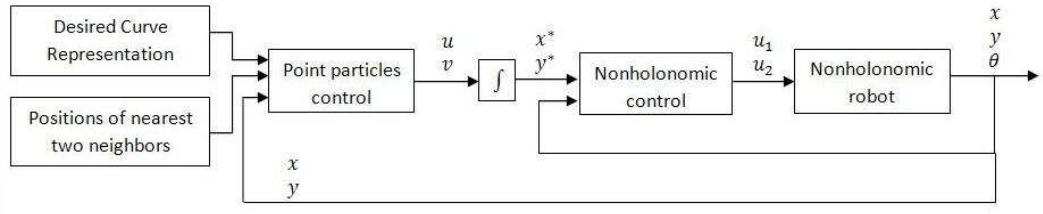

Figure 5.2: Shape formation control for nonholonomic robot

In this figure, the "Desired Curve Representation" block provides the EFD and implicit polynomial representations of the desired formation curve.

"Positions of nearest two neighbors" block finds the two group members which are nearest to the robot and gives their positions. Using these parameters and the position data of the non-holonomic robot, "Point particles control" block produces the shape formation control, u and v. The desired positions x^* and y^* are found by taking the integral of these control inputs. The initial positions of the point particle robots are the same as the initial positions of the non-holonomic robots.

The "Non-holonomic control" block produces the appropriate control inputs, u_1 and u_2, for the non-holonomic mobile robot according to the desired positions and the pose of the robot. This block uses the trajectory tracking control principles presented in Section 4.2. If the control inputs are above the limits of the robot, they are saturated.

Chapter 6

Simulation Results and Discussions

The simulations are done for testing the performance of the proposed formation control presented in Chapter 5. In the first part of the experiments, the mobile robots are modeled as point particles. In the second part of the simulations, robots are modeled as non-holonomic robots according to the model presented in Chapter 4. The desired positions in the point particle robot model have been used as references to non-holonomic robots in the second part.

The first simulation in each part is done with one robot to see the efficiency of the proposed formation control. The remaining simulations are done with multiple robots to see the success of the control with both formation and coordination components.

In the simulations, the robots are initially randomly placed in a predefined area in the workplace. The task of the robots is to reach the desired

formation curve and travel around the curve while keeping the formation in a coordinated manner.

The simulations are performed in Matlab. The program is written to be modular so that the simulations can be carried out with any desired number of robots.

In simulations, parameters are chosen to be $\lambda = 5$, $k_{ShapeFormation} = 6$, $k_{KeepFormation} = 1$ and $d_{desired} = 1$.

6.1 Simulations with Point Particle Model

In this part of the simulations, the robots are modeled as point particles. Three simulations are done in this section, which are simulation with a single robot and with groups of robots with 5 and 6 agents.

6.1.1 Simulation Results for a Single Robot

In this part, one robot is simulated with a complex closed curve which is represented by a Fourier descriptor function using 7 harmonics and a corresponding implicit polynomial with degree of 14 using the methods in Chapter 3. The initial position of the robot is: $x = 2.5$, $y = -2$. The route of the robot under the control of the proposed formation control can be seen in Figure 6.1.

In Figure 6.1, the solid thin line shows the desired formation shape. The thick black line represents the route of the mobile robot.

It is clearly seen that the proposed method is successful to make the robot directly reach the desired complex curve. In the second part of the control,

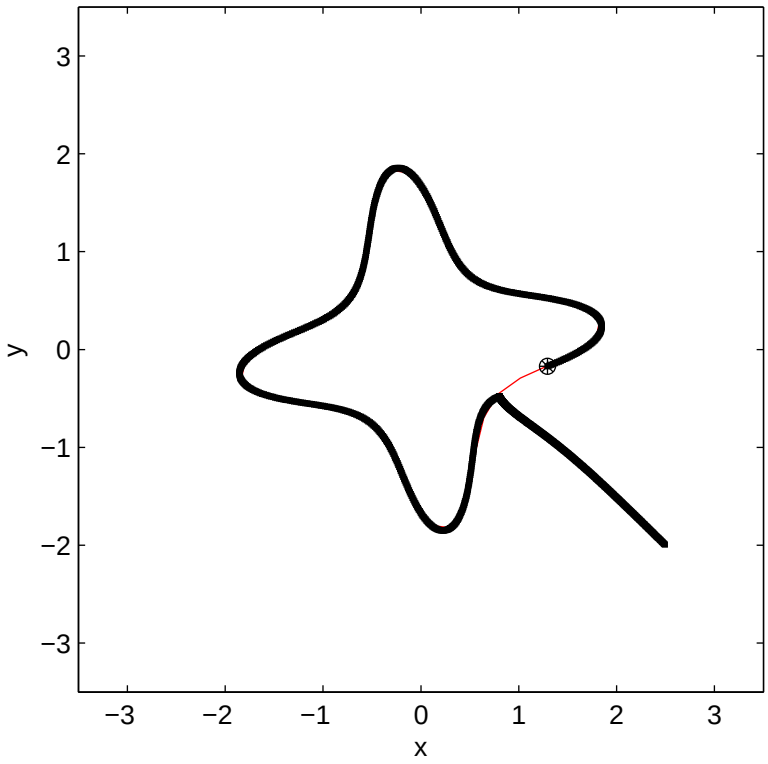

Figure 6.1: Route of a point particle mobile robot

the robot travels around the formation shape successfully.

6.1.2 Simulation Results for 5 Robots

In this part, the proposed method is simulated on a group of 5 robots. The desired formation shape is an ellipse. This desired shape is represented by an elliptic Fourier descriptor with 1 harmonics and corresponding implicit polynomial of degree 2. This desired formation shape and the behaviours of the robots can be seen on Figure 6.2.

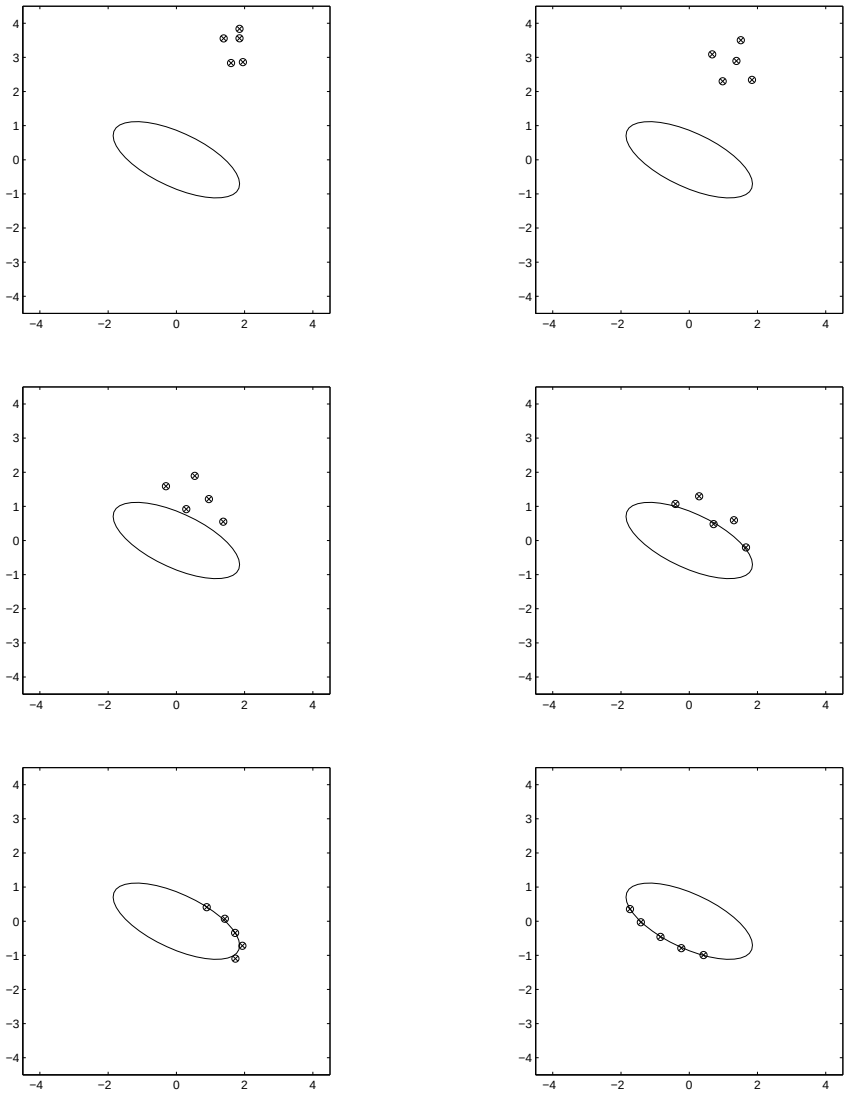

Figure 6.2: Desired formation (ellipse) with 5 point particle robots

It is seen in the figure that the robots approached the desired curve by keeping the desired distance between their nearest two neighbours. When the robots reach the curve, they start traveling around that curve. It is seen that a desired distance is kept between the robots while traveling. The robots were able to keep the formation successfully.

6.1.3 Simulation Results for 6 Robots

In this simulation, the proposed method is applied on a group of 6 robots. The desired pattern is a more complicated star shape which is represented by 7 harmonics and a corresponding implicit polynomial of degree 14. The resulting behaviours of the mobile robots with the proposed formation control can be seen in Figure 6.3

Examination of these figures reveals the fact that although the initial positions of the robots are far away from the desired formation curve, proposed method enables robots to achieve and maintain the desired formation while keeping good coordination with each other.

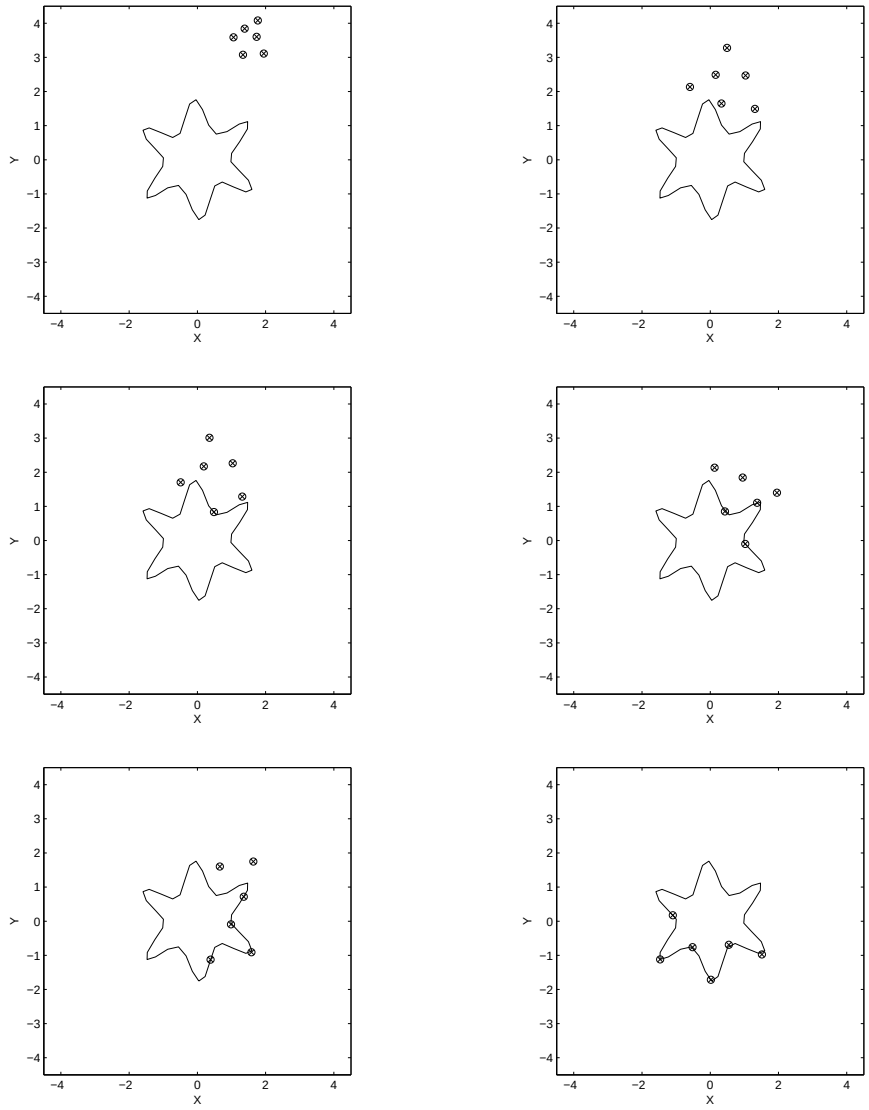

Figure 6.3: Desired formation (star shape) with 6 point particle robots

6.2 Simulations with Non-Holonomic Model

In this part of the simulations, the robots are modeled as non-holonomic mobile robots according to the model presented in Chapter 4. In these simulations, the position references for the non-holonomic mobile robots are found by the desired positions of the mobile robots in the point-particle model, which is simulated in the previous section. This simulation is built on the previous simulation model in that sense.

The success of the proposed shape formation control design mobile robots with point particle model, has been seen by the simulation in the previous section. The aim of the simulations in this section is to see the success of the proposed shape formation control on non-holonomic mobile robots. In this simulations, u_1, the linear velocity of the robots, is limited to 0.04 and u_2, the angular velocity, is limited by 0.04π in a unit time. These are reasonable limitations when the unit time is thought to be 0.01 seconds.

In the first simulation, there is a single robot with a desired formation shape which is a complex curve. The other two simulations are done with a robot group of 5 robots with an elliptic formation shape and a group of 6 robots with a more complex formation shape.

6.2.1 Simulation Results for a Single Robot

In this part of the simulation, a single non-holonomic robot is simulated. This robot starts from the position $x = 2.5$, $y = -2$.

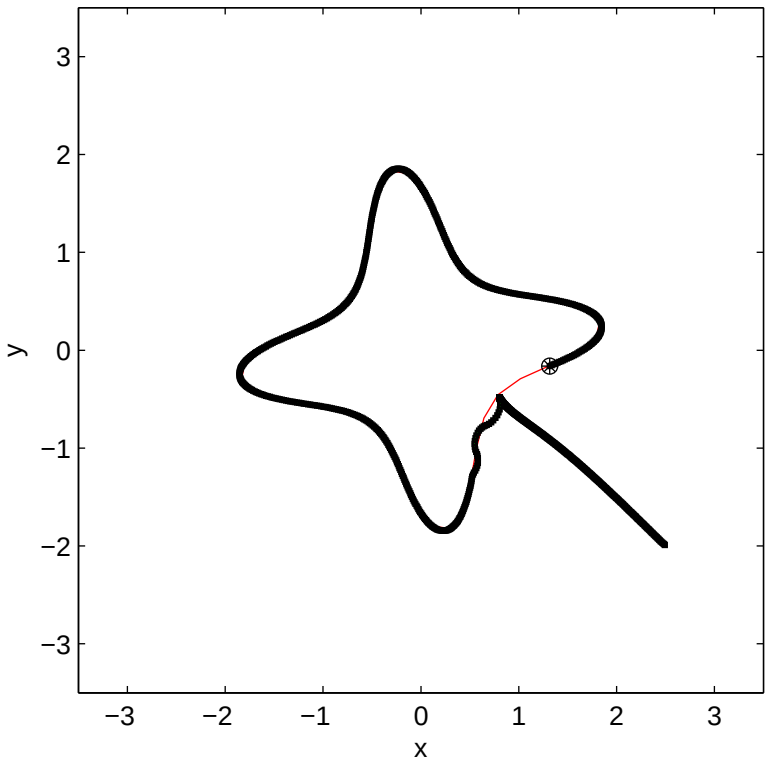

Figure 6.4: Route of non-holonomic mobile robot

When the figure is examined, it can be seen that the robot reached the desired curve successfully. It is seen that some oscillations occured after the robot reaches the curve. This oscillations occured because of the non-holonomic restrictions on the velocity of the robot. After the robot is settled on the formation curve, the robot is seen to travel along the curve successfully.

6.2.2 Simulation Results for 5 Robots

In this part, a group of 5 non-holonomic robots is simulated. In this simulation, the desired curve is an ellipse which is represented by EFDs with 1 harmonic and a corresponding IP of degree 2. The result of this simulation can be seen in Figure 6.5.

In the figure, the black points are the non-holonomic mobile robots and the red points are the desired positions for these robots. It can be seen that the robots reached the formation curve and traveled around this curve in a coordination. Because of the non-holonomic constraints on the linear and angular velocities, the robots could not catch the references at first but it is seen that they catched these references after a while and reached the formation as in the point-particle model.

6.2.3 Simulation Results for 6 Robots

In this simulation, a group of 6 non-holonomic mobile robots are simulated. The desired formation shape is a more complex one which is represented by EFDs with 7 harmonics and a corresponding IP of degree 14. The result can be seen in Figure 6.6.

In the figure, black points are the non-holonomic robots and the red ones are reference points. Inspecting the figures show that the non-holonomic robots reached the curve and traveled around it by keeping the coordination. It is seen that the non-holonomic constraints caused some error on the shape formation but this error decreased by the time.

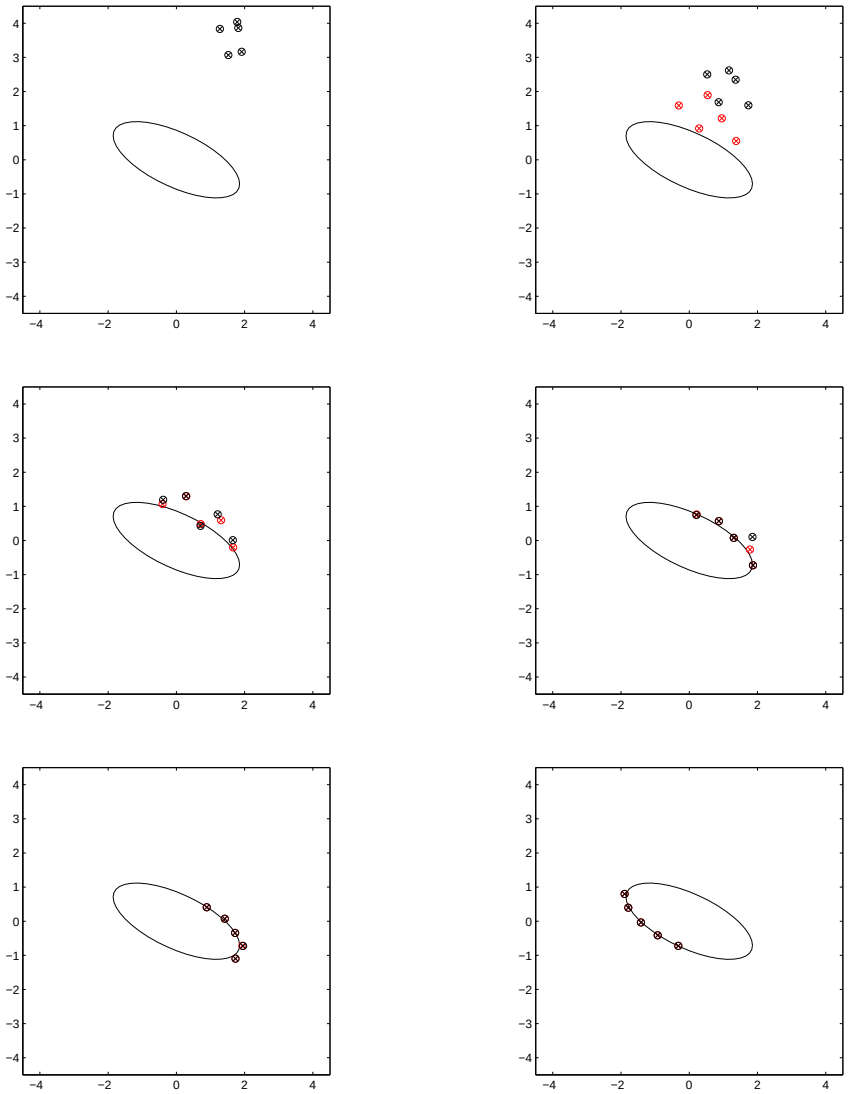

Figure 6.5: Desired formation (ellipse) with 5 non-holonomic robots

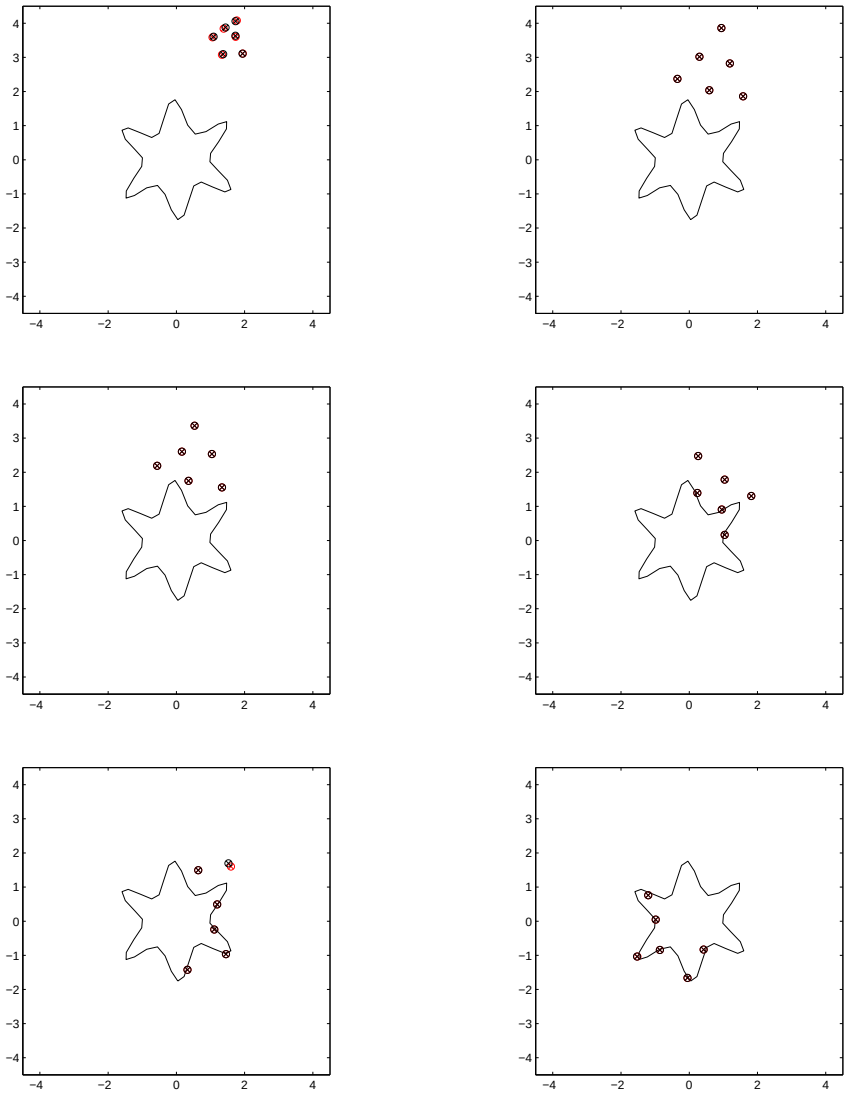

Figure 6.6: Desired formation (star shape) with 6 non-holonomic robots

Chapter 7

Experimental Results

In the previous chapters, our method for shape formation has been described and the success of this method has been shown with simulations. In this chapter, the performance of the algorithm will be examined using real mobile robots.

7.1 Assumptions

The implementations are done on a flat surface of the size $260cm \times 200cm$. It is assumed that there are no obstacles other than the robots. In the implementations, robots are assumed to be visible during the implementation by the vision system which will be presented in detail in Subsection 7.2.2. It is also assumed that the target formation shape is predefined.

7.2 The System

7.2.1 Our Robots

In the implementations, a modified version of Trilobot [38] mobile robots are used. The implementation robot can be seen in Figure 7.2.1. Trilobot is a non-holonomic mobile robot with two actuated wheels. It has several sensors including sonar range sensor, passive infrared motion detector, a digital compass, whiskers and motor encoders. 8 whiskers are placed around the robot near the ground and other sensors are on a pan-tilt head structure. The body dimensions of the robot are $30cm \times 30cm \times 30cm$.

Figure 7.1: The robot used in the shape formation implementations

Trilobot has an onboard microcontroller which is responsible for driving motors and manage the data from the sensors. In Trilobot, a 2k EEPROM is available to the master processor for storage of parameters. Simple programs can be written to the unoccupied space in this memory for onboard processor.

For complex programs, the robot is available to be controlled from an external PC. The microcontroller can communicate with a PC using serial

RS-232 interface. Using the communication protocol of Trilobot, motor driving commands can be sent and sensor data can be reached.

The implementations of this work need complex programming for data processing, networking for data transmission and motor controls. The onboard processors of Trilobots would not be sufficient for these tasks. That is why; new small but powerful PC's; Via Epia EN15000Gs are placed on the robots. This PC can be seen in Figure 7.2

Figure 7.2: The powerful PC on implementation robots

One of the main advantages of Via Epia EN15000G is its small size. This PC is 17cm $\times$17 cm which is quite appropriate considering the size of the Trilobots. Another important feature of this PC is that it consumes very low energy according to similar PCs. EN15000G, has several useful ports like 4 USB2.0 ports, serial port, PS2 mouse port, PS2 keyboard port. Lithium polymer 14.8 V 2200mA rechargeable batteries are used for the robot and the processor instead of the original D-cells batteries. This decreased the load on the robots for a better performance.

Via Epia EN15000G is fully compatible with Microsoft Windows and

Linux operating systems. In the implementations, Windows XP operating system is used. The coding is done with Visual Studio C++. The PCs are connected to the processor of the Trilobot through RS-232 communication from serial port. The data transmission is achieved by a wireless computer-to-computer network which is built using Asus WL-167G usb2.0 wireless LAN adapters which provide wireless network access with IEEE 802.11g protocol.

7.2.2 Vision System

For implementations of the proposed method, the positions of the robots and the desired formation shape should be known implicitly. In our shape formation implementation scenario, there are several mobile robots and there is a target shape which should be formed in a specific coordinate. In this scenario, the target position can be far from the robots that it may not be recognized by their on-board sensors.

A supervision system implemented with a bird-eye-view camera which can see the whole environment is useful for this senario. It is a realistic task definition that the mobile robots are desired to reach a desired formation in a field and some supporting vehicles such as helicopters can send data to the mobile robots about the positions of the desired shape and the robots. This system is implemented by placing a camera on the ceiling of the implementation room.

In the implementation, this bird-eye-view camera is connected to a central computer. The driving commands for robots are produced by this computer. The commands for each robot are sent through wireless connection. For this communication a computer-to-computer connection is used.

In this system, Visual Studio C++ is used for image processing, producing the velocity control commands and sending control commands to the robots in the central unit. OpenCV library [45] is used for image processing to model the desired shape and track the robots. Two signs are placed on each robot to mark its left and right sides. These signs are tracked using a Kalman filter. Using the positions of these two signs, the position and orientation of the robots are calculated.

The wireless data sending is done through a socket programming code. In this socket programming, the central unit works as server and each robot connects to the server as clients. The desired control command is sent as a character array according to a predefined protocol. This protocol is designed for keeping number of sent characters small but sufficient for increasing the communication speed.

The robots listen to the socket constantly and when a data is received from socket, the data is sent to the serial port according to the protocol of Trilobot. Windows XP operating system restrains data sending and receiving through serial port. This problem is solved by using Marshallsoft serial communications component library (WSC4C) [9] with the C++ code. WSC4C allows communicating through serial port using a C/C++, Visual C++, C# or .Net program. The velocity control commands sent to the processor of Trilobot are processed and the motors are controlled accordingly by the processor. In Figure 7.2.2 the connection between the PC and the processor of the robot can be seen in detail.

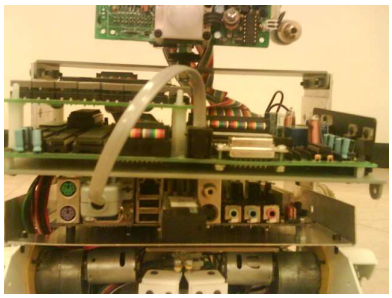

Figure 7.3: RS-232 connection between PC and the processor of Trilobot

7.3 Experiments

In the first part, the implementation will be made using a single robot to see the behaviour of this robot under the shape formation control. Then the implementation will be made on a robot group of 3. In the implementations, a simple circular formation shape is used.

7.3.1 Experiment with One Single Robot

The aim of this experiment is to observe the formation control on a real non-holonomic mobile robot. In this implementation, the robot start from a random position in the workplace, reaches the desired curve and travels around this curve. Some frames from the implementation result can be seen in Figure 7.4.

It is seen that the non-holonomic mobile robot reached the desired curve and traveled around this curve successfully.

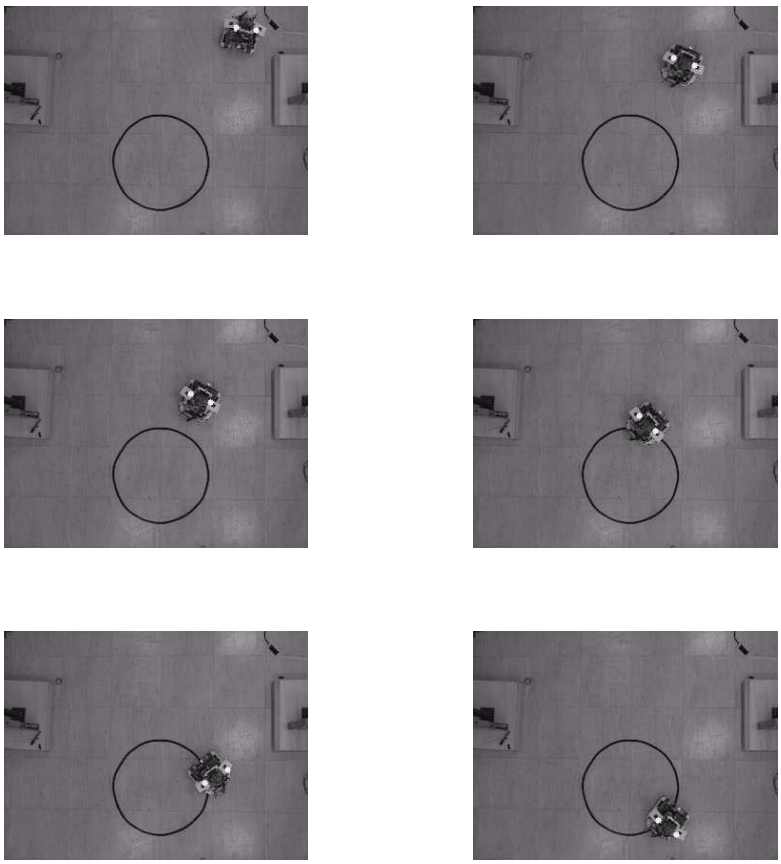

Figure 7.4: Implementation of shape formation with a single robot

7.3.2 Experiment with a Robot Group

The aim of this experiment is to see the effect of the formation control with the coordination control on real robots. Frames from the experiment results can be seen in Figure 7.5.

In the figures it is seen that the robots reached and kept the desired formation shape in coordination. It is seen that even the robots started

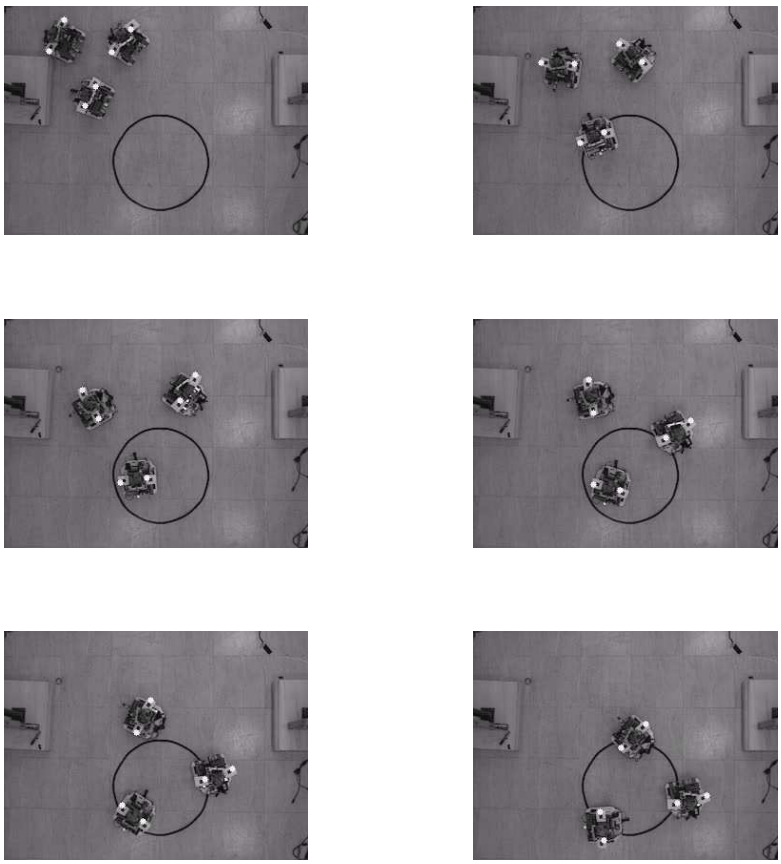

Figure 7.5: Implementation of shape formation with 3 robots

in a positions in which the distance between each other is smaller than the desired, the required distance is achieved. The robots kept this distance until finishing the formation and while keeping the formation.

Chapter 8

Conclusion

In this thesis, a flexible shape formation method which uses elliptic Fourier descriptors and implicit polynomials for representing complex closed curves is presented. The proposed method is applicable for complex formation shapes, groups of different sizes and heteroenous groups. The success of this method is shown through simulations for robots of point-particle and non-holonomic models. This method is also implemented on real non-holonomic mobile robots.

Particularly, in Chapter 2 formation control methods in the literature are summarized. In Chapter 3 the representation of complex closed curves using elliptic Fourier descriptors and implicit polynomials are presented. By the examples, the flexibility of these representations are demonstrated. Chapter 4 was on modeling and trajectory control for non-holonomic mobile robots. A novel method for shape formation control is developed in Chapter 5. The aim of this formation control is to achieve a desired complex formation shape and to keep it while traveling around the formation shape in a coordinated

manner. The coordination aims to keep a desired distance between the robots and their neighbors. In this method, two phases are presented for shape formation. The first phase starts at the beginning of the formation until the robots reach the desired formation shape. In this phase, the implicit polynomial representation of the complex curve is used for producing potential functions. The second phase is employed for the robots to travel around the formation curve after they reach the curve. In this phase, the elliptic Fourier description of the curve is used for the control design. The coordination of the robots is modeled by artificial linear springs between each robot and its nearest two neighbors.

In Chapter 6 simulations are done for robots of particle-point and non-holonomic models. In the first parts of these simulations, a single robot is used to see the success of the shape formation control for a complex shape. In the second parts, multiple robots are used to see the success of the shape formation method with the coordination control. It is seen that the robots have reached and kept the desired formation shape in all of the simulations. In the simulations which multiple robots are used, the desired distance is observed to be kept between the neighbors. In Chapter 7, implementations are done with real non-holonomic mobile robots. In the first implementation single robot is used and the success of the proposed formation control is seen. The second implementation is done with multiple robots to observe the success of the formation control with coordination.

As a future work, obstacle constraints can be added to the presented shape formation control. Also this method can be investigated for 3D shape formation of mobile robots.

Bibliography

[1] V. Kumar J. Ostrowski J. Spletzer C. Taylor A. Das, R. Fierro. A framework for vision based formation control. *IEEE Transactions on Robotics and Automation*, 2001.

[2] M. Mataric A. Howard and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributd, scalable solution to the area coverage problem. *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems*, 2002.

[3] A.S. Morse A. Jadbabaie, J. Lin. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 2953–2958, 2002.

[4] T. Balch and R. Arkin. Behavior-based formation control for multirobotic teams. *IEEE Transactions on Robotics and Automation*, pages 926–934, 1998.

[5] T. Balch and R. C. Arkin. Behavior-based formation control for multiagent robot teams. *IEEE Transactions on Robotics and Automation*, 1999.

[6] T. Balch and M. Hybinette. Social potentials for scalable multi-robot formations. *Proc. of the 2000 IEEE Int. Conference on Robotics and Automation*, pages 73–80, 2000.

[7] R. Brooks. A robust layered control system for a mobile robot. *IEEE J.Robotics and Automation*, page 1423, 1986.

[8] F. Roberti R. Carelli T. F. Bastos-Filho C. C. Gava, R. F. Vassallo. Nonlinear control techniques and omnidirectional vision for team formation on cooperative robotics. *2007 IEEE International Conference on Robotics and Automation*, pages 2409–2414, 2007.

[9] MarshallSoft Computing. available online at http://www.marshallsoft.com/wsc4c.htm (last accessed on July 2008).

[10] S. Chiaverini F. Arrichiello and T. I. Fossen. Formation control of underactuated surface vessels using the null-space-based behavioral control. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5942–5947, 2006.

[11] S. Lacroix G. Hattenberger and R. Alami. Formation flight: Evaluation of autonomous configuration control algorithms. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2628–2633, 2007.

[12] S.S. Sastry R. Murray G. Walsh, D. Tilbury and J.P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, pages 216–222, 1994.

[13] G.H. Granlund. Fourier preprocessing for hand print character recognition. *IEEE Trans. Computers*, pages 195–201, 1972.

[14] N. Gulec. Modeling and control of the coordinated motion of a group of autonomous mobile robots. Master's thesis, Sabanci University, 2005.

[15] M. Unel H. Yalcin and W. Wolovich. Implicitization of parametric curves by matrix annihilation. *International Journal of Computer Vision*, 2003.

[16] G.J. Pappas H.G. Tanner and V. Kumar. Input-to-state stability on formation graphs. *Proc. 41st IEEE Conf. Decision and Control*, pages 2439–2444, 2002.

[17] J. P. Ostrowski J. Desai and V. Kumar. Controlling formations of multiple mobile robots. *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2864–2869, 1998.

[18] D.A. Schneider J. Shen and A.M. Bloch. Controllability and motion planning of multibody systems with nonholonomic constraints. *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 4369–4374, 2003.

[19] P. Johnson and J. Bay. Distributed control of simulated autonomous mobile robot collectives in pay load transportation. *Autonomous Robot*, pages 43–64, 1995.

[20] J. Jongusuk and T. Mita. Tracking control of multiple mobile robots: A case study of inter-robot collision-free problem. *Proc. IEEE International Conference on Robotics and Automation*, pages 2885–2890, 2001.

[21] 0. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *In Proc. IEEE Int. Conf. Robotics and Automation*, 1985.

[22] F.P. Kuhl and C.R. Giardina. Elliptic fourier features of a closed contour. *Computer Graphics and Image Processing*, pages 269–281, 1982.

[23] M. Fields K. Valavanis L. Barnes, W. Alvis and W. Moreno. Heterogeneous swarm formation control using bivariate normal functions to generate potential fields. *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 2006.

[24] G. Bekey L. E. Parker and Eds J. Barhen. Current state of the art in distributed autonomous mobile robotics. *Distributed Autonomous Robotic Systems*, page 312, 2000.

[25] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. *Proc. IEEE Conf. Decision and Control*, pages 2968–2973, 2001.

[26] M.A. Lewis and K.H. Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robot*, pages 387–403, 1997.

[27] M. Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, page 321331, 1995.

[28] K.T. Simsarin M.J. Mataric, M. Nillson. Cooperative multi-robot box-pushing. *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 556–561, 1995.

[29] P. Morin and C. Samson. Practical stabilization of a class of nonlinear systems: Application to chain systems and mobile robots. *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 2989–2994, 2000.

[30] A. Okubo. Dynamical aspects of animal grouping: swarms, schools, flocks and herds. *Advances in Biophysics*, pages 1–94, 1985.

[31] O. A. A. Orqueda and R. Fierro. A vision-based nonlinear decentralized controller for unmanned vehicles. *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006.

[32] O. A. A. Orqueda and R. Fierro. Visual tracking of mobile robots in formation. *Proceedings of the 2007 American Control Conference*, 2007.

[33] J. K. Parrish and W. H. Hammer. *Animal Groups in Three Dimensions*. Cambridge University Press, 1997.

[34] S. Poduri and G. S. Sukhatme. Constrained coverage for mobile sensor networks. *IEEE International Conference on Robotics and Automation*, pages 165–172, 2004.

[35] J. Lawton R. W. Beard and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Trans. Control Syst. Technol.*, pages 777–790, 2001.

[36] J. H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 1999.

[37] C.W. Reynolds. Flocks: herds and schools: A distributed behavioral model. *Proceedings of the 14th annual conference on Computer graphics*, pages 25–34, 1987.

[38] Arrick Robotics. available online at http://www.arrickrobotics.com/trilobot/ (last accessed on July 2008).

[39] T. Koo S. Zilinski and S. Sastry. Optimization-based formation reconfiguration planning for autonomous vehicles. *Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

[40] C. Samson. Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control*, pages 64–77, 1995.

[41] C. Samson. Trajectory tracking for non-holonomic vehicles: overview and case study. *Proceedings of the Fourth International Workshop on Robot Motion and Control*, pages 139–153, 2004.

[42] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1136–1141, 1991.

[43] C. Samson and K. Ait-Abderrahim. Feedback stabilization of a nonholonomic wheeled mobile robot. *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1242–1247, 1991.

[44] P. Song and V. Kumar. A potential field based approach to multi-robot manipulation. *Roc. of the 2002 IEEE Int. Conference on Robotics and Automation*, pages 870–876, 2002.

[45] SourceForge. available online at http://sourceforge.net/projects/opencvlibrary/ (last accessed on July 2008).

[46] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[47] M. A. Bender S. P. Fekete T.-R. Hsiang, E. M. Arkin and J. S. B. Mitchell. Algorithms for rapidly dispersing robot swarms in unknown environments. *Algorithmic Foundations of Robotics*, pages 77–94, 2003.

[48] E. Ben-Jacob I. Cohen T. Vicsek, A. Czirok and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, pages 1226–1229, 1995.

[49] R.L. Frost T.B. Gold, J.K. Archibald. A utility approach to multi-agent coordination. *International Conference on Robotics and Automation, ICRA '00*, pages 2052–2057, 2000.

[50] S. L. Veherencamp. Ahandbook of behavioral neurobiology. *P. Marler and J.G. Vandenbergh*, pages 354–382, 1987.

[51] T. P. Wallace and O. R. Mitchell. Analysis of three-dimensional movement using fourier descriptors. *IEEE Trans. Pattern Analysis ad Machine Intelligence*, pages 583–588, 1980.

[52] P. K. C. Wang. Navigation strategies for multiple autonomous mobile robots moving in formation. *Journal of Robotic Systems*, pages 177–195, 1991.

[53] H. Yamaguchi. A cooperative hunting behavior by multiple nonholonomic mobile robots. *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3347–3352, 1998.

[54] H. Yamaguchi. A cooperative hunting behavior by mobile robot troops. *Intl. J. Robotics Research*, pages 931–940, 1999.

[55] H. Yamaguchi and T. Arai. Distributed and autonomous control method for generating shape of multiple mobile robot group. *International Conference on Intelligent Robots and Systems*, pages 800–807, 1994.

[56] C.T. Zahn and R.Z. Roskies. Fourier descriotors for plane closed curve. *IEEE Trans. Computers*, pages 269–281, 1972.