HELMUT SCHMIDT
UNIVERSITÄT
Universität der Bundeswehr Hamburg

Anh Duc Dang

Robot Formation Control Methodology Based on Artificial Vector Fields

Hamburg 2017

Robot Formation Control Methodology Based on Artificial Vector Fields

To the Faculty of Electrical Engineering of
Helmut Schmidt University / University of the Federal Armed Forces Hamburg
for the attainment of the academic degree of
Doctor of Engineering
submitted

DISSERTATION

by

Anh Duc Dang

from Thai Nguyen, Vietnam

Hamburg 2017

First advisor: Prof. Dr.-Ing. Joachim Horn

Second advisor: Prof. Dr.-Ing. Gerd Scholl

Date of examination: August 15, 2017

Acknowledgments

Firstly, I would like to thank the Vietnamese Government, and specially thank the MOET (Ministry of Education and Training) and the WUS (World University Service-Deutsches Komitee e.V.), who supported me to implement this research.

I would like to express my deepest gratitude to my advisor, Prof. Dr.-Ing. Joachim Horn, for his guidance, patience and continuous support in the completion of my Ph.D. I would also like to thank Prof. Dr.-Ing. Gerd Scholl for his tutelage and the extra support given during the research studies towards the completion of my Ph.D and Prof. Dr.-Ing. Stefan Dickmann for serving as the chair of the Examination Commission.

I would like to thank all my colleagues for their valuable comments, discussions, suggestions and for a pleasant working atmosphere created during my studies at the Helmut-Schmidt-University. In addition, I would like to express my sincere thanks to Dr.-Ing. Klaus Frick at the Helmut-Schmidt-University, who was always willing to help and give best suggestions.

I would like to thank all my friends for their cheerful friendship, encouragements, assistance, and their extramural support these past years.

Finally, I would like to specifically thank my family for always encouraging and supporting me throughout my journey these past years. Last but not least, I would like to thank my wife, Hoang Anh Phung, for her love and encouragement.

Hamburg, August 2017

Anh Duc Dang

Abstract

Formation control has been one of the important topics covered in the researches on the multi-agent systems. The applications of the multi-agent systems are significant in variety of tasks such as search and rescue missions, forest fire detection, reconnaissance, surveillance, etc. Inspired by the cooperative ability as well as the intelligence of natural animal groups such as schools of fishes, flocks of birds, swarm of ants, etc., this dissertation develops the artificial vector field method for formation control of autonomous robots while tracking one or more moving targets in a dynamic environment.

In our approach, the proposed artificial vector fields, which consist of the attractive, repulsive, and rotational force field, are combined with the damping term in the formation control laws in order to control the velocity, heading, connectivity, as well as the obstacle avoidance of a swarm of autonomous robots while in motion. Using this approach, autonomous robots are not only controlled to move along a desired trajectory towards the target, but are also held in a specified formation without collisions during movement. In other words, under the effects of the proposed artificial vector fields, the member robots of a swarm will move together in a specified formation with the velocity matching, without collisions among them while tracking the target. In addition, the free robots will themselves approach the created formation from their swarm in order to obtain the fixed position in this formation. Especially, the thesis then explains that by using the proposed hybrid force field in the obstacle avoiding controller, the local minima problems that still exist in the traditional potential field method (for example, when a robot is trapped in U-shape obstacle, etc.) will be solved. In the proposed hybrid force field, the local repulsive force field surrounding obstacles, which is stronger when the robot is closer to the obstacles, is utilized to repel the robot away from the obstacles, while a local rotational force field is added to surround the obstacles in order to drive robot to escape the obstacles in the direction of the target's trajectory. Therefore, robots can easily and quickly avoid obstacles, as well as escape complex obstacles along their moving trajectory in order to complete the assigned tasks with their swarm.

The thesis focuses on two main issues in formation control, namely, (i) formation control following the desired formations and (ii) cooperative formation control. The first issue concerns how robots are controlled by the proposed formation control algorithm in order to approach the coordinated virtual nodes in the desired formation (for example, V-shape, line or circular shape), and to maintain following these virtual nodes during tracking; while the second issue showcases the use of the proposed cooperative formation control law, where robots will automatically cooperate with each other in their neighboring relationship in order to generate and maintain the cohesion in their formation.

Table of Contents

Symbols

p_i	Position of robot *i*
v_i	Velocity of robot *i*
p_k	Position of robot *k*
v_k	Velocity of robot *k*
p_o	Position of obstacle
v_o	Velocity of obstacle
p_l	Position of leader
v_o	Velocity of obstacle *o*
p_l	Position of leader
v_l	Velocity of leader
p_j	Position of virtual node *j*
v_j	Velocity of virtual node *j*
p_c	Center position of swarm
v_c	Velocity of swarm's center
p_t	Position of target
v_t	Velocity of target
n_i^k	Unit vector between robot *i* and robot *k*
n_i^o	Unit vector between robot *i* and obstacle *o*
n_i^l	Unit vector between robot *i* and leader
n_i^j	Unit vector between robot *i* and virtual node *j*
n_i^c	Unit vector between robot *i* and virtual swarm's center
n_i^t	Unit vector between robot *i* and target
n_l^t	Unit vector between leader and target
$(p_i - p_k)$	Relative position between robot *i* and *k*
$\|p_i - p_k\|$	Distance between robot *i* and *k*
$(v_i - v_k)$	Relative velocity between robot *i* and *k*

$\left(p_i - p_t\right)$	Relative position between robot *i* and target
$\left\|p_i - p_t\right\|$	Distance between robot *i* and target
$\left(v_i - v_t\right)$	Relative velocity between robot *i* and target
$\left(p_i - p_o\right)$	Relative position between robot *i* and obstacle *i*
$\left\|p_i - p_o\right\|$	Distance between robot *i* and obstacle *i*
$\left(v_i - v_o\right)$	Relative velocity between robot *i* and obstacle *i*
$\left(p_i - p_c\right)$	Relative position between robot *i* and swarm's center
$\left\|p_i - p_c\right\|$	Distance between robot *i* and swarm's center
$\left(v_i - v_c\right)$	Relative velocity between robot *i* and swarm's center
$\left(p_i - p_l\right)$	Relative position between robot *i* and leader
$\left\|p_i - p_l\right\|$	Distance between robot *i* and leader
$\left(v_i - v_l\right)$	Relative velocity between robot *i* and leader
$\left(p_i - p_j\right)$	Relative position between robot *i* and virtual node *j*
$\left\|p_i - p_j\right\|$	Distance between robot *i* and virtual node *j*
$\left(v_i - v_j\right)$	Relative velocity between robot *i* and virtual node *j*
$\left(p_l - p_t\right)$	Relative position between leader and target
$\left\|p_l - p_t\right\|$	Distance between leader and target
$\left(v_l - v_t\right)$	Relative velocity between leader and target
N	Number of robots
M	Number of obstacles
$N_i^k(t)$	Set of robots in the neighborhood of robot *i* at time *t*
$N_i^o(t)$	Set of obstacles in the neighborhood of robot *i* at

1 Introduction

1.1 Motivation

In recent years, researches on multi-agent systems have widely been conducted in physic [6, 7, 8], chemistry [9, 10], biology [11], and especially in control and cybernetic science [24]-[94] over the world. Formation control is one of the necessary and important problems in the research field on the multi-agent systems. The formation control of autonomous robots, such as the formation of autonomous underwater vehicles [21, 84], unmanned aerial vehicles [29, 30, 31, 83], flocking control [34]-[42], mobile sensor networks [43]-[57], etc., and its potential applications into a variety of tasks including search and rescue missions, forest fire detection, reconnaissance, and surveillance, etc., have attracted a lot of attention from researchers worldwide.

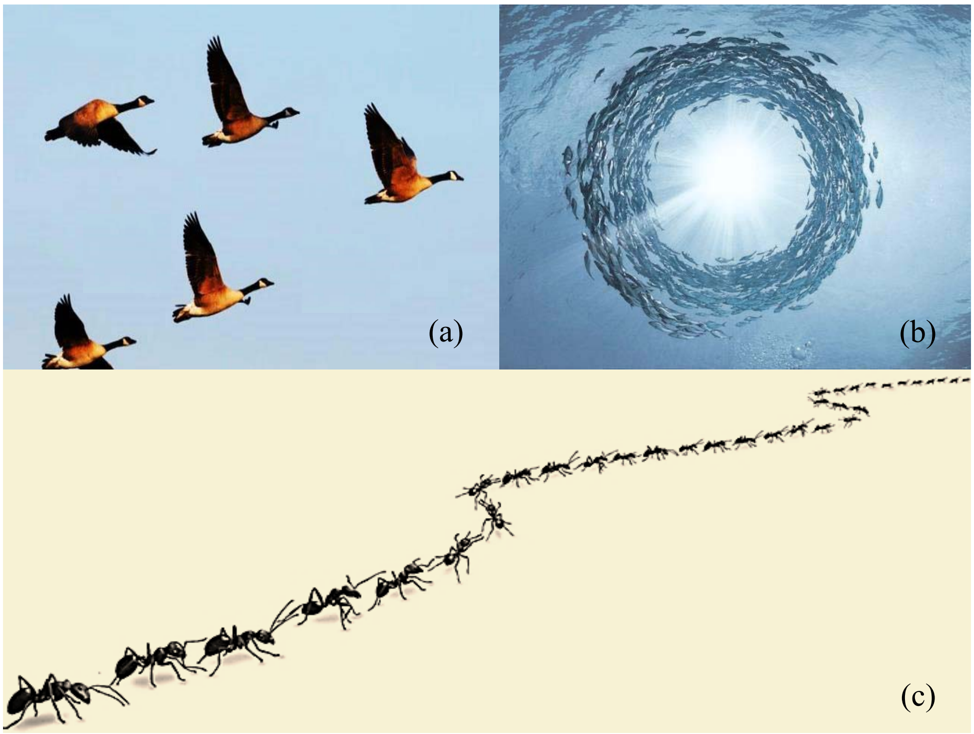

Figure 1.1: Examples of the special swarms in nature: the V-shape flying formation of birds (a) (Source: http://www.grahamowengallery.com/photography/geese.html), the circular formation of fish (b) (Source: http://www.simontuckett.com/_Portfolio/PortPages_Hi /Il_FishSchool.html), the collinear formation of ants (c) (Source: http://www.e-swarm.org / images/ant-trail.png).

In observing the animal groups in nature, such as schools of fish, flocks of birds, swarm of ants, etc., we can see the intelligences as well as the several interesting features of these animal groups. Moreover, from the natural animal groups we can also find some special formations that are organized into the particular shapes such as V-shape, line or circle, etc., see figure 1.1. Natural features as well as the intelligences of a swarm [1]-[14] that have suggested our designing the control laws for the multi-agent systems, are expressed as follows:

- A swarm can still search and track a source of food (a target) efficiently while avoiding obstacles. This natural phenomenon helps us to design the target tracking control algorithm for a swarm of autonomous robots in a dynamic environment.
- A swarm can also split into smaller sub-groups in order to search and approach multiple food sources (targets) and avoid obstacles simultaneously. If any food source runs out, the individuals that are using this food source will continue to search and approach other food sources. This situation directs us to the building of sensor merging and splitting in a mobile sensor network when the number of the targets changes. This control law guarantees that all free sensors can easily and quickly approach their swarm as well as some sensors will be split from a main group into a sub-group to track a new target.
- A swarm has the ability to change its size to adapt to the environment. Based on this feature, we can design the adaptive formation control algorithm for a swarm of autonomous robots while escaping obstacles to track a moving target. Using this control law, the swarm's size shrinks into the smaller size in order to adapt to the complex environment while maintaining the formation connectivity.
- Each member in a swarm can communicate and interact with its neighbors within its limited sensing range in order to avoid the collisions with its neighbors, and generate the robust connectivity in its formation. Moreover, based on this organization, the information that each member obtains from the environment can be sent to all other members in formation. Hence, in cases where only some members in the swarm detect the obstacles or the food source, etc., but all other members in the swarm can still avoid these obstacles, or approach this food source with their swarm. This natural feature motivates us as researchers to design the formation cooperative control algorithm. This control algorithm guarantees that the formation of a swarm is maintained without collisions among the members in the swarm while tracking the dynamic target.

- Each member in a swarm can still combine and move with its neighbors in the cohesive formation, although it may not sense the position as well as the velocity of its neighbors accurately. This natural feature has inspired the authors to think about the formation control of a swarm of autonomous robots in the noisy environments.
- A swarm can move in the particular formations such as V-shape, line or circle, etc., see figure 1.1. This special ability encourages us to study the formation control of a swarm of multiple autonomous robots following the desired formations. If successful, this control law will ensure that multiple autonomous robots can uphold a specific formation while traversing a trajectory and avoiding collisions among them simultaneously. The potential applications of this approach in the specific tasks or missions such as search, observation, supervision or surrounding, etc., play the important roles in reality.

Finally, motived by the features, the abilities, and the intelligences of the animal groups in the nature with the potential applications from multi-agent systems in reality, this thesis focuses on the research and the design of the control algorithms for multi-agent systems, such as: formation connection control; path planning for the formation of a swarm in a dynamic environment; formation control of autonomous robots in a noisy environment; adaptive and cooperative formation control in complex environments, etc.

1.2 Problem statement

Although there are many research directions on the multi-robot systems, but the main issue is that the member robots in a group have to collaborate in order to achieve the desired tasks, such as observing, tracking or encircular a moving target, etc. As presented in the preceding passages, formation control of autonomous robots is observed in various kinds of the animal groups in the nature, such as schools of fishes, flocks of birds, swarm of ants, etc., [1]–[7]. This guarantees that the members in the formation of a group have to move together in a desired organization under the velocity matching without collisions among them. In addition, the free robots in a group, which still do not have the cohesion with their formation, must themselves approach their formation in order to obtain the fixed position in their formation. Moreover, the formation of a swarm must be maintained while moving in a free environment as well as in a dynamic environment. All of these analyses have identified our research concentration issues as follows:

- "Formation control following the desired formations", the content of this issue is to control the formation of autonomous robots following the particular shapes such as V-shape, line or circle, etc., while observing and tracking the dynamic target. In this method, the desired formation, which consists of the coordinated virtual nodes, is first generated. The shape of the desired formation is decided on its potential application in the specific tasks or missions such as search, observation, supervision, tracking or surrounding, etc. Then, the robots are controlled to approach the coordinated virtual nodes in the desired formation and to maintain at these virtual nodes while moving simultaneously. Using this method, robots can easily uphold a specific formation while traversing a trajectory and avoiding collisions among them simultaneously. Although this topic is very interesting and important in the field of the research on the multi-robot systems, but the research results in this field are still very limited. Therefore, this research direction will be developed in this thesis, such as the adaptive shape-formation control while tracking a moving target under the influences of the varying environments.

- "Cooperative formation control", the content of this issue is to control the formation of a group of autonomous robots based on the stable and robust connections among neighboring members to complete the specific tasks or missions such as search, observation, supervision or tracking the dynamic targets. In this method, the neighboring robots are linked to each other by the attractive/repulsive force fields among them to create a robust cohesion in a formation and avoid collisions among them simultaneously. The success of the cooperative formation control method based on the connections among neighboring members in a swarm as an α-lattice configuration has been published in some literature, such as flocking control [34]-[42]. The published results show that this topic is very interesting, and has potential applications in military areas as well as in civilian areas. However, there are some constraints on the approach, such as: when the cohesion in the formation of the swarm is broken while avoiding obstacles, while we need the maintaining of the formation in order to perform a certain job. Moreover, the formation adaptation of a swarm in the complex environments such as U-shape or the narrow space between obstacles, etc., is also an important issue that needs attention. Additionally, the thesis addresses other arising issues namely the swarm–finding of the free robots that have not been offered a fixed position in a formation (for example: the robots that get lost, or are separated from their formation for certain cause).

1.3 Method of approach

There are many methods to generate and control the formation of a swarm of autonomous mobile robots. One of the simple and effective methods utilized to control the coordination, the motion, the formation connectivity, as well as the obstacle avoidance for a swarm in order to track the dynamic targets is the artificial vector field tool. This vector field tool is built on the relative positions and velocities among the targets, robots, and obstacles in the environment.

In this thesis, the proposed solution for formation control of autonomous robots is based on the improved vector fields that consist of the artificial potential fields and the artificial rotational fields. The proposed potential fields, which include the attractive and repulsive force fields and are generated by negative gradient of the corresponding potential functions, are not only used to control autonomous robots moving on a desired trajectory (path planning for a swarm), but are also used to hold these robots in a specified formation without collisions during movement (formation connection control). The attractive force field is generated surrounding the goals such as the target of the swarm, the virtual nodes in the desired formation, etc., to drive robots towards these goals (for example the target tracking control or the swarm-finding control, etc.,) with the velocity matching. Furthermore, surrounding the neighboring robots in a formation, the local attractive force field is combined with the local repulsive force field in order to create and keep the formation connectivity, as well as to avoid collisions among the members of the swarm. Moreover, using the hybrid force field, which consists of the repulsive force field and the rotational force field surrounding obstacles, robots can easily and quickly avoid and escape obstacles along their moving trajectory while maintaining their formation. In this obstacle avoiding control law, the repulsive force field that is stronger when the robot is closer to the obstacles is utilized to repel the robot away from the obstacle, while a rotational force field is added to drive the robot to escape the obstacles in the direction of the target's trajectory. Especially, this added rotational force field will solve the local minima problems in the traditional potential field method such as when robots are trapped in the complex obstacles (for example U-shape obstacle, etc.), [22], [24].

Finally, the proposed artificial vector fields combined with the damping term in the formation controller will generate the desired velocities and headings, as well as the stable formation connectivity for the robots of a swarm while tracking the dynamic targets. Moreover our proposed formation controller also guarantees that the formation adaptation in different tasks and missions, and in varied environments is performed.

1.4 Research contributions

As presented above, “formation control following the desired formations” and “cooperative formation control” are the main contributions of our research in this dissertation. As such, the research is expected to indicate:

- **Formation control following the desired formations**

  The content of this contribution is to control the formation of autonomous robots following the desired shapes including V-shape, line and circle, while observing and tracking a dynamic target under the influence of the dynamic environment. In this approach, the desired formations, which consists of the equidistant neighboring virtual nodes, is first generated based on the relative position between the swarm’s virtual leader and the target. The swarm’s virtual leader is randomly chosen from a swarm’s member that is closest to the target. This leader plays an important role to create and lead its formation moving towards the target in a stable direction. Hence, in many undesired cases, such as the leader is broken or trapped in the obstacles (for example U-shape obstacle) one new leader is replaced in order to continue to lead the swarm tracking the target. Further, the motion of the robots towards the desired positions in the desired formation is controlled by the artificial force fields that consist of the local and global potential fields surrounding the virtual nodes. Under the effect of these artificial force fields, robots will automatically find their desired position at the virtual nodes in the desired formation. Additionally, the local repulsive force field surrounding each robot is used to avoid the collision with each other. Moreover, using the repulsive vector field combined with the rotational vector field in the obstacle avoiding controller, robots can easily escape the obstacles while tracking a moving target.

  In summary, this main contribution is performed as follow: firstly, the desired formation (V-shape, line or circular shape) of the virtual nodes, which are equally spaced, is designed on the relative position between the swarm’s virtual leader and the target. Secondly, the control algorithms, which are developed based on the artificial vector fields, guarantee that the motion of robots always converges to the created virtual nodes in the desired formation under the effects of the dynamic environment and avoid collisions simultaneously. Furthermore, the formation adaptation and the target approaching direction are decided by the target’s position sense of the virtual leader. Using our proposed control algorithms, all robots can easily approach the desired formation and maintain their formation connectively while tracking a moving target in a dynamic environment.

- **Cooperative formation control**

The content of this contribution is to control the formation of a group of autonomous robots based on the stable and robust connections among the neighboring members in order to track the dynamic targets in the various environments. In this approach, the neighboring robots are first linked to each other by the artificial attractive/repulsive force fields among them in order to create a robust cohesion in a formation and avoid collisions among them simultaneously. Further, in order to adapt to the changing environments, for example in the case when a group of multi robots must pass through a narrow space among obstacles, we propose the adaptive formation control algorithm. Using this control algorithm, the members in a formation can cooperatively learn the swarm's parameters to decide the size of their formation so that the formation connection and the target tracking performance can be improved. Furthermore, in order to solve the local minima problems in the traditional potential field method as presented in [20, 21, 22], and help robots find a good way to approach the moving target while maintaining their formation simultaneously, we propose the developed obstacle avoiding controller. In this proposed obstacle avoiding controller we also utilize the repulsive/ rotational force fields combined with the direction of the target's trajectory to drive the robot formation so as to escape the obstacle. Moreover, in order to solve the problems of the robot contribution and distribution in the scenario of multiple dynamic target tracking, we use the robot splitting/merging control algorithm. Using our control law, the free robots can easily approach their formation, as well as some member robots from a main group can be split into smaller subgroup.

Finally, this main contribution is built as follows: the formation's connectivity is first developed on the local attractive and repulsive force fields among the neighboring members of a group. Furthermore, the cooperation control algorithm for the formation maintenance of a group while avoiding obstacles is built on the data that are collected and updated by the sensing of each member about the environment around it.

1.5 Literature review

In this section, the current formation control methods, which will be evaluated and compared with our proposed approach, are reviewed. The topic of formation control of a group has been introduced extensively in the literatures with its applications to path planning or navigation for autonomous mobile robots [23]-[28], autonomous underwater vehicles (AUVs) [21, 84], unmanned aerial vehicles (UAVs) [29, 30, 31, 83]. Formations, which are discussed and compared in this section, include both cooperative formations and specific geometric formations. The different control topologies are designed and utilized for specific formations. An overview of previous and relevant studies of the topic is presented below as a framework that comprises the main focus of the research.

1.5.1 Specific geometric formation control

In this subsection, we review existing works in the area of specific geometric formation control [82]-[95] including virtual structure formation control, dynamic region following formation control, formation control following desired shapes.

Firstly, in the virtual structure or a framework formation control method [82]-[88] the formation of a group of robots is designed as a rigid structure or a fixed framework, in which the agents are assumed that they do not connect to each other, but they can move together in a continuous space. The concept of the virtual structure or a fixed framework has been introduced in [82]. Results from this literature have shown that this approach is capable of achieving high precision movement which is fault tolerant and exhibits graceful degradation of performance. In addition, this proposed algorithm does not require the leader selection as in other cooperative robotic strategies. Moreover, in [83], a dynamic virtual structure formation control scheme is also proposed to enable a formation of fixed-wing UAVs to execute coordinated formation manoeuvres along a planned formation trajectory. In [85], the authors described a framework for controlling and coordinating a group of mobile robots for cooperative tasks ranging from scouting and reconnaissance to distributed manipulation. This approach shows that the proposed control algorithm to composition also guarantees stability and convergence in a wide range of tasks. The main advantage of the virtual structure method for formation control is that it is simple to generate a rigid formation structure. The members in the formation can still move together along a specified linear trajectory while maintaining a rigid geometric relationship among them, although there is no the leadership in the formation. The main limitation of this approach is that the rigidity of the virtual structure or a fixed framework affects the formation's turning performance when moving along a curvature trajectory.

Another approach for the specific geometric formation control is to use a dynamic region. In this method, all member robots of a group are controlled to move together in a given dynamic region [89], [90]. The published results in this approach have shown that robots stay within a moving region, and are able to adjust their formation by rotating and scaling during movement together simultaneously. This method does not require specific ordering or positioning of the robots inside the given dynamic region. The motion of each robot in formation depends on the motion of its neighbors and the given dynamic region.

One positive approach, which has also attended the growing attention from researchers worldwide, for the specific geometric formation control is the formation control following desired shapes [91]-[95]. In this approach, robots are independently controlled to move towards the virtual nodes in the designed desired formation (V-shape, line, or circle) and converge at these virtual nodes simultaneously during movement. Moreover, the designed virtual nodes in the desired formation must guarantee that the robots that are occupying the neighboring virtual nodes do not interact with each other.

1.5.2 Cooperative control

Cooperative control is also known as an interesting research direction in multi robot systems. This research direction has received a lot of attention from researchers in recent years [32]-[76]. It has been used for a variety of application fields, however, in this section we only review the existing works that relate to our research in this thesis, including: cooperative control while tracking a moving target; cooperative control in the dynamic targets tracking; adaptive flocking control in a dynamic environment.

Firstly, cooperative control of multi robot while tracking a moving target [32]-[63] can be split into the different research sub-issues based on the knowledge of graph theory, potential field, network communication, and system stability analysis.

In [32]-[37], the approach for the flocking control of multi agent with a virtual leader is introduced. In this approach, the organization of a group and its motion depend on the position, velocity, and trajectory of the leader. In [32, 33], the leader robot of a group tracks a predefined trajectory, and the other member robots in the group will follow this leader while maintaining a particular formation. Further, a method, which is built based on the extension of the flocking control algorithm in [41], for flocking control of multi-agents with a virtual leader was also presented in [34]-[36]. In addition, in [37], the authors investigated the dynamic properties of the group for the case where the state of the virtual leader may be time-varying and the topology of the neighboring relations among agents is dynamic. Although this approach is simple, but its main disadvantage is that the group's motion is dependent on the leader, so any failures from the leader will influence on the motion of the whole system.

In [38]-[62], the approach for the flocking control of multi agent without any leader is presented. In this approach, all robots of a group are controlled together to achieve a target, in other words, all these robots take a similar role while in motion. Each agent of a group will connect with other agents in its communicating range while it moves towards the target position. Thus, the formation's cohesion of a group will be automatically generated on the local connectivity among the neighboring agents in group. Using this control method the stability and robustness of a formation are attained [77]-[81], but the quick swarm-approach of the free agents in the global workspace is still limited. In [39, 40, 41, 49, 50, 51], the algorithm for the quick swarm-approach of the free agents utilizes the parabolic attractive potential function, so the corresponding attractive force will converge linearly towards zero as the free agents will approach the target position with their decreasing velocity. These have demonstrated that the cooperation among the free agents in the local workspace has been well maintained. However, in the global workspace when the free agents that are far from the target will be acted by a large attractive force from the target, thus they will move with the very high speed to the target. In addition, in [50], a distributed flocking algorithm for mobile sensor network to track a moving target is also presented. In this literature, the author used an extension of a distributed Kalman filtering algorithm for the sensors to estimate the target's position. Moreover, the cooperative sensing and learning in a mobile sensor network have been studied by many researchers in recent years. The cooperative sensing in a mobile sensor network can be applied in many different fields such as target tracking, monitoring, observation, or environmental mapping, etc., and can be found in [51, 56, 57, 58,]. The approaches to the cooperative learning in a mobile sensor network including game theory, evolutionary computation or reinforcement learning, etc., are introduced in [54, 60, 86]. The published results around this topic show that the problems of the environment mapping as well as estimation based on the multi agent cooperative sensing and learning are very interesting and still open research directions.

Secondly, cooperative control in the dynamic targets tracking is presented in [64]-[70]. In [64], the problem of motion planning and sensor assignment strategies for tracking multiple targets with a mobile sensor network is discussed. The authors focused on triangulation based tracking where two sensors merge their measurements in order to estimate the position of a target. Further, in [66], robots equipped with sensors and communication devices discover and track as many evasive targets as possible in an open region. Additionally, a technique for multiple moving objects tracking with a mobile robot is discussed in [68]. In this approach, the authors have introduced a sample-based variant of joint probabilistic data association filters to track features originating from individual objects, and to solve the correspondence problem between the detected features and the filters. Moreover, the control algorithms for the dynamic targets tracking in a mobile sensor network are also discussed in [69, 70].

Thirdly, adaptive flocking control in a dynamic environment [71]-[76] is also an interesting issue in cooperative control. In [71], the authors presented a distributed approach for adaptive flocking of the swarms of mobile robots that enables the robots to navigate autonomously in complex environments populated with obstacles. In this approach, an integrated algorithm that allows a swarm of robots to navigate in a coordinated manner, split into multiple swarms, or merge with other swarms according to the environment conditions is proposed. However, the problems for controlling the size of a group were not considered in this literature. In [75], four novel collision avoidance processes for mobile robots to generate effective collision-free trajectories when forming and maintaining a formation are discussed. In addition, robust adaptive flocking control of multi-agent systems with nonlinear dynamics is introduced in [76]. In this method, the coupling weights, which are perturbed by asymmetric uncertain parameters, are dynamically updated while the network topology for velocity is fixed.

1.6 Organization of this dissertation

The rest of this dissertation is formed by separated chapters, in which:

Chapter 2 introduces the approach to path planning for a single mobile robot in a dynamic environment. This approach is based on the artificial vector fields that include the developed potential force fields and the proposed rotational force field.

Chapter 3 presents the approach for formation control of the autonomous robots following the desired formations while tracking a moving target under the influence of the dynamic environment.

Chapter 4 proposes the cooperative formation control algorithms for a group of multi robots while avoiding obstacles as well as tracking a moving target.

Chapter 5 presents the sensor merging/spitting algorithm for a mobile sensor network while tracking the moving targets in a dynamic environment.

Chapter 6 outlines the conclusion drawn from the findings, implications for practice as well as recommendations for future researches.

2 Path Planning For a Single Robot

2.1 Introduction

The problem of path planning for an autonomous robot in a dynamic environment is how to plan and control the robot to move towards the target position in a desired path while avoiding obstacles in the environment. Therefore, navigation or path planning for the autonomous mobile robot is one of the most important applications for robot control systems. This interesting topic has attracted the attention from researchers in recent years.

Obstacle avoidance is an important issue in path planning for autonomous robots while reaching the target position. The artificial potential field as shown in [14]-[23] is known as a positive method in order to solve this problem. Recently, the potential field method has been widely studied and applied powerfully to path planning or navigation for autonomous mobile robot to reach the position of the target in a simple environment, see [14]-[20]. However, in a complex environment, in which the U-shaped obstacles or connected walls exist, the application of the potential field method to path planning for the autonomous robots is very difficult. Robot can be trapped in these obstacles before reaching the target position.

This chapter presents a novel improved artificial vector field method (AVFM) for path planning for a single robot to reach a target, which can be stationary or dynamic, in a dynamic environment. This approach is developed based on the traditional potential field method combined with the rotational field method. Using this approach, the robot can easily avoid and escape the obstacles in the environment without collision while reaching a stationary target. Furthermore, when the target moves in an unknown environment, the obstacle avoiding direction of the robot has a great influence on finding the fastest way towards the target. Following approach, a global attractive force field, which is built surrounding the target, is used to drive the robot towards the target. On the other hand, the repulsive force field, which is stronger when the robot is closer to the obstacle, is also designed surrounding the obstacles to repel the robot away from the obstacle. This repulsive force helps the robot to avoid the collision with the obstacles. Moreover, the rotational vector field is added to control the robot to avoid and escape the obstacles in the direction of the target's trajectory. Under the effect of this blended vector field, the robot can easily escape the obstacles to continue tracking the target. However, in order for the robot to quickly exit the complex obstacles, the computed rotational force must be larger than the sum of the repulsive forces of the obstacles and the attractive force of the target. The direction of the target's movement is determined on the relative position between the current position and the future position of the target with the preselected time-step Δt.

The rest of this chapter is organized as follows: The basic potential field method is reviewed in section 2.2. Then, from the basic potential fields we propose the control algorithms for path planning of a single robot while tracking a moving target under the effects of the dynamic environment in section 2.3. The effectiveness of the proposed approach is verified in simulations in section 2.4. Finally, section 2.5 summarizes this chapter.

2.2 Background

This subsection presents the background of the artificial vector field method that will be extended, and applied to path planning for an autonomous robot in a dynamic environment. Firstly, the idea for this method is presented in subsection 2.2.1. Then, the traditional potential fields are introduced as background for the artificial vector field method in subsection 2.2.2.

2.2.1 Idea of the artificial vector field method

The idea of the artificial vector field method is based on Newton's law of universal gravitation [98], [99]. According to Newton's law of universal gravitation, any two objects in space exert gravitational forces on each other along the line connecting the centers of these objects. These gravitational forces are attractive, equal in magnitude, and have opposite directions. The magnitude of these gravitational forces is proportional to the product of the masses of these objects and inversely proportional to the square of the distance between them, see figure 2.1.

Consider the gravitational force of a point mass m_2 located at the position $p_2 = (x_2, y_2, z_2)^T$ acting on a point mass m_1 located at the position $p_1 = (x_1, y_1, z_1)^T$. As presented in [98], [99], this gravitational force is given by

$$f_{12} = -G\frac{m_1 m_2}{\|r\|^2}\left(\frac{r}{\|r\|}\right) = -G\frac{m_1 m_2}{\|r\|^2} n_r. \tag{2.1}$$

Where G is the gravitational constant, and $n_r = r/\|r\|$ is the unit vector along the direction from the point mass m_2 to the point mass m_1. The relative position vector and the distance between these point masses are described as follows: $r=(p_1-p_2)$ and $\|r\| = \|p_1-p_2\| = = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$, respectively. As shown in [98], [99], the gravitational force (2.1) is a potential field, which is a negative gradient field of a scalar function $V_{12} = -Gm_1m_2/\|r\|$. Hence, equation (2.1) can rewritten as follows:

$$f_{12} = -\nabla V_{12}. \tag{2.2}$$

The proof of this algorithm is given in the Appendix. In contrast, we have $f_{21} = -f_{12}$. For more details of this consensus design please refer to [98], [99].

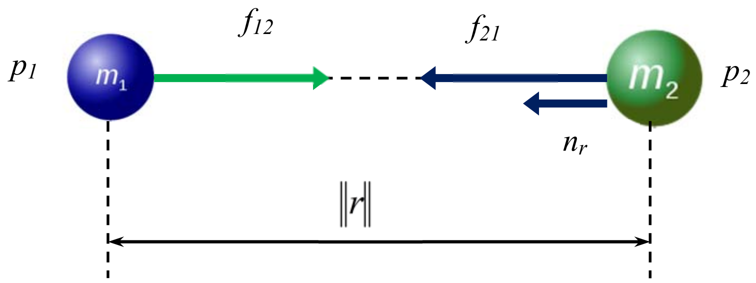

Figure 2.1: Description of Newton's law of universal gravitation [98], [99]

Based on Newton's law of universal gravitation, we design and apply the artificial force fields to path planning for the autonomous robots in a dynamic environment in the next sections.

2.2.2 Potential field method

The artificial potential field is known in control technology as an effective method for robot's path planning. This potential field is the combination of the attractive force field to the target and the repulsive force fields away from the obstacles. In order to generate these control forces, some literatures such as [14]-[23] gave the method by using the negative gradient of the respective attractive/repulsive potential functions.

In this subsection, we review the basic vector fields used in the traditional potential field method [14]-[23]. In this traditional potential field method, the target is surrounded by a global attractive potential field that produces the attractive force on the robot. Under the action of the attractive force from the target, the robot will move towards the target position along the direction of this attractive force. On the other hand, in order to avoid the collision with the obstacles, the local repulsive potential field, which produces the repulsive force on the robot, is built surrounding each obstacle. Therefore, the robot's motion is driven towards the target position by a total force field that consists of the attractive force from the target combined with the sum of the repulsive forces from the obstacles in the environment. The robot's motion is depicted in figure 2.2.

As shown in figure 2.2, a mobile robot is controlled to reach the target while avoiding the obstacle. We assume that the robot and target are as the moving point masses, and they move in two-dimensional (2-D) space. The robot's position and the target's position are denoted by $p_r = (x_r, y_r)^T$, $p_t = (x_t, y_t)^T$, respectively. The obstacle's position $p_o = (x_o, y_o)^T$ is denoted as a point mass on the obstacle such that the distance between this point and the robot is minimal.

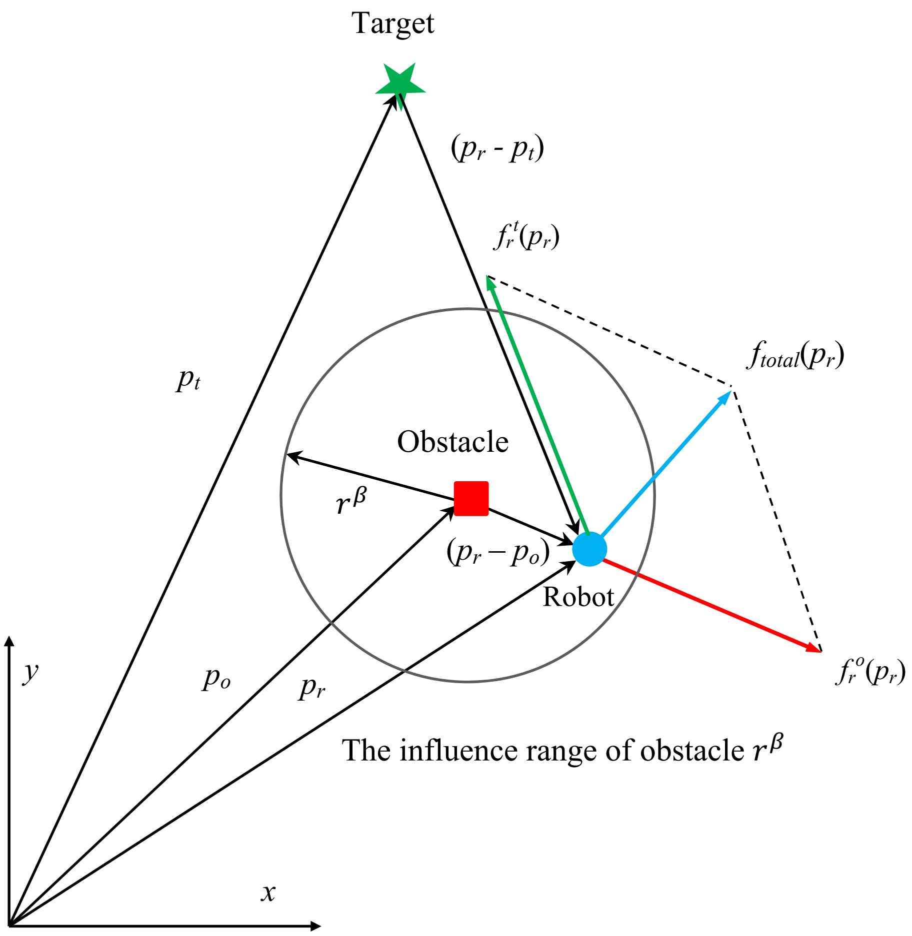

Figure 2.2: Description of the schematic diagram of forces exerting on a mobile robot: the attractive force to the target $f_r^t(p_r)$, the repulsive force from the obstacle $f_r^o(p_r)$ and the robot control force $f_{total}(p_r)$.

A. Attractive potential field

As shown in [15]-[23], the most commonly used attractive potential function is given as follows

$$V_{att}(p_r) = \frac{1}{2} k^t \rho^m (p_r, p_t). \tag{2.3}$$

The corresponding attractive force field is given by the negative gradient of the potential function (2.3) as follows

$$\begin{aligned} f_r^t(p_r) &= -\nabla V_{att}(p_r) \\ &= \begin{cases} -k^t \dfrac{(p_r - p_t)}{\rho(p_r, p_t)}, & \text{if } m = 1 \\ -k^t (p_r - p_t), & \text{if } m = 2. \end{cases} \end{aligned} \tag{2.4}$$

Where k^t is a positive scaling factor, and *m=1* or 2. $\rho(p_r, p_t) = \|p_r - p_t\|$ and $(p_r - p_t)$ are the distance and the relative position vector between the robot and the target, respectively. The equation (2.4) shows that: For *m=1*, the attractive potential is conic in shape, and the corresponding attractive force has constant amplitude. For *m=2*, the attractive potential is parabolic in shape, and the corresponding attractive force converges linearly towards zero as the robot approaches the target.

B. Repulsive potential field

Similarly, one commonly used repulsive potential function is given in [15]-[23] as

$$V_{rep}(p_r) = \begin{cases} \dfrac{1}{2} k^o \left(\dfrac{1}{\rho(p_r, p_o)} - \dfrac{1}{r^\beta} \right)^2, & \rho(p_r, p_o) \le r^\beta \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

Taking the negative gradient of the potential function (2.5), we obtain the corresponding attractive force as follows

$$f_r^o(p_r) = -\nabla V_{rep}(p_r)$$
$$= \begin{cases} k^o \left(\dfrac{1}{\rho(p_r, p_o)} - \dfrac{1}{r^\beta} \right) \dfrac{1}{\rho^2(p_r, p_o)} n_r^o, & \rho(p_r, p_o) \le r^\beta \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

Where k^o is a positive constant, $r^\beta > 0$ is the influence range of the obstacle, and $n_r^o = (p_r - p_o) / \rho(p_r, p_o)$ is a unit vector along the direction from the obstacle to the robot. The magnitude of the relative position vector $(p_r - p_o)$ between the robot and the obstacle is $\rho(p_r, p_o) = \| p_r - p_o \|$, which is the distance from the robot to the obstacle.

C. Total potential field

Finally, the total force, which is used to control a mobile robot to move towards the target while avoiding an obstacle of the environment as depicted in figure 2.2, is the sum of the attractive force from the target and the repulsive force from the obstacle as

$$f_{total}(p_r) = f_r^t(p_r) + f_r^o(p_r). \tag{2.7}$$

In general, the total force on the robot is given by

$$f_{total}(p_r) = f_r^t(p_r) + \sum_{i=1}^{M} f_r^{oi}(p_r). \tag{2.8}$$

Where M is the number of the obstacles in the environment, and $f_r^{oi}(p_r)$ is the repulsive force generated by the i^{th} obstacle.

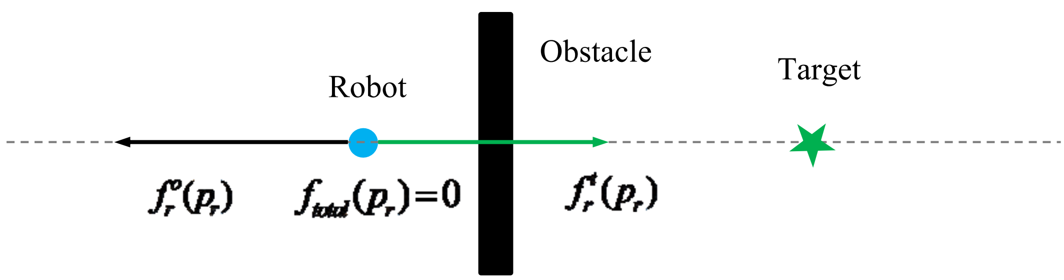

Figure 2.3: Description of the local minimum problem in case: the attractive force from the target and the repulsive force from the obstacle are equal and collinear but on the opposite direction.

2.2.3 Conclusion

Although artificial potential field is known as a positive method for path planning of mobile robots, but in several cases of local minimum problems this approach has demonstrated limitations; for instance, when the attractive force of the target and the repulsive force of the obstacles are equal in magnitude and collinear but on the opposite direction, the total force on the robot is equal to zero, leading to a halt in the robot's motion is stopped. Moreover, in complex environments, such as U-shaped obstacles or long walls, etc., the application of the traditional potential field method for path planning of autonomous robots is very difficult. Robots can be trapped in these obstacles before reaching the target, as seen in figure 2.3.

2.3 Path planning algorithm for a mobile robot

2.3.1 Problem statement

In this sub-section, we consider a mobile robot that moves in a two-dimensional Euclidean space $\{R^2\}$ with M obstacles of the environment. The robot's motion, which is assumed as a moving point in the space, is described by the dynamic model as follows

$$\begin{aligned} \dot{p}_r &= v_r \\ \dot{v}_r &= u_r. \end{aligned} \tag{2.9}$$

In which $p_r,\ v_r \in \left\{R^2\right\}$ are the position vector, the velocity vector of the robot i, respectively. The control input u_r is given as

$$u_r = u_r^t + u_r^o. \tag{2.10}$$

In this equation, the first component u_r^t is used to control the target tracking, and the second component u_r^o is used to drive robot out of the obstacles without the collision. These controllers are presented in subsections below.

In order to simplify the analysis, it is assumed that:

Assumption 1. The position $p_r = (x_r, y_r)^T$ and the velocity $v_r = (v_{xr},\ v_{yr})^T$ of the robot are known. The robot is equipped with sensors such as cameras, sonars, laser sensors, GPS sensors, and associated algorithms, etc., to estimate the trajectory (position $p_t = (x_t, y_t)^T$, and velocity $v_t = (v_{xt}, v_{yt})^T$) of the target precisely.

Assumption 2. The velocity of the moving target is limited by the maximum velocity of the robot $\|v_t\| < v_{r\max}$.

Assumption 3. The robot can sense the position $p_o = (x_o, y_o)^T$ and the velocity $v_o = (v_{xo}, v_{yo})^T$ of the obstacles in the environment precisely.

2.3.2 Target tracking control algorithm

In order to control an autonomous mobile robot as it moves towards the target position, the control law u_r^t is proposed as follows

$$u_r^t = f_r^t(p_r) - k_{rv}^t(v_r - v_t) + \dot{v}_t. \tag{2.11}$$

In this equation, the relative velocity vector $(v_r - v_t)$ between the robot and the target is added as a damping term with the damping scaling factor k_{rv}^t.

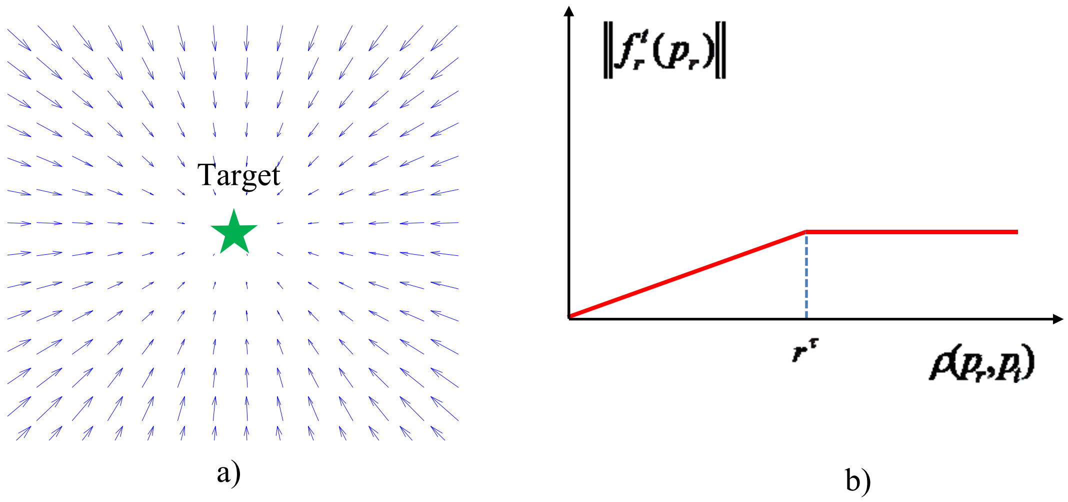

Figure 2.4: The sketch of the attractive force field $f_r^t(p_r)$ directed toward the target position (a), and the dependence of the value of this force field on distance $\rho(p_r, p_t)$ (b).

Under the effect of the attractive force $f_r^t(p_r)$ from the target, the robot will move towards the target position along the direction of this attractive force until it reaches the target. This attractive force is designed as follows:

$$f_r^t(p_r)=\begin{cases} -k_{rp}^t(p_r-p_t)/r^\tau, & \text{if } \rho(p_r,p_t)<r^\tau \\ -k_{rp}^t(p_r-p_t)/\rho(p_r,p_t), & \text{otherwise.} \end{cases} \tag{2.12}$$

Here $r^\tau>0$ is the range around the target, at which the robot's speed is reduced before reaching the target, and (p_r-p_t) is the relative position vector between the robot and the target. The magnitude of this force is dependent on the control factor k_{rp}^t and the distance $\rho(p_r,p_t)=\|p_r-p_t\|$ between the robot and the target. This attractive force is depicted in figure 2.4.

Theorem 2.1. *Consider a mobile robot (p_r, v_r) with its dynamic model (2.9) and control input u_r^t given in equation (2.11). If the given assumptions in sub-section 2.3.1 are satisfied, then the system (2.9) will be stable at the equilibrium state ($p_r=p_t$ and $v_r=v_t$).*

Proof of theorem 2.1

In order to analyze the convergence of the system (2.9) to the equilibrium state, at which $(p_r=p_t, v_r=v_t)$, we first let $x_1=p_r-p_t$, $x_2=v_r-v_t$ be the relative position and velocity between the robot and the target, respectively. The error dynamic model of the system is given as follows

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \dot{v}_r-\dot{v}_t=u_r^t-\dot{v}_t \end{aligned} \tag{2.13}$$

Substitute u_r^t in (2.11) into (2.13) we obtain the error dynamic model of the system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\nabla V_{att}(p_r)-k_{rv}^t(v_r-v_t). \end{aligned} \tag{2.14}$$

Where $\nabla V_{att}(p_r)$ is the gradient field of the potential function $V_{att}(p_r)$ that is described in (2.3).

To analyze the stability of model (2.14) at the equilibrium position $(p_r=p_t, v_r=v_t)$, the positive definite function is selected as follows

$$V = V_{att}(p_r) + \frac{1}{2} x_2^T x_2 \tag{2.15}$$

Consider the potential function (2.3) we have the relationship as follows $\left(\partial V_{att}(p_r)/\partial p_r\right)^T = \left(\partial V_{att}(p_r)/\partial(p_r - p_t)\right)^T$. Taking the time derivative of (2.15) along the trajectory of the system (2.14), we obtain:

$$\begin{aligned}
\dot{V}(t) &= (\nabla V_{att}(p_r))^T \dot{x}_1 + x_2^T \dot{x}_2 \\
&= x_2^T \left(\nabla V_{att}(p_r) + \dot{x}_2\right) \\
&= x_2^T \left(\nabla V_{att}(p_r) - \nabla V_{att}(p_r) - k_{rv}^t x_2\right) \\
&= -k_{rv}^t x_2^T x_2 \le 0.
\end{aligned} \tag{2.16}$$

Equation (2.16) shows that the selected positive definite function V is a Lyapunov function [95], which guarantees that the system (2.14) is stable at the equilibrium point $(p_r = p_t,\ v_r = v_t)$.

2.3.3 Obstacle avoidance control

A. Control algorithm

This sub-section presents the control algorithm for a mobile robot passing through M obstacles to reach a target. As analyzed above, the obstacle avoidance control algorithm is also proposed as follows

$$u_r^o = \sum_{o=1}^{M} \left(f_r^{op}(p_r) + f_r^{or}(p_r) + k_{rv}^o c_r^o (v_r - v_o) \right). \tag{2.17}$$

Where, the relative velocity vector $(v_r - v_o)$ between the robot and its neighbor-obstacle o is used as a damping term with the damping scaling factor k_{rv}^o. Let $N_r^o(t)$ be the set of the obstacles in the neighborhood of the robot at time t, such that:

$$N_r^o(t) = \left\{ \forall o : \rho(p_r, p_o) = \left\| p_r - p_o \right\| \le r^\beta,\ o \in \{1,..M\} \right\}. \tag{2.18}$$

In (2.18), $r^\beta > 0$ and $\rho(p_r, p_o) = \left\| p_r - p_o \right\|$ are the influence range of obstacle and the Euclidean distance between the robot and the obstacle o, respectively. The scalar c_r^o is defined as:

$$c_r^o = \begin{cases} 1 & \text{if } o \in N_r^o(t) \\ 0 & \text{otherwise.} \end{cases} \tag{2.19}$$

The repulsive force field f_r^{op} is created to drive the robot i away from its neighboring obstacle, see figure 2.5. This force field is designed as follows:

$$f_r^{op}(p_r) = c_r^o \left(\left(\frac{1}{\rho(p_r, p_o)} - \frac{1}{r^\beta} \right) \frac{k_{r1}^o}{(\rho(p_r, p_o))^2} - k_{r2}^o \left(\rho(p_r, p_o) - r^\beta \right) \right) n_r^{op}. \tag{2.20}$$

Wherein, the positive factors k_{r1}^o, k_{r2}^o are used to control the fast obstacle avoidance, and $n_r^{op} = (p_r - p_o)/\|p_r - p_o\|$ is a unit vector from the obstacle to the robot.

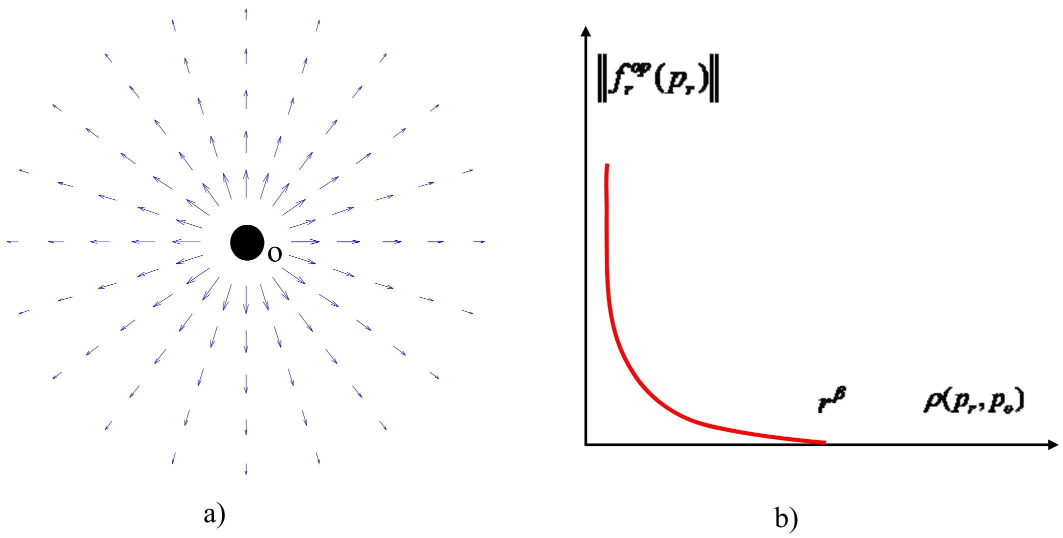

Figure 2.5: The description of the repulsive force field $f_r^{op}(p_r)$ surrounding the neighboring obstacle o of the robot (a), and its magnitude $\|f_r^{op}(p_r)\|$ (b).

In control law (2.17), the rotational force field $f_r^{or}(p_r)$ is added to combine with the repulsive force to drive the robot out of its neighboring obstacle quickly. While the potential force field is used to drive the robot away from its neighboring obstacle, the rotational force field is used to solve the local minimum problems that are still constrained in the potential field method, such as, when robot meets the trapping point, at which the repulsive force of the obstacles and the attractive force of the target are balanced. Under the effect of the added rotational force field, the robot always escapes this trapping point. Furthermore, when the robot is trapped in complex obstacles (for example U-shape or long wall) the rotational vector field will help it to find a new path to escape these obstacles. The direction of the rotational force can be clockwise or counter-clockwise (see figure 2.6). Hence, this rotational force is built as:

$$f_r^{or}(p_r) = w_r^{or} c_r^o n_r^{or}. \tag{2.21}$$

Wherein, the unit vector n_r^{or} is given as:

$$n_r^{or} = c_r^{or}\left((y_r - y_o)/\rho(p_r, p_o), \quad -(x_r - x_o)/\rho(p_r, p_o)\right)^T. \tag{2.22}$$

In this equation, the scalar c_r^{or} is used to define the direction for the rotational force: the rotational force is clockwise if $c_r^{or} = 1$, and counter-clockwise if $c_r^{or} = -1$. Let σ be the angle between the unit vector n_r^{or} and the vector $(p_r - p_o)$, we obtain a relationship as follows:

$$\begin{aligned}\cos\sigma &= c_r^{or}\left((x_r - x_o)(y_r - y_o) - (y_r - y_o)(x_r - x_o)\right)/\left(\rho(p_r, p_o)\right)^2 \\ &= 0.\end{aligned} \tag{2.23}$$

Equation (2.23) shows that the unit vector n_r^{or} is always perpendicular with the vector $(p_r - p_o)$. The positive gain factor $w_r^{or} > 0$, which is the magnitude of the rotational force $f_r^{or}(p_r)$ acting on the robot, is used as a control element to drive the robot to quickly escape obstacles. Therefore, this control element is designed such that the total force on the robot always has the direction in the selected rotational direction. Now, in order to determine this control element, we let $f_r^{tol}(p_r) = (x_r^{tol}, y_r^{tol})^T$ be the total force on the robot. We obtain the relationships as follows

$$\left(x_r^{tol}, y_r^{tol}\right)^T = \left(x_r^{top} + w_r^{or}\frac{c_r^{or}(y_r - y_o)}{\rho(p_r, p_o)}, \; y_r^{top} - w_r^{or}\frac{c_r^{or}(x_r - x_o)}{\rho(p_r, p_o)}\right)^T. \tag{2.24}$$

Where $x_r^{top} = x_r^t + x_r^{op}$ and $y_r^{top} = y_r^t + y_r^{op}$ are the coordinates of the force $f_r^{top}(p_r) = (x_r^{top}, y_r^{top})^T$ that is the sum of the attractive force $f_r^t(p_r) = (x_r^t, y_r^t)^T$ from the target and the repulsive force $f_r^{op}(p_r) = (x_r^{op}, y_r^{op})^T$ from the obstacle on the robot. $x_n^{or} = c_r^{or}(y_r - y_o)/\rho(p_r, p_o)$ and $y_n^{or} = -c_r^{or}(x_r - x_o)/\rho(p_r, p_o)$ are the coordinates of the unit vector n_r^{or} as proposed in (2.22).

Let $\alpha_d = \angle(f_r^{tol}(p_r),\, n_r^{op})$ be the desired angle between the total force $f_r^{tol}(p_r)$ and the unit vector $n_r^{op} = \left(x_n^{op},\, y_n^{op}\right)^T = \left((x_r - x_o)/\|p_r - p_o\|,\; (y_r - y_o)/\|p_r - p_o\|\right)^T$, see figure 2.7, we have the relationship as

$$cos(\alpha_d) = \frac{x_n^{op}(x_r^{top} + w_r^{or} x_n^{or}) + y_n^{op}(y_r^{top} + w_r^{or} y_n^{or})}{\sqrt{\left(x_r^{top} + w_r^{or} x_n^{or}\right)^2 + \left(y_r^{top} + w_r^{or} y_n^{or}\right)^2}}. \tag{2.25}$$

This equation can be rewritten as follows

$$a\left(w_r^{or}\right)^2 + bw_r^{or} + c = 0, \tag{2.26}$$

Here: $a = (cos(\alpha_d)x_n^{or})^2 + (cos(\alpha_d)y_n^{or})^2 - (x_n^{or}x_n^{op} + y_n^{or}y_n^{op})^2$,

$b = 2\left(cos(\alpha_d)\right)^2\left(x_r^{top}x_n^{or} + y_r^{top}y_n^{or}\right) - 2\left(x_n^{op}x_n^{or} + y_n^{op}y_n^{or}\right)\left(x_r^{top}x_n^{op} + y_r^{top}y_n^{op}\right)$,

$c = \left(cos(\alpha_d)x_r^{top}\right)^2 + \left(cos(\alpha_d)x_n^{op}\right)^2 - \left(x_n^{op}x_r^{top} + y_n^{op}y_r^{top}\right)^2$.

The control element w_r^{or} is proposed in Algorithm 2.1.

Algorithm 2.1: Determination the control element w_r^{or}

The control element w_r^{or} is determined as follows

if $b^2 - 4ac \ge 0$ **then**

  if $\frac{-b+\sqrt{b^2-4ac}}{2a} > 0$ **then**

$$w_r^{or} = \frac{-b+\sqrt{b^2-4ac}}{2a}$$

  else

$$w_r^{or} = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

  end

else

$$w_r^{or} = (1+c_k)\left(\sqrt{(x_r^t + x_r^{op})^2 + (y_r^t + y_r^{op})^2} + \lambda_o\right)$$

end

In the Algorithm 2.1, the positive factor λ_o guarantees that if the sum of the attractive force and the repulsive force on the robot is equal to zero, then the added rotational force is large enough in order to drive the robot quickly escaping the trapped point as described in figure 2.3. The scaling factor c_k, which depends on the angle α between the sum vector $f_i^{top}(p_r)$ and the unit vector n_r^{or} (see figure 2.7), is described as follows:

$$c_k = \begin{cases} c_1\ , & \text{if } \alpha < \pi/2 \\ c_2\ , & \text{otherwise.} \end{cases} \tag{2.27}$$

Where, two constants c_1 and c_2 can be chosen as follows: $-1 < c_1$, $0 < c_2$, and $c_1 < c_2$. Algorithm (2.1) also shows that the robot is driven in the direction of the rotational force $f_r^{or}(p_r)$ when it meets the obstacle. Hence, it can easily escape the obstacles in order to continue to reach the target, see figure 2.7.

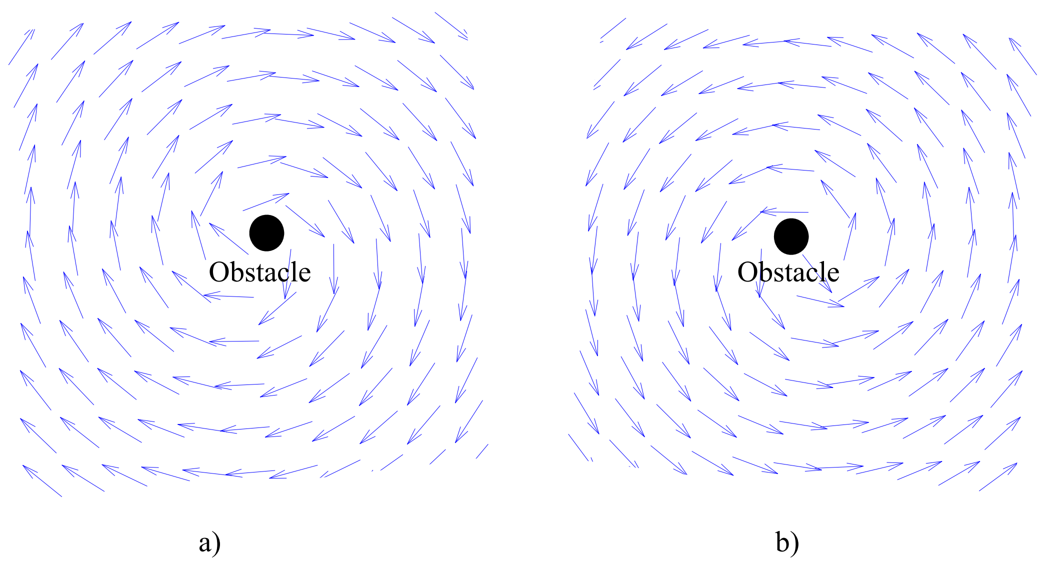

Figure 2.6: Description of the clockwise rotational force field (a) and the counter-clockwise rotational force field (b).

B. Obstacle avoiding direction

As introduced in section 2.1, the obstacle avoiding direction for the mobile robot has a great influence on finding the fastest way towards the moving target. The path planning for a mobile robot to track a moving target in an unknown environment is shown in figure 2.7. The robot has to overcome the U-shaped obstacle in order to track the moving target. In an unknown environment, it is very difficult to determine the desired direction of movement for the robot to easily escape obstacles and simultaneously reach the target quickly. Hence, a positive method to solve this problem is to control the robot to escape obstacles following the direction of the target's trajectory. This method is built on the geometry, as depicted in figure 2.7 and figure 2.8.

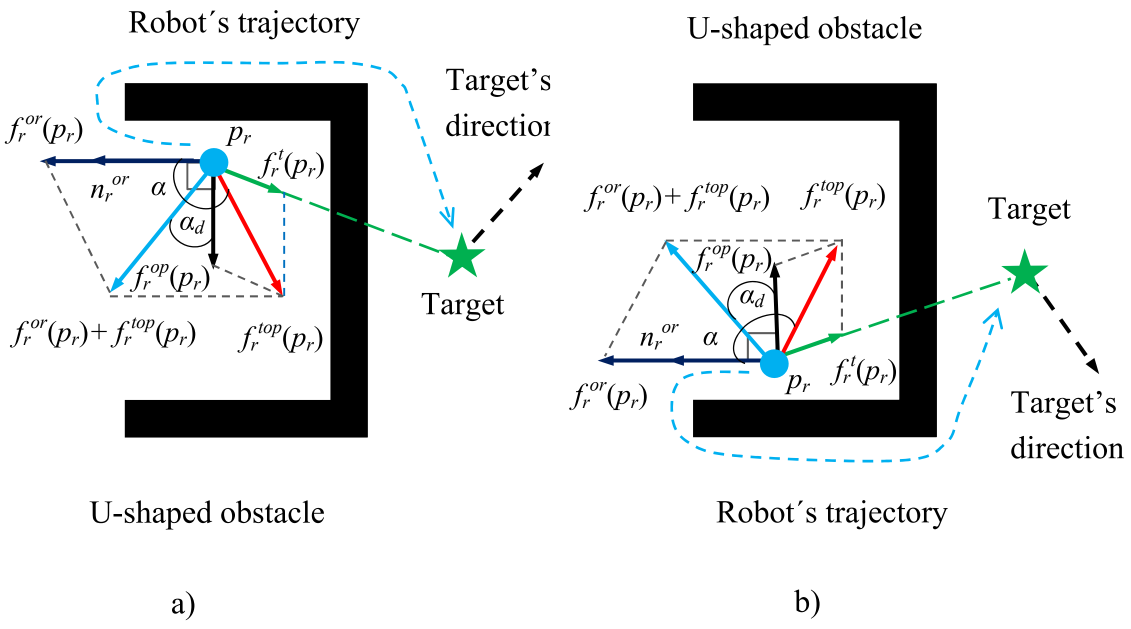

Figure 2.7: The geometric description of the obstacle avoidance and escape for a mobile robot while reaching the target: clockwise (a) and counter-clockwise (b).

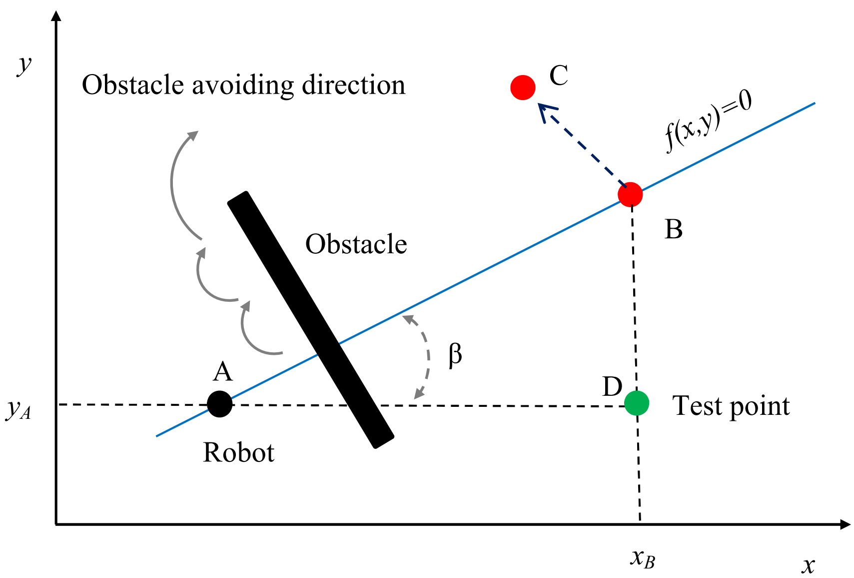

Figure 2.8: The geometric description of the obstacle avoiding direction of a mobile robot according to the clockwise direction in case $0 < \beta < \pi / 2$.

In figure 2.8, the points B and C are denoted as the position of the target at time (t) and at time (t+Δt), respectively. The positive constant Δt is a preselected time-step used to determine the relative position between B and C. Point A is the robot's position at time (t). The line *f(x,y)=0* through points A and B (see in [96], [97]) is used as the basis to determine the moving direction of the target. Here *f(x,y)* is described as follows:

$$f(x,y) = \frac{x - x_A}{x_B - x_A} - \frac{y - y_A}{y_B - y_A} \ . \tag{2.28}$$

The angle β between the vector BA and the unit vector along the coordinate x-axis is used to determine the direction of the relative position vector between B and A. As shown in [96], the constituted line from two points A and B *f(x,y)=0* will split the coordinate plane *xy* into two half-planes. One side of this boundary line consists of all points that satisfy the inequality *f(x,y)<0.* Otherwise, all points on the opposite side satisfy the inequality *f(x,y)>0.* In order to know which side of the boundary line *f(x,y)=0* the target is moving towards, we choose a test point $D = (x_B, y_A)^T$. This test point is used for comparison against the position of the target (point C) at time (t+Δt), see figure 2.8. If point C lies on the half-plane containing the test point D then *f(C)f(D)>0.* In contrast, if C and D lie on the different sides of the boundary line *f(x,y)=0* then *f(C)f(D)<0.* In figure 2.8, the rotational direction c_r^{or} of the rotational force is depicted in the case $0 < \beta < \pi / 2$. This rotational direction is determined by the moving direction of the target. It is described as follows:

$$c_r^{or} = \begin{cases} 1 & \text{if } f(C)f(D) < 0 \\ -1 & \text{otherwise }. \end{cases} \tag{2.29}$$

However, in practice the angle β can change in the quadrants of the coordinate system *xy*. Moreover, if the target does not move or the test point D sits on the boundary line *f(x,y)=0,* then *f(C)f(D)=0.* In these situations, the scalar c_r^{or} can be chosen arbitrary as $c_r^{or} = 1$ or $c_r^{or} = -1$. In summary, the obstacle avoiding direction for the robot is proposed in table 2.1.

TABLE 2.1: DETERMINATION ROTATIONAL DIRECTION

β	$f(C)f(D)$	c_r^{or}
$0 \le \beta < \pi/2$	$f(C)f(D) \le 0$	1
	$f(C)f(D) > 0$	-1
$\pi/2 \le \beta < \pi$	$f(C)f(D) \ge 0$	1
	$f(C)f(D) < 0$	-1
$\pi \le \beta < 3\pi/2$	$f(C)f(D) \le 0$	1
	$f(C)f(D) > 0$	-1
$3\pi/2 \le \beta < 2\pi$	$f(C)f(D) \ge 0$	1
	$f(C)f(D) < 0$	-1

2.4 Simulation results

In this section we test our proposed theoretical results for the target tracking of a mobile robot. For the simulations, we assume that the initial velocities of the robot and the target are set to zero, and the general parameters are listed as follows: $r^{\tau} = 100$, $r^{\beta} = 20$, $k_{rp}^{t} = 3$, $k_{rv}^{t} = 2{,}1$, $k_{r1}^{o} = 80$, $k_{r2}^{o} = 6$, $k_{rv}^{o} = 1.6$, $c_1 = 0.7$, $c_2 = 3{,}8$, $\lambda_0 = 2$.

2.4.1 The target tracking in a free environment

In this sub-section, we test the control algorithms for a mobile robot tracking a moving target in a free environment, in which no obstacle exists. The parameters for this simulation are given as follows: The initial position of the robot is set at the position $p_r = (50,\ 200)^T$, and the target moves along the trajectory $p_t = (0.7\mathrm{t} + 200,\ -0.5\mathrm{t} + 800)^T$.

The results of the simulations in figure 2.9, figure 2.10 and figure 2.11 show that the target tracking of a mobile robot is successful, the robot approaches the position of the target very well. At the initial time, the position of the robot is far from the target, but after a period of circa 400s the robot can catch up the moving target and then continue to track this moving target. The trajectory of the robot is always to follow up the trajectory of the target with very small error (the position error between the robot and the target, see figure 2.10). Moreover, the state of the robot (p_r, v_r) as described in (2.9) always converges to the equilibrium state, at which $p_r = p_t$ and $v_r = v_t$, see figure 2.10 and figure 2.11. These simulation results confirm the theory stated in Theorem 2.1.

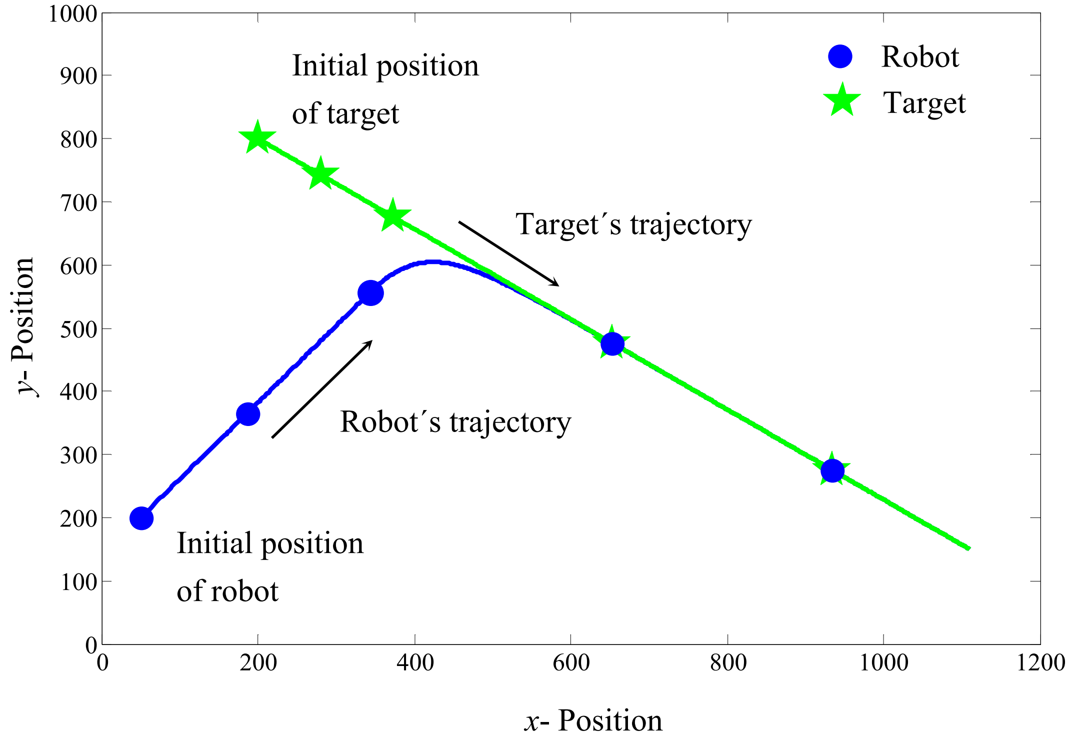

Figure 2.9: A mobile robot tracks a moving target in free environment.

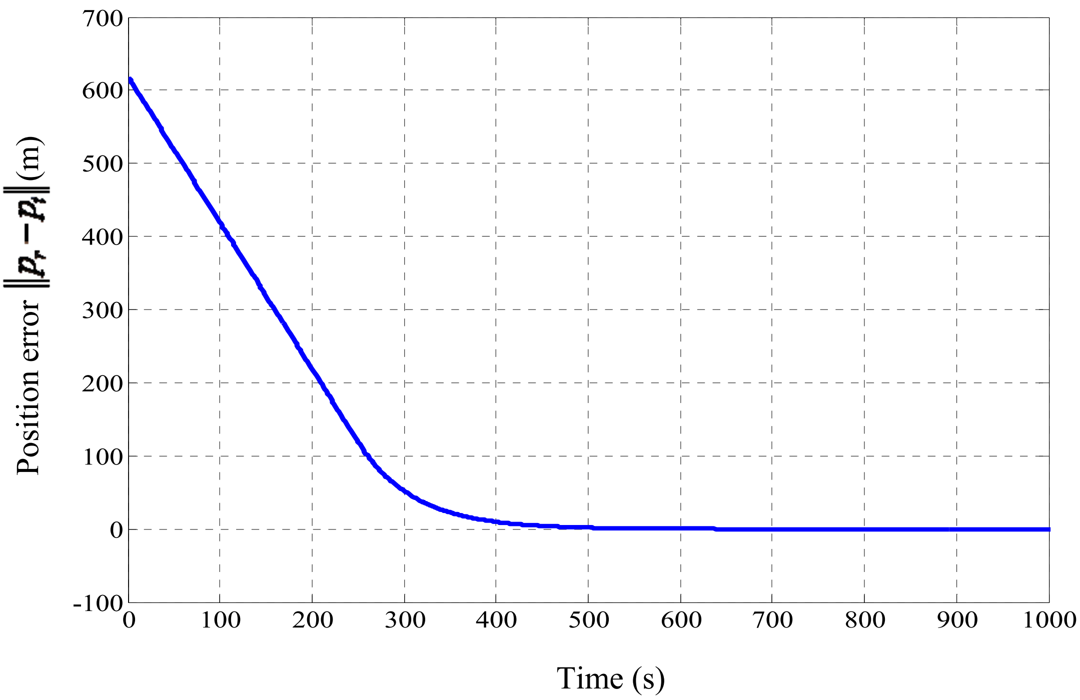

Figure 2.10: Tracking error between the mobile robot and the moving target.

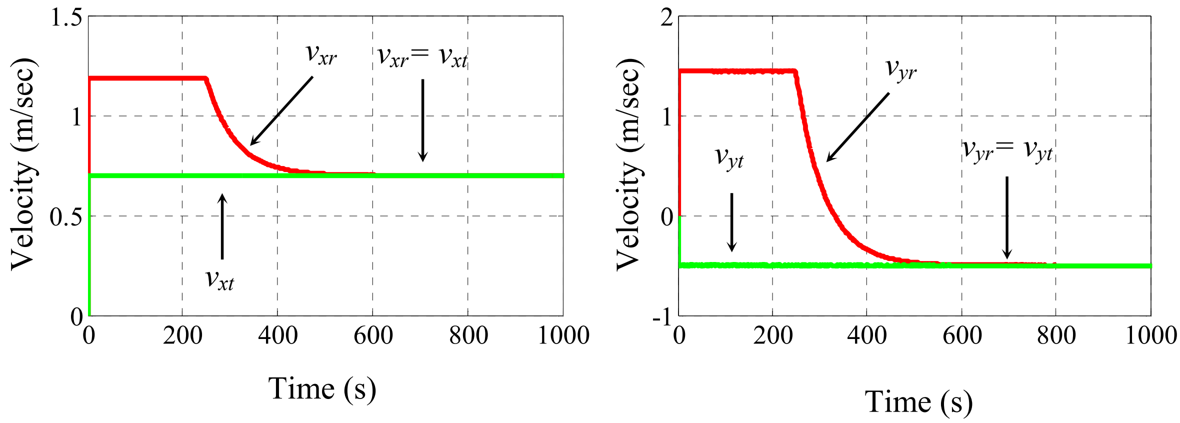

Figure 2.11: The velocities of the robot and the target while tracking.

2.4.2 The target tracking under the influence of the obstacles

In this sub-section, we test the control algorithms for a mobile robot reaching a target in an environment, in which there are the obstacles. For this simulation, we assume that the obstacles of the environment are stationary, and their position and shape (U-shape and long wall) are also determined.

Case 1. Consider a stationary target.

First of all, the obstacle avoidance control algorithm is tested when the target is stationary. The parameters for this simulation are specified as follows: The position of the target is set at the position $p_t = (100,\ 450)^T$, and the initial position of the robot is set at the position $p_r = (50,\ 450)^T$.

The result of the simulation in figure 2.12 shows that the robot is trapped in the U-shape obstacle while reaching the target when we use only the repulsive potential field in the obstacle avoidance controller. In this situation, the attractive force of the target and the sum of the repulsive forces from the obstacles at the local minimum point are equal and collinear but on the opposite direction. Hence, the total force on the robot is equal to zero, and the motion of the robot is stopped before the robot can approach the target position.

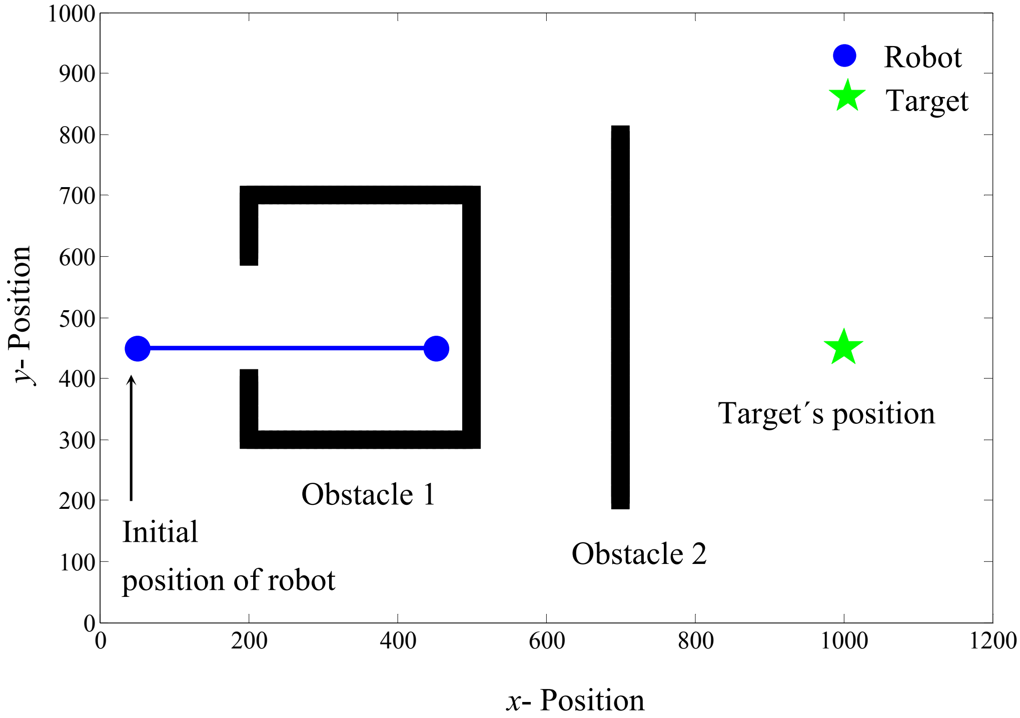

Figure 2.12: Path planning for a mobile robot reaching a stationary target in a dynamic environment using only the repulsive potential field in the obstacle avoidance controller. The motion of the robot is stopped before the robot can approach the target position.

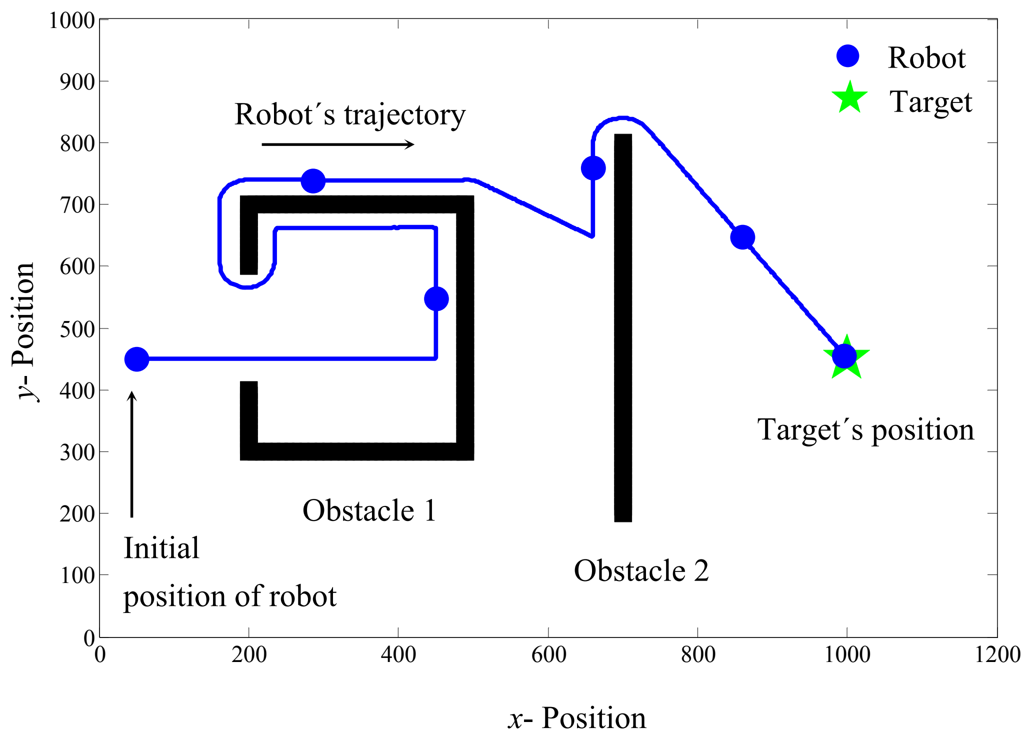

Figure 2.13: Path planning for a mobile robot reaching a stationary target in a stationary environment using the clockwise rotational vector field combined with the repulsive potential field in the obstacle avoidance controller.

In contrast, the result of the simulation in figure 2.12 shows that using the blended force field, which consists of the rotational vector field (in this simulation the clockwise rotational vector field is applied) combined with the repulsive potential field in the obstacle avoidance controller, the local minimum problem as shown in figure 2.12 is solved. This blended force field drove the robot to get out the U-shape obstacle when it is trapped in this U-shape obstacle. In other words, the obstacle avoidance of a mobile robot is successful without the collisions with these obstacles. The robot easily escaped the trapping point at which the sum of the repulsive forces of the obstacles and the attractive force of the target are balanced as shown in figure 2.12. Then, it followed the direction of the clockwise rotational force around the obstacles to find a path to exit these obstacles. After the robot overcame the obstacles, it continued to move towards the target, see figure 2.13.

Case 2. Consider a moving target.

In this case, the obstacle avoidance control algorithm is tested when the target is dynamic. The parameters for this simulation are specified as follows: The target moves along the trajectory $p_t = (0.1t + 200,\ -0.4t + 800)^T$, and the initial position of the robot is also set at the position $p_r = (50,\ 450)^T$.

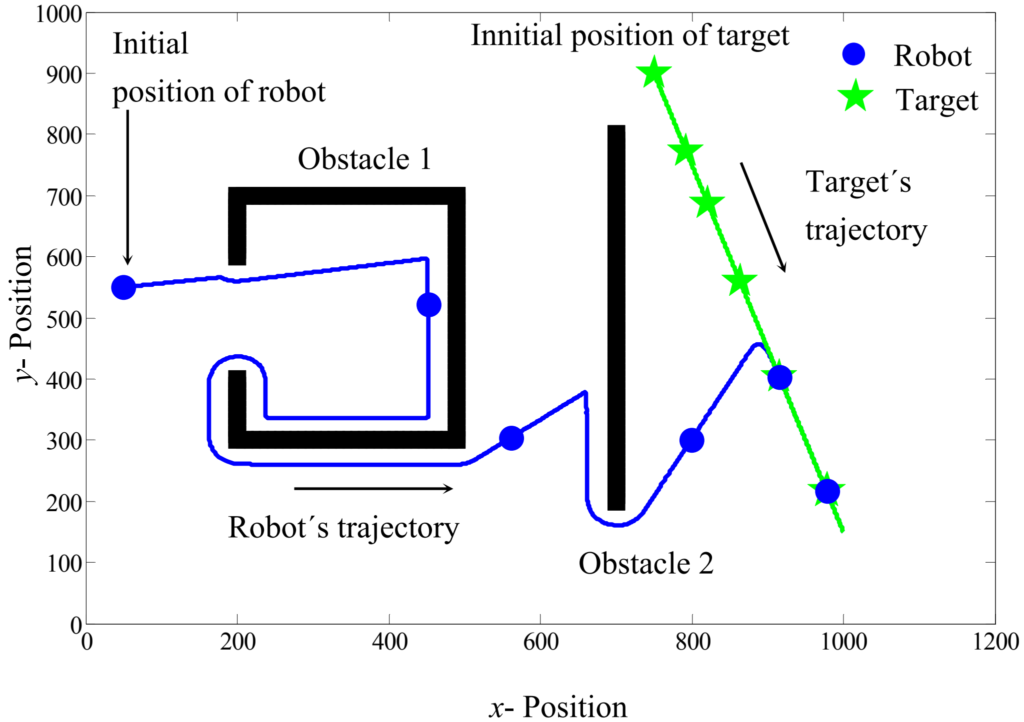

Figure 2.14: Path planning for a mobile robot to track a moving target in a stationary environment using the rotational vector field combined with the repulsive potential field in the obstacle avoidance controller.

The result of the simulation in figure 2.14 shows the intelligence of a mobile robot while pursuing a moving target. When this robot detects the obstacle it changes its moving direction to avoid collision with this obstacle and searches the new path to chase the target simultaneously. Figure 2.14 also shows that the obstacle avoidance of a mobile robot according the moving direction of the target is successful. The movement direction of the robot is driven towards the left of the U-shape and the wall-shape obstacles by the counter-clockwise rotational force field from these obstacles. Hence, this mobile robot can easily and quickly escape the U-shaped and the wall-shape obstacles in order to move towards the target without collisions.

2.5 Summary

This chapter has presented an approach to path planning for a mobile to reach a target in a dynamic environment based on the combination of the traditional potential fields and the rotational vector field. In this approach, the artificial attractive force field from the target is used to drive the robot towards the target position. The repulsive force field surrounding the obstacles is used to drive the robot away from the obstacles, while the rotational force field is added surrounding the obstacles to help the robot to quickly escape the balance point at which the sum of the attractive force from the target and the repulsive forces from the obstacles is equal to zero. Especially, in case the robot is trapped in complex obstacles, such as U-shaped obstacles, this added rotational vector field also plays an important role in helping the robot to find a new path to escape these obstacles. The obstacle avoiding direction for the mobile robot is designed to be in the moving direction of the target, such that the robot can easily escape the obstacles and find the fastest path towards the target. The results of the simulations have shown that, under the effect of the blended force field, an autonomous robot can easily find a path to reach a target in a dynamic environment.

The development of this proposed approach to path planning for the formation of a swarm of multi robots to track a moving target in a dynamic environment, as an interesting research topic, will be examined in the next chapters.

3 Formation Control Following Desired Formations

3.1 Introduction

This chapter presents a method for formation control of the autonomous robots following the desired formations while tracking a moving target under the influence of the dynamic environment. This method is built and developed based on the artificial potential field method. Formation-shape control of the autonomous robots has been one of the positive research directions in robotics control. This interesting research direction has attracted the attention from researchers in recent years. Some of the typical researches focusing on the topic include: "Formation control following a dynamic framework" is presented in [85] - [87], "Formation control following dynamic region" is published in [88], [89]. In these approaches, all robots in the group move together inside a given framework or region. They stay within a moving region and are able to adjust the formation by rotating and scaling during movement together. This method does not require specific orders or positions of the robots inside the given region. Each robot's motion depends on the motion of its neighbors and framework or region. Furthermore, formation control of the autonomous robots following desired shapes has also been a positive research direction, however it is still limited. In this approach, robots are controlled to achieve the given positions in the desired shape [90] - [92].

In this chapter, this formation control following the desired formation is developed and applied for the formations in practice, such as collinear, V-shape or circular shape formation, see figure 3.2. In our approach, the desired formation (collinear, V-shape or circular shape formation) with the constant distances between neighboring virtual nodes is firstly generated based on the relative position between the leader and the target. Then, robots are independently controlled by the attractive potential field from the virtual nodes in the desired formation. Moreover, the designed virtual nodes in the desired formation guarantee that the neighboring robots do not interact with each other. Hence, robots can easily converge to these virtual nodes under the velocity matching without collisions. The leader is randomly chosen from a member robot which is closest to the target. This leader plays an important role to create and lead its formation moving towards the target in a stable direction. Hence, in many undesired cases, such as the leader is broken or trapped in the obstacles (for example U-shape obstacle) one new leader is replaced in order to continue to lead the swarm tracking the target. Furthermore, the motion of the robots to the desired positions in the desired formations is controlled by the artificial force fields, which consist of local and global potential fields surrounding the virtual nodes. Under the effect of these artificial force fields, robots will automatically find their desired position at the virtual nodes in the desired formation. Additionally, the local repulsive force field surrounding each robot is used to avoid the collision with each other. Moreover, using the repulsive vector field combined with the rotational vector field in the obstacle avoiding controller, robots can easily escape the obstacles while tracking a moving target. The control architecture for each autonomous robot is depicted in figure 3.1.

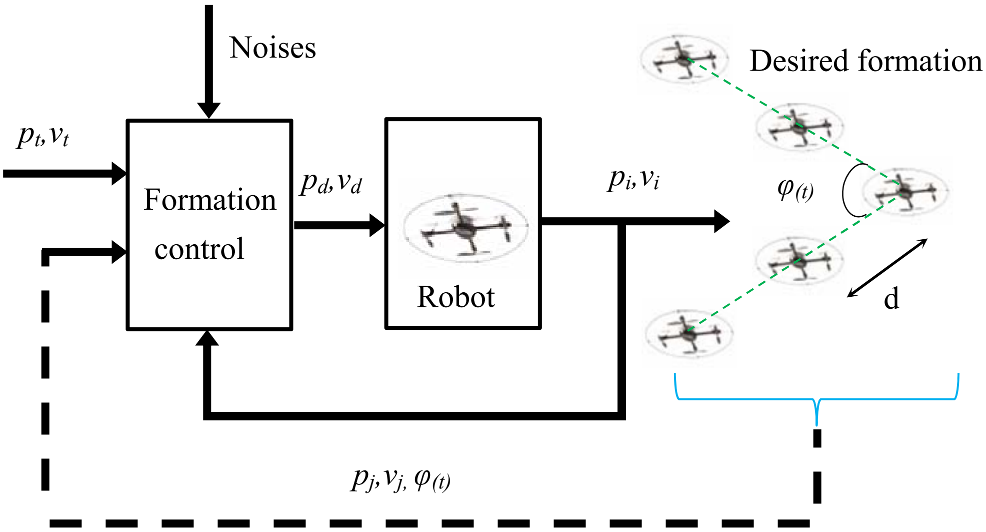

Figure 3.1: Illustration of the architecture for formation control of autonomous robots following desired formation.

(a) (b)

Figure 3.2: The V-shape flying formation of birds (a) (Source: http://www. grahamowengallery.com/photography/geese.html), and the collinear flying formation of aircraft (b) (Source: http://avioners.net/2013/03/breitling-acrobatics-team-using-l-39.html/).

The main contributions of this chapter are as follows:

- "Formation adaptation while tracking a moving target" is presented in section 3.4. In this content, we consider an approach for the adaptive formation control of autonomous robots following desired shapes under the influence of the environment, such as noises, obstacles, etc. In this approach, the V-shape and circle-shape formation are used as the desired formations. Firstly, the V-shape formation is used to track a moving target in the global potential field from the target. As analyzed in literatures [4]-[7], the V-shape formation of birds (for example the formation flight of the Canada geese during migration, see figure 3.2a) has a lot of advantages, such as energy savings during flight. The research results in these published literatures indicated that the members in the V-shape formation realized up to 51% in energy savings over solo flight. Moreover, by flying in V-shape formation the members in the formation can easily communicate with each other, etc. Additionally, the circular shape formation with its advantages, such as, the members in the formation can easily cooperate with each other in tasks for example the encirclement, surveillance of the target [93], [94], is used to approach the target in the local potential field surrounding the target.

- "Direction control for collinear formation" is presented in section 3.5. In this content, we consider an approach to control autonomous robots to achieve a desired collinear formation during movement towards the target position. As presented in [91], the collinear formation has a lot of advantages, such as the members in the formation can easily cooperate to each other in tasks for example the observation, monitoring and tracking the targets, see figure 3.2b. In this approach, one robot, which has the closest distance to the target, is firstly selected as the leader of the swarm. The desired formation is built based on the relative position between this leader and the target. Secondly, the trajectory of the remaining robots towards the desired positions in the desired formation is driven by the artificial force fields. These force fields consist of the local and global attractive potential fields surrounding each virtual node in the desired formation. Furthermore, an orientation controller is added in order to guarantee that the desired formation is always headed in the invariant direction to the target position. In addition, the local repulsive force fields around each robot and obstacle are employed in order to avoid collisions during movement. The stability of a swarm following a desired collinear formation in invariant direction towards the target position is verified in simulations.

The remaining sections of this chapter are organized as follows: the problem formulation is presented in the section 3.2. Section 3.3 presents the method in order to build the desired formations. In section 3.4, the formation adaptation control algorithm for a swarm while tracking a moving target is presented. The direction control for a collinear formation is discussed in section 3.5.

3.2 Problem formulation

In this section, we consider a swarm of N robots that has the mission to track, encircle a moving target in two-dimensional space. Let $p_i = (x_i, y_i)^T$, $v_i = (v_{ix}, v_{iy})^T$ be the position, velocity vector of the robot i ($i=1,2,..,N$), respectively. The dynamic model of the robot i is described as:

$$\begin{aligned} \dot{p}_i &= v_i \\ \dot{v}_i &= u_i, \qquad i = 1,..,N. \end{aligned} \tag{3.1}$$

As introduced above, the aim of this chapter is to control the formation of autonomous robots following desired formations that are generated based on the relative position between the leader and the target. While tracking the moving target, robots must avoid the collisions with each other, avoid the obstacles, and their formation must be maintained and kept stable under the influence of the environment. Hence, in order to solve these conditions we propose the control input u_i for each robot as:

$$u_i = \begin{cases} u_i^t + u_i^o, & \text{if robot } i \text{ is the leader } l;\ i,l = 1,..,N \\ u_i^j + u_i^o + u_i^k, & \text{otherwise.} \end{cases} \tag{3.2}$$

Where, the first controller u_i^j is used to control the formation connection. The second controller u_i^o is used to drive robots avoiding obstacles. The controller u_i^k is added to help robots avoid the collision during movement. Using the tracking controller u_i^t the leader can easily drive its swarm towards the target. Now, in order to design these controllers, firstly, we have some definitions and remarks as follows:

Definition 3.1. The desired collinear formation is the locus of all virtual nodes that lie on a line and equidistant to each other.

Definition 3.2. The desired V-shape formation is a formation that is linked by two linear formations. These line formations are driven by a selected leader, and connected by a formation angle φ. In each line formation, the virtual nodes are equidistant to each other.

Definition 3.3. The desired circular shape formation is the locus of all virtual nodes that are equidistant to each other and equidistant from the target.

Definition 3.4. Robot i ($i=1,2,..,N;\ p_i=(x_i,\ y_i)^T;\ v_i=(v_{ix},\ v_{iy})^T$) is called an active robot at time t if the distance from it to the virtual node j ($j=1,2,..,N$) of the desired formation is smaller than the radius of the active circle surrounding each virtual node ($d_i^j < r_a = d/2-\lambda_a$, λ_a is a positive constant), see figure 3.3. Otherwise, it is a free robot.

Definition 3.5. Virtual node j ($j=1,2,..,N;\ q_j=(x_j,\ y_j)^T;\ v_j=(v_{jx},\ v_{jy})^T$) of the desired formation is active if there is a robot i ($i=1,2,..,N$) in the active circle of this virtual node. In contrast, it is free.

Definition 3.6. Desired position for each robot i in the desired formation is a virtual node j at which $\lim_{t\to\infty}(p_i(t)-q_j(t))=0$, and the virtual node ($j$-$1$) is also active.

Remark 3.1. Consider a desired formation (V-shape or circular shape formation) of N virtual nodes as shown in figure 3.3. Each Robot must find an desired position in this desired formation. Firstly, each free robot i will pursue the closest free virtual node j in order to be active at this virtual node. If the position of this active robot at the active node j is still not desired (for example robot k ($k=1,2,..,N,\ k\neq l$ in figure 3.3), then this active robot will automatically move into the virtual nodes (j-1) until it achieves an desired position in the desired formation.

Remark 3.2. The motion of the formation depends on the relative position between the leader and the target. At initial time, one robot, which is closest to the target, is chosen as the leader, and then it is saved to lead its formation towards the target. During the movement, if this leader encounters any risk, such as it is broken or hindered by the environment, then a new leader is replaced. This new leader will reorganize the formation and continue to lead the new formation tracking the target.

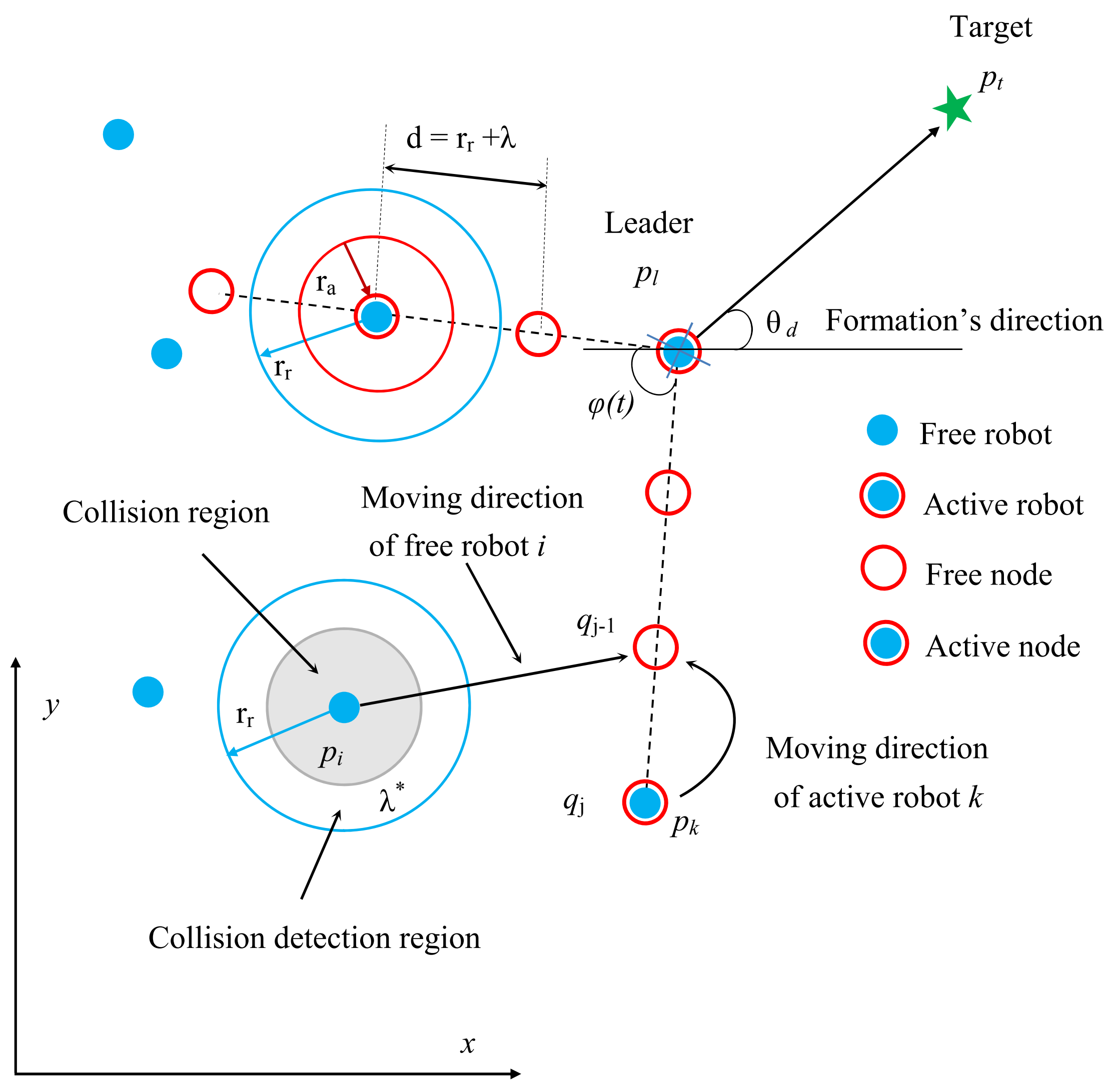

Figure 3.3: Description formation control following the V-shape and circular shape desired structure.

3.3 Building desired formations

This section presents methods in order to build the desired formations (Collinear, V-shape and circular formation) based on the relative position between the target and the leader combined with the coordinate system rotation and translation. Assume that the leader's position and the target's position are located at the positions $p_l = (x_l, y_l)^T$, $p_t = (x_t, y_t)^T$, respectively. The relative position vector between the leader and the target is $(p_l - p_t) = (x_l - x_t, y_l - y_t)^T$, and the distance between them is determined as $d_l^t = \sqrt{(x_l - x_t)^2 + (y_l - y_t)^2}$.

3.3.1 Collinear desired formation

In order to build the collinear desired formation including the virtual nodes j ($j=1,2,..,N;\ q_j=(x_j,\ y_j)^T;\ v_j=(v_{jx},\ v_{jy})^T$), which are equidistant from each other with the constant distance d, and deviating from the vector $(p_t - p_l)$ at desired angle δ_d, one base node $q_\mu = (x_\mu, y_\mu)^T$ is firstly generated with the distance $d = \left\| q_\mu - p_l \right\|$, and $\delta_d = \angle(q_\mu - p_l),(p_t - p_l)$, see figure 3.4.

As depicted in figure 3.4, the coordinates of the base node q_μ on the coordinate system $x'y'$ ($q'_\mu = (x'_\mu, y'_\mu)^T$) are determined as follows:

$$\begin{pmatrix} x'_\mu \\ y'_\mu \end{pmatrix} = \left\| q_\mu - p_l \right\| \begin{pmatrix} \cos \delta_d \\ \sin \delta_d \end{pmatrix}. \tag{3.3}$$

By rotating and translating equation (3.3) according to coordinate systems $x''y''$ and xy, see [96], [97], we obtain the position of the desired node q_μ on the coordinate system xy as follows:

$$q_\mu = p_l + Rq'_\mu. \tag{3.4}$$

The rotational matrix R, which depends on the rotational angle theta θ, is determined as follows:

$$R = \begin{cases} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} & \text{if } \theta \text{ rotates clockwise,} \\ \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} & \text{otherwise.} \end{cases} \tag{3.5}$$

From the base node q_μ and the leader we obtain a unit vector along the line connecting from q_μ to p_l as $n_{\mu l} = (p_l - q_\mu) / \left\| p_l - q_\mu \right\|$, see figure 3.4. Now, a virtual node j ($d_j^l = jd$; $q_j = (x_j,\ y_j)^T;\ v_j = (v_{jx},\ v_{jy})^T; j = 1,2,..,N$) is determined by the unit vector $n_{\mu l}$ as follows:

$$\left(q_j - p_l\right) = jdn_{\mu l}\ . \tag{3.6}$$

Substitute $n_{\mu l} = \left(p_l - p_\mu\right) / \left\| p_l - p_\mu \right\|$ into equation (3.6) we obtain:

$$q_j = \left(1 + j\right) p_l - jq_\mu\ . \tag{3.7}$$

Equation (3.7) can be rewritten as follows:

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} (1+j)x_l \\ (1+j)y_l \end{pmatrix} - \begin{pmatrix} jx_\mu \\ jy_\mu \end{pmatrix}. \tag{3.8}$$

Equation (3.8) shows that when j changes from $j=1$ to $j=N$ we get the formation of the N virtual nodes, which lie on a line through p_l and q_μ, and are equally spaced see figure 3.4. However, in this solution, the leader always has the outer position of the formation, so this situation is suitable for the row formation. For the parallel formation, the leader is positioned as the center of the formation, so the algorithm to generate desired formation is redesigned as follows:

$$q_j = \begin{cases} jp_l - (j-1)q_d, & if \ j \le N/2+1 \\ (1-\xi)p_l + \xi q_d, & Otherwise, \end{cases} \tag{3.9}$$

where ξ is described as $\xi = j - floor(N/2) - 1$. Using this algorithm, the virtual nodes will be evenly distributed to both sides of the leader as depicted in figure 3.5.

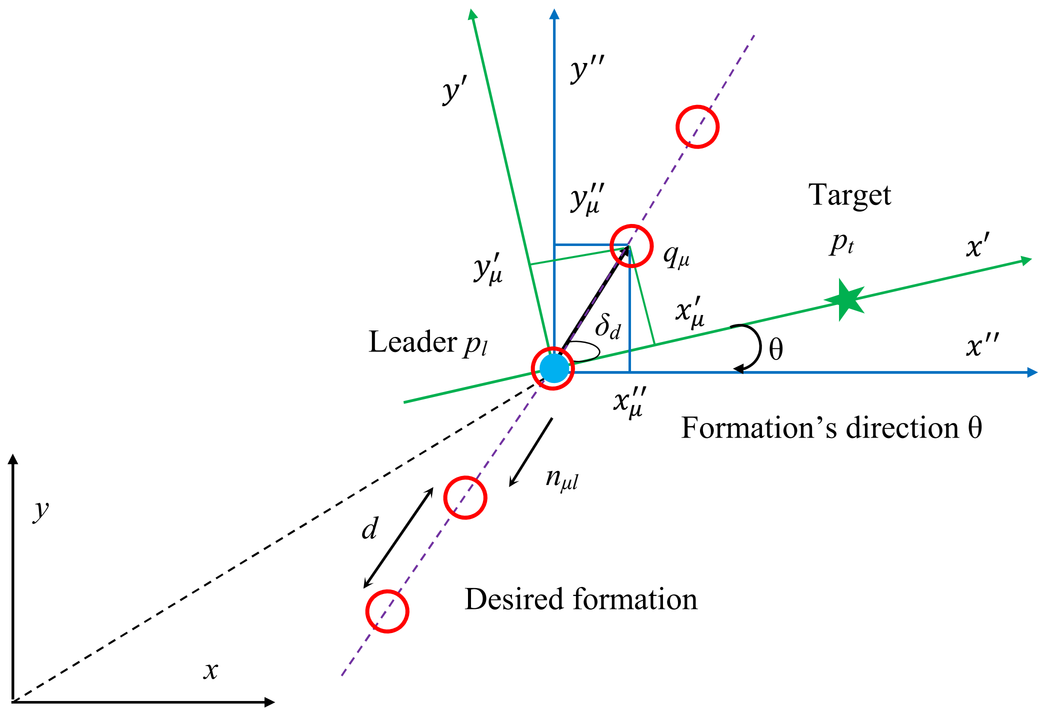

Figure 3.4: The description of the method to build the collinear desired formation.

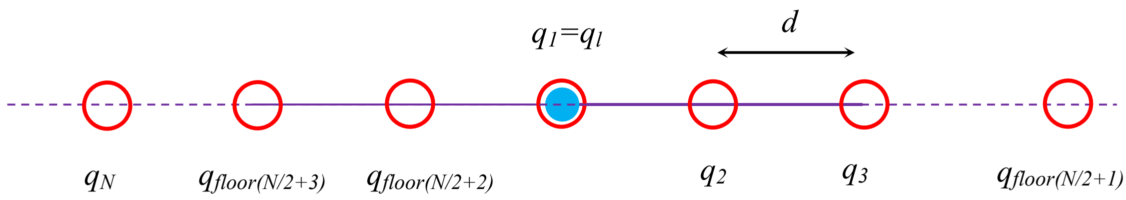

Figure 3.5: The description of the distributed virtual nodes *j* (*j=1,2,..,N*) in the collinear desired formation.

3.3.2 V-shape desired formation

As presented in definition 2, the desired V-shape formation is generated by two linear formations. These linear formations use a leader together and are connected by a formation angle $\varphi(t)$. In the linear formations, the virtual nodes are equidistant from each other. Therefore, in order to build the V-shape desired formation we firstly use equation (3.7) to create the right side of this desired V-shape formation, see figure 3.6. Similarly, the virtual nodes j on the left side of the desired V- shape formation is also designed based on the desired formation angle φ_d and the relative position between the leader and the target as

$$(q_j - p_l) = jdn_{l\eta} \ . \tag{3.10}$$

Wherein, the unit vector $n_{l\eta}$ is calculated as $n_{l\eta} = (q_\eta - p_l)/\|q_\eta - p_l\|$. Substituting this unit vector into (3.10) we obtain as follows:

$$q_j = (1-j)p_l + jq_\eta \ . \tag{3.11}$$

In Equation (3.11), q_η is the position of the base node on the line deviates with the line through the leader and the target an angle (π- δ_d), see figure 3.6. This base node is determined similarly to equation (3.4) as follows:

$$q_\eta = p_l + Rq'_\eta. \tag{3.12}$$

Using equation (3.12), we obtain the formation of the virtual nodes $j\,(j=1,2,..,N)$ that lie on the line through p_l and q_η and are equally spaced (the left side of the V-shape desired formation), see figure 3.6.

Finally, the algorithm to generate the desired V-shape formation of the virtual nodes j ($j=1,2..,N$) is proposed as:

$$q_j = \begin{cases} (1+\xi_1)p_l - \xi_1 q_\mu, & \text{if } j \le N/2+1 \\ (1-\xi_2)p_l + \xi_2 q_\eta, & \text{otherwise.} \end{cases} \tag{3.13}$$

Where, $\xi_1 = j-1$ and $\xi_2 = j-1-floor(N/2)$ are the positive constants. Using equation (3.13), the virtual nodes j ($j=1,2,..,N$) will be evenly distributed to both sides of the leader. Hence, we obtain a V-shape formation owning these desired virtual nodes and a constant formation angle $\varphi_d = 2\delta_d$ as depicted in figure 3.7.

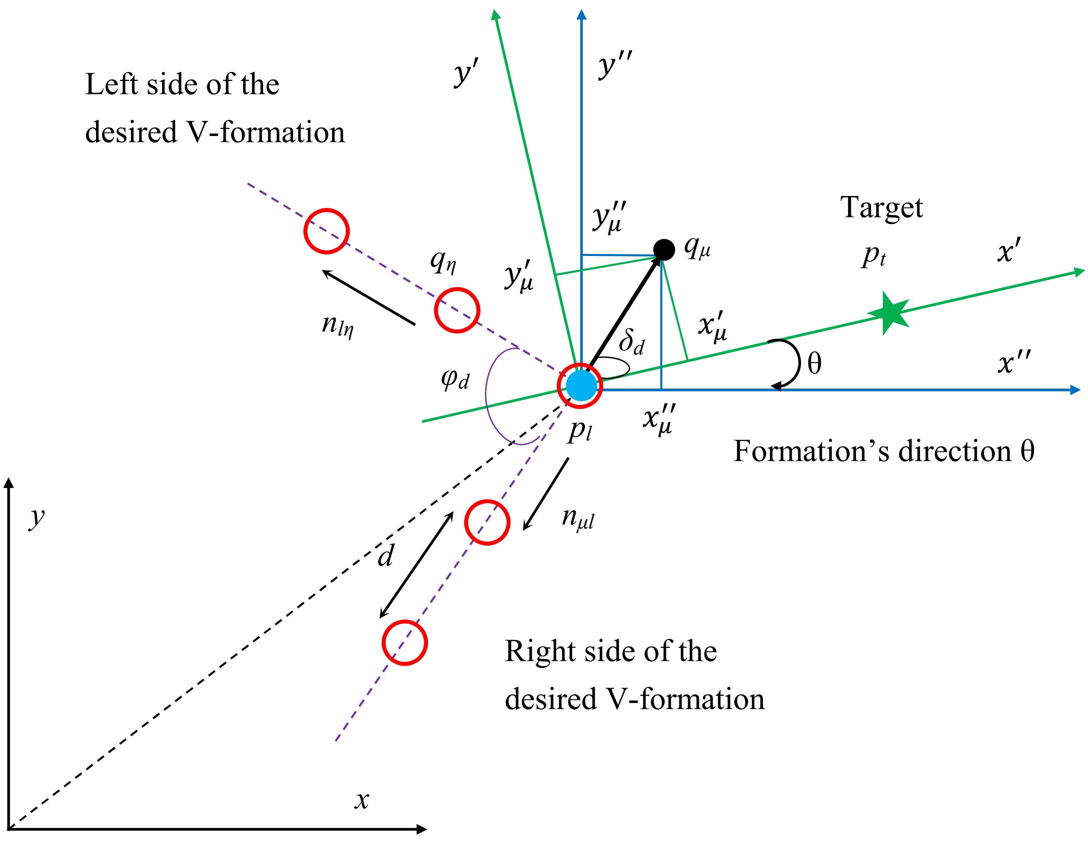

Figure 3.6: The description of the method to build the V-shape desired formation.

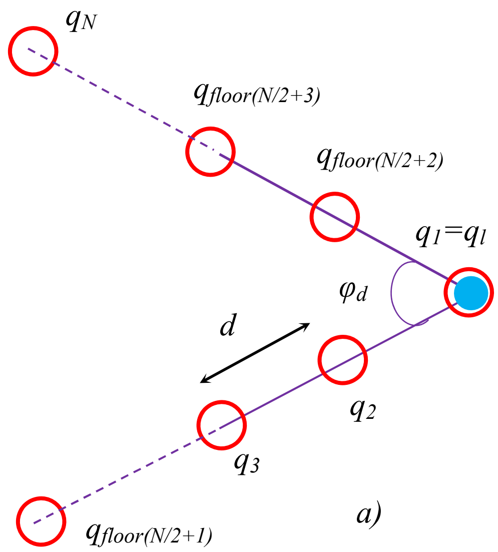

Figure 3.7: The description of the distributed virtual nodes *j* *(j=1,2,..,N)* in the V-shape desired formation.

In some real cases, such as under the influence of the environment (noises, wind, obstacle avoidance, etc.,), the formation angle needs to change in order to adapt to the effect of this environment. Thus, generally, we propose the formation angel as follows:

$$\varphi(t) = \varphi_d + \varepsilon_1 \phi(t). \tag{3.14}$$

In this equation, ε_1 is a positive factor, and $\phi(t)$ is used as a sensing function that decides the formation angle $\varphi(t)$. However, this formation angle $\varphi(t)$ must guarantee that there are no collisions among the members in the formation. In other words, it depends on the repulsive radius around each robot. Hence, the smallest formation angle is computed as $\varphi_{dmin} = arccos(1 - r_r^2 / 2d^2)$.

3.3.3 Circular desired formation

As presented above, the circular desired formation is used to encircle the moving target when the distance between the leader and the target is shorter than the target approaching radius $d_l^t = \|p_l - p_t\| \leq r^t$. Hence, this desired formation is designed based on the relative position between the target and the leader as in figure 3.8. The position of the virtual node *j* (*j=1,2..,N*) on the circle that has the central point at the target's position p_t and the radius $d_l^t = \|p_l - p_t\|$ is computed as follows:

$$q_j = p_t + Rq'_j, \qquad j = 1,2..,N. \tag{3.15}$$

In equation (3.15), the rotation matrix R is determined similarly to equations (3.5). The position of the virtual node j on the coordinate system $x'y'$ ($q'_j = (x'_j, y'_j)^T$) is computed as follows:

$$\begin{pmatrix} x'_j \\ y'_j \end{pmatrix} = d_l^t \begin{pmatrix} cos(2j\pi/N) \\ sin(2j\pi/N) \end{pmatrix}, \quad j = 1,2..,N. \tag{3.16}$$

Substitute equation (3.16) into (3.15) and let the virtual node that is being owned by the leader be the first position in the circular desired formation, we have the circular formation of the virtual nodes j as follows:

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \end{pmatrix} + Rd_l^t \begin{pmatrix} cos(2\zeta\pi/N) \\ sin(2\zeta\pi/N) \end{pmatrix}, \quad j = 1,2..N; \zeta = j-1. \tag{3.17}$$

Equation (3.17) shows that the distributed virtual nodes j *(j=1,2,..,N)* on the circular desired formation are equidistant from each other, and the distance from them to the target is $d_l^t = \|p_l - p_t\|$, see figure 3.9. However, in order to avoid the collision between the active neighboring members, the distance d_l^t must satisfy $r_{min}^\tau \le d_l^t$, here $r_{min}^\tau = r_r / \sqrt{2}\sin(4\pi/N)$ is the allowed minimum radius of the desired circular shape formation.

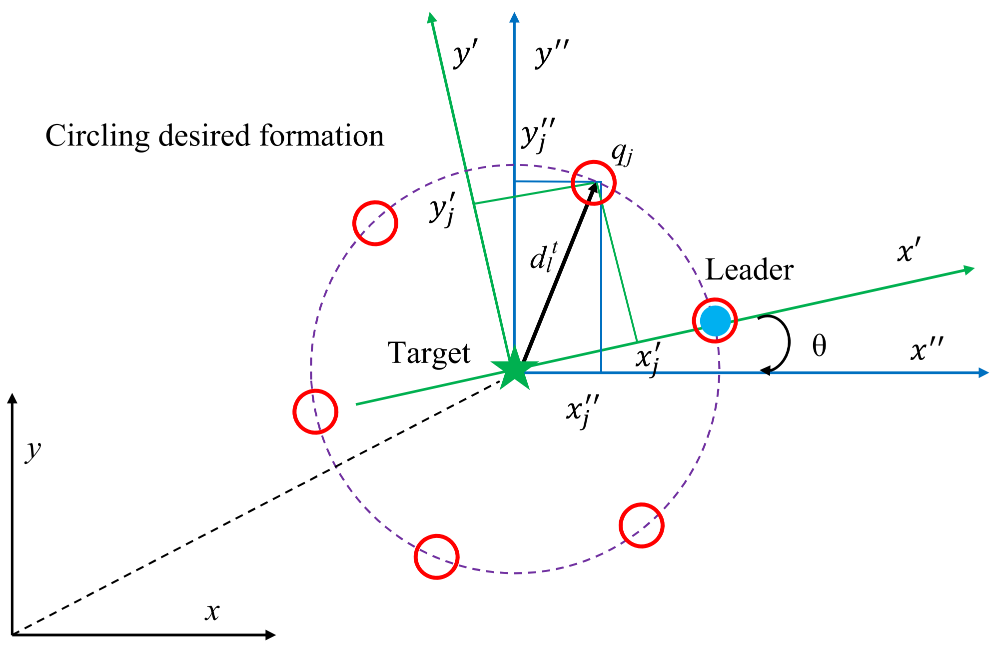

Figure 3.8: The description of the method to build the circular desired formation.

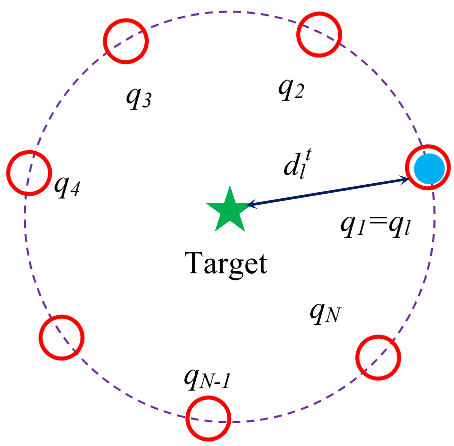

Figure 3.9: The description of the distributed virtual nodes j $(j=1,2,..,N)$ in the desired circular shape formation.

3.4 Formation adaptation while tracking a moving target

This section presents an approach for the adaptive formation control of autonomous robots following the given desired shapes (V-shape or circular shape) under the influence of the environment, such as noises, obstacles, etc. Firstly, the proposed control algorithms have to guarantee that: The motion of robots always converges to the created virtual nodes in the desired formation under the effects of the dynamic environment. While tracking the moving target, the stability of the formation must be maintained, and there are no collisions among members. Additionally, robots must also automatically escape the obstacles in order to continue to track the moving target with their swarm. Secondly, the adaptation of formation is decided by the target position sense of the leader. Finally, simulation results are provided to verify the effectiveness of the proposed algorithms.

3.4.1 Formation connection control algorithm

A. No influence of the noises

Firstly, surrounding the virtual nodes j ($j=1,2,\ldots N$) of the desired formation, the attractive force fields are created to drive the free robots towards the desired positions in the desired formation. Then, these free robots will occupy these desired positions, and become the active robots. The tracking task is to make the distance $d_i^j = \left\| p_i - q_j \right\|$ approach to zero as fast as possible. This means that $\lim_{t\to\infty}(p_i(t) - q_j(t)) = 0$ and $\lim_{t\to\infty}(v_i(t) - v_j(t)) = 0$.

Algorithm 3.1: Reaching the desired position at the virtual nodes in the desired formation.

Consider: A robot i and virtual nodes j ($i,j=1,..,N,\ i\neq l$). Determine the shortest distance from p_i to all the virtual nodes q_j and the scalar factor c_i^j given as follows: $d_i^{j\mathrm{m1}} = \min\{d_i^j = \|p_i - q_j\|,\ j = 1,..,N\}$, $c_i^j = 1$ if q_j is active, $c_i^j = 0$ if q_j is free.

if $d_i^{j\mathrm{ml}} \le r_a$ & $c_i^{j\mathrm{ml-1}} = 1$ **then**

$$u_i^j = -k_{ip1}^j(p_i - q_{jml}) - k_{iv}^j(v_i - v_{jml}) + \dot{v}_{jml}$$

else if $d_i^{j\mathrm{ml}} \le r_a$ & $c_i^{j\mathrm{ml-1}} = 0$ **then**

$$u_i^j = -k_{ip2}^j(p_i - q_{jml-1}) - k_{iv}^j(v_i - v_{jml-1}) + \dot{v}_{jml-1}$$

else if $d_i^{j\mathrm{ml}} > r_a$ **then**

if $c_i^{j\mathrm{ml}} = 0$ **then**

$$u_i^j = -k_{ip3}^j(p_i - q_{jml}) / \|p_i - q_{jml}\| - k_{iv}^j(v_i - v_{jml}) + \dot{v}_{jml}$$

else

Determine the shortest distance from p_i to the free virtual nodes q_j in the desired formation as:

$$d_i^{jm2} = \min\{d_i^j = \|p_i - q_j\|, c_i^j = 0,\ j = 1,..,N,\ j \neq jm1\}$$

$$u_i^j = -k_{ip3}^j(p_i - q_{jm2}) / \|p_i - q_{jm2}\| - k_{iv}^j(v_i - v_{jm2}) + \dot{v}_{jm2}$$

end

end

Based on the above analyze, the formation control law for formation connection is proposed as in Algorithm 1. Where, $k_{ip1}^j, k_{ip2}^j, k_{ip3}^j, k_{iv}^j$, $(p_i - q_j)$ and $(v_i - v_j)$ are the positive gain factors, the relative position vector, the relative velocity vector between the robot i and the virtual node j, respectively. In this algorithm, we use two potential fields $f_{1i}^j = -k_{ip3}^j(p_i - q_j)/\|p_i - q_j\|$ and $f_{2i}^j = -k_{ip1}^j(p_i - q_j)$ as the artificial attractive forces. The constant potential field f_{1i}^j is used to drive the free robots towards the desired formation, while the linear potential field f_{2i}^j is used to control the active robots approaching to the virtual nodes of the desired formation. Additionally, the component $-k_{iv}^{j}(v_i - v_j)$ is is also utilized as the damping term. Therefore, using the desired position finding algorithm (Algorithm 3.1) robots can quickly approach their desired position at the virtual nodes in the desired formation.

Theorem 3.1. *Consider the active robot i with its dynamic model (3.1) and control input u_i^j given in Algorithm 3.1 at the active node j in the desired formation. If the velocity of the node j is smaller than the maximum velocity of the robot i, and the node j-1 is also active, then the system (3.1) will be stable at the equilibrium point, at which $p_i = q_j$ and $v_i = v_j$ for all i and j.*

Proof of theorem 3.1

In order to analyze the stability of the robot *i* at the active node *j* when the node *j-1* is also active, we rewrite the control law u_i^j as follow:

$$u_i^j = f_{2i}^j - k_{iv}^j(v_i - v_j) + \dot{v}_j. \tag{3.18}$$

Consider the vector field $f_{2i}^j = -k_{ip1}^j(p_i - q_j) = (P_i, Q_i, 0)^T$, here $P_i = -k_{ip1}^j(x_i - x_j)$, $Q_i = -k_{ip1}^j(y_i - y_j)$, and $Z_i = 0$. According to [96], [97], we obtain:

$$\text{curl } (f_{2i}^j) = \left(\frac{\partial Z_i}{\partial y_i} - \frac{\partial Q_i}{\partial z_i},\ \frac{\partial P_i}{\partial z_i} - \frac{\partial Z_i}{\partial x_i},\ \frac{\partial Q_i}{\partial x_i} - \frac{\partial P_i}{\partial y_i}\right) = (0,\ 0,\ 0). \tag{3.19}$$

Equation (3.19) shows that the vector field f_{2i}^j is irrotational. Now, we consider the scale function as follows:

$$V_i^j = \frac{1}{2}k_{ip1}^j\left(p_i - q_j\right)^T\left(p_i - q_j\right). \tag{3.20}$$

Taking the negative gradient of the function V_i^j we obtain:

$$\begin{aligned} -\nabla V_i^j &= -\nabla\left(\frac{1}{2}k_{ip1}^j\left(p_i - q_j\right)^T\left(p_i - q_j\right)\right) \\ &= -\nabla\left(\frac{1}{2}k_{ip1}^j\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right)\right) \end{aligned} \tag{3.21}$$

$$= \begin{pmatrix} -\frac{1}{2} k_{ip1}^{j} \frac{\partial}{\partial x_i}\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right) \\ -\frac{1}{2} k_{ip1}^{j} \frac{\partial}{\partial y_i}\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right) \end{pmatrix}^T$$

$$= \begin{pmatrix} -k_{ip1}^{j}\left(x_i - x_j\right) \\ -k_{ip1}^{j}\left(y_i - y_j\right) \end{pmatrix}^T$$

$$= -k_{ip1}^{j}(p_i - q_j) = f_{2i}^{j} \ .$$

So, (3.19) and (3.21) show that the vector field f_{2i}^{j} is a potential field, and its potential function is V_i^j.

Let $x_1 = p_i - q_j,\ x_2 = v_i - v_j$ be the relative position and velocity of the robot i and node j. We have the error dynamic model of the system as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u_i^j - \dot{v}_j \quad i, j = 1, 2.., N. \end{aligned} \tag{3.22}$$

Substitute u_i^j in (3.18) into (3.22) we obtain the error dynamic model as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\nabla V_i^j - k_{iv}^{j}(v_i - v_j). \end{aligned} \tag{3.23}$$

To analyze the stability of model (3.23) at the equilibrium position $(p_i - q_j = 0,\ v_i - v_j = 0)$, the positive definite function is selected as follows:

$$V_f = V_i^j + \frac{1}{2} x_2^T x_2. \tag{3.24}$$

Consider the potential function (3.20) we have the relationship as follows $\left(\partial V_i^j / \partial p_i\right)^T =$ $= \left(\partial V_i^j / \partial (p_i - q_j)\right)^T$. Taking the time derivative of (3.24) along the trajectory of the system (3.23), we obtain:

$$\begin{aligned} \dot{V}_f(t) &= (\nabla V_i^j)^T \dot{x}_1 + x_2^T \dot{x}_2 = x_2^T \left(\nabla V_i^j + \dot{x}_2\right) \\ &= x_2^T \left(\nabla V_i^j - \nabla V_i^j - k_{iv}^{j} x_2\right) \\ &= -k_{iv}^{j} x_2^T x_2 \le 0. \end{aligned} \tag{3.25}$$

Equation (3.25) shows that the selected positive definite function V_f is a Lyapunov function [95], which guarantees that the system (3.22) is stable at the equilibrium point $(p_i = q_j, v_i = v_j)$.

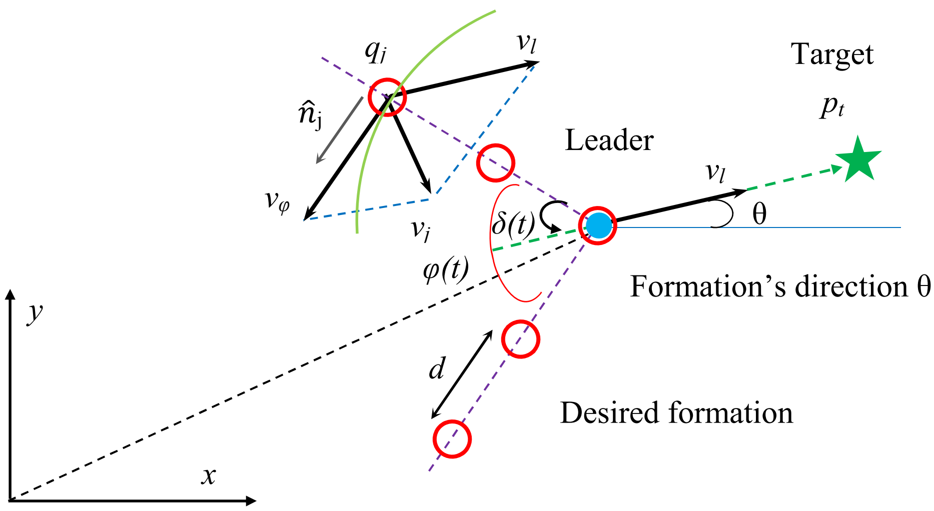

Figure 3.10: The description of the velocity of the node j under the influence of the leader's velocity v_l and the angular velocity $\dot{\delta}(t)$.

Now, in order to allow the robot i to approach to the node j as fast as possible, we use the attractive force from the node j proportional with its velocity. This means that factor k_{ip1}^{j} in Algorithm 1 depends on the velocity v_j. This factor is given as

$$k_{ip1}^{j} = k_{ipd}^{j} + \varepsilon_2 \left\| v_j \right\|. \tag{3.26}$$

In equation (3.26), k_{ipd}^{j}, ε_2 are the positive constants. Moreover, while tracking a moving target, the velocity of the node j is also dependent on the velocity of the leader v_l and the formation angle $\varphi(t)$ (see figure 3.10). This relation is calculated as

$$v_j = v_l + v_\varphi. \tag{3.27}$$

In this equation, v_φ is the robot's velocity when the formation angle $\varphi(t)$ changes, and it is computed as follows:

$$v_\varphi = \dot{\delta}(t)\, d_j^l \hat{n}_j. \tag{3.28}$$

Where, the angular velocity of the formation angle $\varphi(t)/2$ is $\dot{\delta}(t)=\partial(\varphi(t)/2)/\partial t$. The distance between the leader and the node j is $d_j^l=\|q_j-p_l\|$. The unit vector $\hat{n}_j$, which is perpendicular with the vector (q_j-p_l), is given as:

$$\hat{n}_j=\hat{c}_i\left(-(y_j - y_l)/d_j^l,\quad (x_j - x_l)/d_j^l\right)^T. \tag{3.29}$$

Where, the scalar $\hat{c}_i$ is dependent on the rotational direction of the formation angle. Consider the robots on the left side of the V-shape formation (see Fig.7), the scalar $\hat{c}_i$ is defined as:

$$\hat{c}_i=\begin{cases}1 & \text{if } \varphi(t)<\varphi_d\\ -1 & \text{otherwise.}\end{cases} \tag{3.30}$$

Similarly, for the robots on the right side of the V-shape formation the scalar $\hat{c}_i$ is also defined as:

$$\hat{c}_i=\begin{cases}1 & \text{if } \varphi(t)>\varphi_d\\ -1 & \text{otherwise.}\end{cases} \tag{3.31}$$

B. Influence of the noises

To consider the stability of the formation under the influence of the noises, which cause the position errors between the robot i and the virtual node j, we assume that the estimates of the position and the velocity of the robot i are: $\hat{p}_i=p_i+z_{pi}$ and $\hat{v}_i=v_i+z_{vi}$, where z_{pi} and z_{vi} are the position and velocity measurement errors of the robot i, respectively. Similarly, the estimates of the position and the velocity of the virtual node j are also defined as: $\hat{q}_j=q_j+z_{qj}$ and $\hat{v}_j=v_j+z_{vj}$, where z_{qj} and z_{vj} are the position and velocity noises of the node j, respectively. Now, we propose the new control law for the robot i at the active node j in noisy environment as:

$$u_i^j=-\hat{k}_{ip1}^j(\hat{p}_i-\hat{q}_j)-\hat{k}_{iv}^j(\hat{v}_i-\hat{v}_j)+\dot{v}_j-\dot{z}_{vij}. \tag{3.32}$$

Where $\hat{k}_{ip1}^j=k_{ipd}^j+\varepsilon_2\|\hat{v}_j\|$ and $\hat{k}_{iv}^j$ are the positive factors. Let $\hat{x}_1=\hat{p}_i-\hat{q}_j=p_i-q_j+z_{pij}$ and $\hat{x}_2=\hat{v}_i-\hat{v}_j=v_i-v_j+z_{vij}$ be the relative position and velocity of the robot i and node j in noisy environment, here $z_{vij}=z_{vi}-z_{vj}$ and $z_{pij}=z_{pi}-z_{qj}$. We have the error dynamic of the system as:

$$\dot{\hat{x}}_1 = \hat{x}_2$$
$$\dot{\hat{x}}_2 = \dot{v}_i - \dot{v}_j + \dot{z}_{vij}. \tag{3.33}$$

However, in order to guarantee that the active neighboring robots while moving in a formation do not repel, the noise z_{pi} must satisfy $\|z_{pi}\| < r_{n1}$, here r_{nl} is a noise radius. This noise radius can be selected as depicted in figure 3.11a, such that: $r_{n1} = \lambda/2$, here the positive constant $\lambda = d - r_r$ as shown in figure 3.11a. Moreover, the noise's amplitude must also guarantee that robots do not collide to each other during movement. Thus, we can choose another noise radius $r_{n2} = \lambda^*/2$, see figure 3.11b, here $\lambda^* > 0$ is a region used to detect the collision among robots. Finally, in order to solve both above conditions the noise z_{pi} has to satisfy $\|z_{pi}\| < \min(r_{n1},\ r_{n2})$.

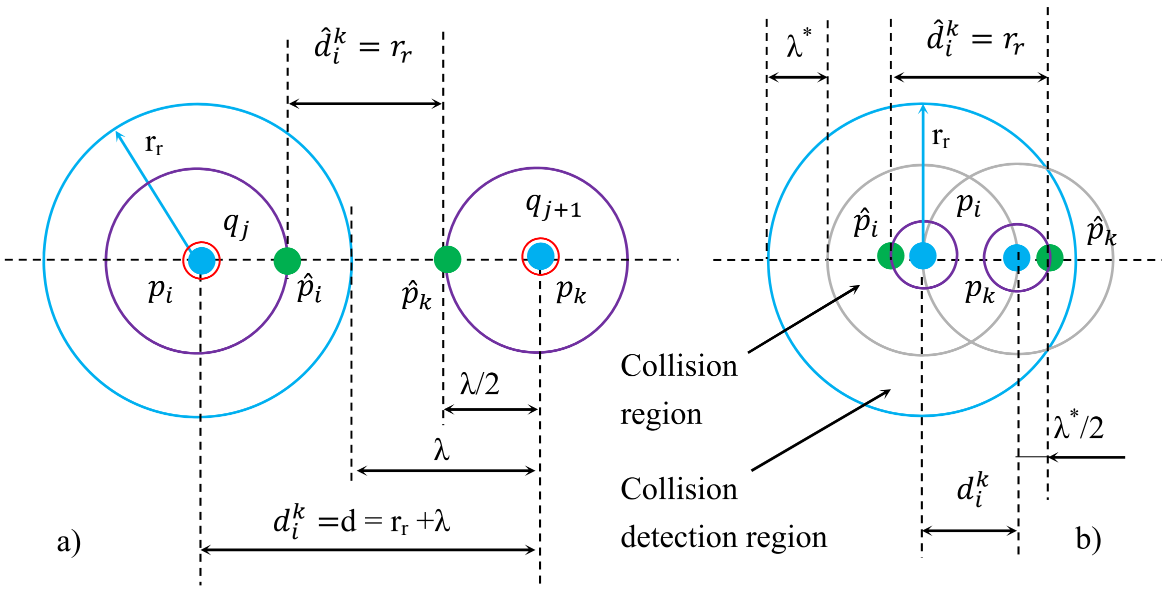

Figure 3.11: Description the noise's boundary guarantees that there is no effect to each other between two active neighboring robots *i* and *k* (a), and there is no collision between robots *i* and *k* (b). Where d_i^k and $\hat{d}_i^k$ are the actual distance and the estimate distance between robots, respectively.

Proposition 3.1. *Consider the active robot i with its dynamic model (3.1) and control input* u_i^j *given as (3.32) at the active node j of the desired formation in noisy environment. If the velocity of the node j is smaller than the maximum velocity of the robot i, and the node j-1 is also active, and the noise is bounded by* $\|z_{pi}\| < \min(\lambda/2, \lambda^*/2)$, *then the system (3.33) is stable at the equilibrium state* $(\hat{p}_i = \hat{q}_j, \hat{v}_i = \hat{v}_j)$ *for all i and j.*

Proof of proposition 3.1

Similar to the proof of theorem 1, to analyze the stability of the system (3.33) we choose the Lyapunov function as:

$$V_z = \frac{1}{2}\hat{k}_{ip}^j \hat{x}_1^T \hat{x}_1 + \frac{1}{2}\hat{x}_2^T \hat{x}_2 \ . \tag{3.34}$$

Taking the time derivative of equation (3.34) along the trajectory of the system (3.33), we obtained as follows:

$$\dot{V}_z(t) = \hat{k}_{ip}^j \hat{x}_1^T \dot{\hat{x}}_1 + \hat{x}_2^T \dot{\hat{x}}_2 = \hat{x}_2^T(\hat{k}_{ip}^j \hat{x}_1^T + \dot{\hat{x}}_2) \tag{3.35}$$

Substitute $\dot{\hat{x}}_2$ in (3.33) into (3.35) we obtain:

$$\dot{V}_z(t) = -\hat{k}_{iv}^j \hat{x}_2^T \hat{x}_2 \leq 0. \tag{3.36}$$

So, equation (3.36) shows that the system (3.33) is stable with the control law (3). However, this stability is limited by the boundary of the noise. If $\|z_{pi}\| \geq \min\left(\lambda/2, \lambda^*/2\right)$, then the active neighboring robots can repel to each other or the robots can collide, so the stability is broken, see figure 3.11.

3.4.2 Collision avoidance control algorithm

This section presents a method for the collision avoidance among the robots during movement based on the artificial repulsive potential field. Let $N_i^k(t)$ be the set of the robots in the neighborhood of the robot *i* at time *t*, such that:

$$N_i^k(t) = \left\{\forall k : d_i^k = \|p_i - p_k\| \leq r_r, k \in \{1,..N\}, k \neq i\right\}. \tag{3.37}$$

Where, r_r is the repulsive radius surrounding each robot, and d_i^k is the Euclidean distance between robot *k* and robot *i*. Now, in order to avoid the collision between robots *i* and *k* (*i, k=1,2,..,N*; *i≠k, i≠l*), the local repulsive force field is created surrounding each robot within the repulsive radius r_r as:

$$f_i^k = \left(\left(\frac{1}{d_i^k} - \frac{1}{r_r}\right)\frac{k_{i1}^k}{(d_i^k)^2} - k_{i2}^k\left(d_i^k - r_r\right)\right)c_i^k n_i^k. \tag{3.38}$$

Where, the positive factors k_{i1}^{k}, k_{i2}^{k} are used to control the fast interaction. The unit vector n_i^k from robot k to robot i is given as $n_i^k = (p_i - p_k)/\|p_i - p_k\|$. The scalar c_i^k is defined as follows:

$$c_i^k = \begin{cases} 1 & \text{if } k \in N_i^k(t) \\ 0 & \text{otherwise.} \end{cases} \tag{3.39}$$

The algorithm for the collision avoidance is built based on the repulsive vector field (3.38) combined with the relative velocity vector $k_{iv}^k(v_i - v_k)$ between robot k and robot i as follows:

$$u_i^k = \sum_{k=1,k\neq i}^{N} \left(f_i^k - c_i^k k_{iv}^k (v_i - v_k) \right). \tag{3.40}$$

The controller (3.40) shows that the neighboring robots are always driven to leave each other. In other words, this controller guarantees that there are no collisions among robots in the swarm.

3.4.3 Obstacle avoidance control algorithm

This sub-section presents the control algorithm for robots passing through M obstacles to track a moving target. As analyzed in chapter 2, this obstacle avoidance control algorithm is also proposed as follows:

$$u_i^o = \sum_{o=1,o\neq k}^{M} \left(f_i^{op} + f_i^{or} + k_{iv}^o c_i^o (v_i - v_o) \right). \tag{3.41}$$

Where, the linear repulsive force field f_i^{op} and the rotational force field f_i^{or} surrounding obstacles are presented in chapter 2. The relative velocity vector $(v_i - v_o)$ between the robot i and its neighbor-obstacle o is used as a damping term with the damping scaling factor k_{iv}^o. Similar to definition (3.37), here we can also define the set of the obstacles in the neighborhood of the robot i at time t as follows:

$$N_i^o(t) = \left\{ \forall o : d_i^o \leq r^\beta,\ o \in \{1,..M\}, o \neq k \right\}. \tag{3.42}$$

In (3.42), $r^\beta > 0$ and $d_i^o = \|p_i - p_o\|$ are the obstacle detection range and the Euclidean distance between the robot i and the obstacle o, respectively. The scalar c_i^o is defined as:

$$c_i^o = \begin{cases} 1 & \text{if } o \in N_i^o(t) \\ 0 & \text{otherwise.} \end{cases} \tag{3.43}$$

As presented in chapter 2, using the control law (3.41) robots can easily escape obstacles to continue to track the target.

3.4.4 Target tracking control algorithm

A. Leader selection

Firstly, one robot, which is closest to the target, is selected as the leader in order to generate the desired formation. Then, this leader is saved to lead its formation to track a moving target. In case, the leader meets risk, such as it is broken or trapped in obstacles, it must transfer its leadership to another, and becomes a free member as other free robots in the swarm. The leader is selected as Algorithm 3.2.

Algorithm 3.2: Leader selection

Update data: The actual position of robots p_i ($i=1,..,N,\ i\neq l$), obstacle's information, the target's position p_t, the actual position of the leader ($p_\xi = p_l$).

if time $t=0$ (at initial time) **then**

  Compute the shortest distance from the robot p_i to the target p_t in order to determine the leader as follows:

$$d^t_{imin1} = \min\{\|p_i - p_t\|, i = 1,..,N\};\ p_l = p_{imin1}$$

else

  if the actual leader meets obstacle or is broken **then**

    Leadership is transferred to other member that is free and has the closest distance to the target.

$$d^t_{imin2} = \min\{\|p_i - p_t\|, i = 1,.,N, i \neq \xi, \text{free}\};\ p_l = p_{imin2}$$

  else

    Maintain the leadership of the actual leader.

$$p_l = p_\xi$$

  end

end

B. No influence of the noises

The target tracking controller, which is designed based on the relative position between the leader and the target, has to guarantee that the formation's motion is always driven towards the target. The tracking task is to make the distance between the leader and the target $d_l^t = \|p_l - p_t\|$ approaching to the radius of the desired circular formation r^r as fast as possible. This means that $\lim_{t\to\infty}(v_l(t) - v_t(t)) = 0$ and $\lim_{t\to\infty}(p_l(t) - p_t(t)) = r^r(p_l(t) - p_t(t))/d_l^t$. Based on the above analysis, the control law for the target tracking is proposed as follows:

$$u_l^t = f_l^t - k_{lv}^t(v_l - v_t) + \dot{v}_t. \tag{3.44}$$

Where, k_{lv}^t and $\dot{v}_t$ are the positive factor and the acceleration of the target, respectively. $(v_l - v_t)$ is the relative velocity vector between the leader and the target. The potential field f_l^t from the target is used to drive the leader moving towards the target, and it is given as:

$$f_l^t = \begin{cases} \left(\left(\dfrac{1}{d_l^t} - \dfrac{1}{r^\tau}\right)\dfrac{k_{l2}^t}{(d_l^t)^2} - \dfrac{k_{l1}^t(d_l^t - r^\tau)}{(r^t - r^\tau)}\right)n_l^t, & \text{if } d_l^t < r^t \\ -k_{l1}^t n_i^t, & \text{otherwise.} \end{cases} \tag{3.45}$$

Where, k_{l2}^t, r^t and $r_{min}^t < r^\tau < r^t$ are the positive constant, the target approaching radius, and the desired radius of the circular formation, respectively. $k_{l1}^t = k_{l1d}^t + \varepsilon_3\|v_t\|$, here k_{l1d}^t, ε_3 are the positive constants. The unit vector along the line connection from the target to the leader is computed as $n_l^t = (p_l - p_t)/\|p_l - p_t\|$. In the equation (3.45), the constant attractive force $f_{2l}^t = -k_{l1}^t n_i^j$ is sued to track the target when $d_l^t > r^t$. On the other hand, the attractive/repulsive force field $f_{1l}^t = (k_{l2}^t(1/d_l^t - 1/r^\tau)/(d_l^t)^2 - k_{l1}^t(d_l^t - r^\tau)/(r^t - r^\tau))n_l^t$ surrounding the equilibrium position, at which $\|p_l - p_t\| = r^\tau$, $(v_l - v_t) = 0$ is used to encircle the target when $d_l^t \le r^t$. Hence, using this combined vector field the leader can easily approach to the target at the equilibrium position.

Theorem 3.2. *Consider the leader l is described by the model (3.1) and controlled by the control law (3.44) when* $d_l^t \le r^t$. *If the velocity of the target is smaller than the maximum velocity of the leader, then the system (3.1) will be stable at the equilibrium state, at which* $v_l = v_t$ *and* $(p_l - p_t) = r^\tau(p_l - p_t)/\|p_l - p_t\|$.

Proof of theorem 3.2

Firstly, we consider the vector field $f_{ll}^t = K_l(p_l - p_t)/\|p_l - p_t\|$ in (3.45). Let $K_l = (k_{l2}^t/d_l^t - k_{l2}^t/r^\tau)/(d_l^t)^2 - k_{l1}^t(d_l^t - r^\tau)/(r^t - r^\tau)$, $P_l = K_l(x_l - x_t)/\|p_l - p_t\|$, $Z_l = 0$ and $Q_l = K_l(y_l - y_t)/\|p_l - p_t\|$, we have:

$$\text{curl}\ (f_{ll}^t) = \left(\frac{\partial Z_l}{\partial y_l} - \frac{\partial Q_l}{\partial z_l},\ \frac{\partial P_l}{\partial z_l} - \frac{\partial Z_l}{\partial x_l},\ \frac{\partial Q_l}{\partial x_l} - \frac{\partial P_l}{\partial y_l}\right)^{\mathrm{T}} = 0. \tag{3.46}$$

Equation (3.46) shows that the vector field f_{ll}^t is not rotational.

Consider the scale function as follow:

$$V_l^t = \frac{1}{2}\left(k_{l2}^t\left(\frac{1}{d_l^t} - \frac{1}{r^\tau}\right)^2 + \frac{k_{l1}^t(d_l^t - r^\tau)^2}{(r^t - r^\tau)}\right). \tag{3.47}$$

Taking the negative gradient of V_i^t we obtain:

$$\begin{aligned}
f_{ll}^t &= -\nabla V_l^t \\
&= -\nabla\left(\frac{k_{l2}^t}{2}\left(\frac{1}{d_l^t} - \frac{1}{r^\tau}\right)^2 + \frac{k_{l1}^t(d_l^t - r^\tau)^2}{2(r^t - r^\tau)}\right) \\
&= -\left(\frac{k_{l2}^t}{d_l^t} - \frac{k_{l2}^t}{r^\tau}\right)\nabla\left(\frac{1}{d_l^t} - \frac{1}{r^\tau}\right) - \frac{k_{l1}^t(d_l^t - r^\tau)}{(r^t - r^\tau)}\nabla(d_l^t - r^\tau) \\
&= \left(\frac{k_{l2}^t}{d_l^t} - \frac{k_{l2}^t}{r^\tau}\right)\frac{1}{(d_l^t)^2}\nabla d_l^t - \frac{k_{l1}^t(d_l^t - r^\tau)}{(r^t - r^\tau)}\nabla d_l^t \\
&= \left(\left(\frac{1}{d_l^t} - \frac{1}{r^\tau}\right)\frac{k_{l2}^t}{(d_l^t)^2} - \frac{k_{l1}^t(d_l^t - r^\tau)}{(r^t - r^\tau)}\right)\nabla d_l^t.
\end{aligned} \tag{3.48}$$

Similar to equation (3.21), here we also obtain a gradient field $\nabla d_l^t = (p_l - p_t)/\|p_l - p_t\|$. So, (3.46) and (3.48) show that the vector field f_{ll}^t is also a potential field, and its potential function is V_l^t.

In order to analyze the stability of the leader at the equilibrium position, at which $\|p_l - p_t\| = r^r$ and $(v_l - v_t) = 0$, we let $\breve{x}_1$ be the position error between the vector $(p_l - p_t)$ and the vector $r^r(p_l - p_t)/\|p_l - p_t\|$, and $\breve{x}_2$ be the relative velocity between the leader and the target. We get: $\breve{x}_1 = \lambda_l^t(p_l - p_t)$ and $\breve{x}_2 = (v_l - v_t)$, here $\lambda_l^t = (1 - r^r/\|p_l - p_t\|)$. The error dynamic of the system is described as follows:

$$\dot{\breve{x}}_1 = \lambda_l^t \breve{x}_2$$
$$\dot{\breve{x}}_2 = \dot{v}_l - \dot{v}_t. \tag{3.49}$$

Substitute (3.44) into (3.49), we obtain:

$$\dot{\breve{x}}_1 = \lambda_l^t \breve{x}_2$$
$$\dot{\breve{x}}_2 = -\nabla V_i^t - k_{lv}^t \breve{x}_2. \tag{3.50}$$

To analyze the stability of the system (3.50), we choose the Lyapunov function as follows:

$$V_\tau = V_l^t + \frac{1}{2}\breve{x}_2^T \breve{x}_2. \tag{3.51}$$

Let $\breve{x}_1^* = (p_l - p_t) = \breve{x}_1 / \lambda_l^t$ be the position error between the leader and the target, and note that the relation in (3.47) $\left(\partial V_l^t / \partial p_t\right)^T = \left(\partial V_l^t / \partial (p_l - p_t)\right)^T$. Taking the time derivative of equation (3.51) along the trajectory of the system (3.50) we obtain:

$$\begin{aligned} \dot{V}_\tau(t) &= \left(\nabla V_i^t\right)^T \dot{\breve{x}}_1^* + \breve{x}_2^T \dot{\breve{x}}_2 = \frac{1}{\lambda_l^t}\left(\nabla V_i^t\right)^T \dot{\breve{x}}_1 + \breve{x}_2^T \dot{\breve{x}}_2 \\ &= \breve{x}_2^T(\nabla V_i^t + \dot{\breve{x}}_2) = -k_{lv}^t \breve{x}_2^T \breve{x}_2 \le 0. \end{aligned} \tag{3.52}$$

So, this equation guarantees that system (3.50) is stable at the equilibrium position when using the control law (3.44). In other words, using the controller (3.44) the leader will be driven towards the desired position, at which the distance between leader and the target is equal to the radius of the desired circular formation.

C. Influence of the noises

Now, we consider the influence of the noisy environment to the stability of the leader at the equilibrium position. Firstly, we assume that the estimates of the position and the velocity of the leader are: $\tilde{p}_l = p_l + z_{pl}$ and $\tilde{v}_l = v_l + z_{vl}$, where z_{pl} and z_{vl} are position and velocity noises of the leader, respectively. Similarly, the estimates of the position and the velocity of the target are also defined as: $\tilde{p}_t = p_t + z_{pt}$ and $\tilde{v}_t = v_t + z_{vt}$, where z_{pt} and z_{vt} are position and velocity noises of the target, respectively. Then we have: $(\tilde{p}_l - \tilde{p}_t) = (p_l - p_t + z_{plt})$ and $(\tilde{v}_l - \tilde{v}_t) = (v_l - v_t + z_{vlt})$, here $z_{plt} = z_{pl} - z_{qt}$ and $z_{vlt} = z_{vl} - z_{vt}$. Based on these definitions, we propose the new control law for the leader's tracking in noisy environment as follows:

$$u_l^t = \tilde{f}_l^t - \tilde{k}_{lv}^t(\tilde{v}_l - \tilde{v}_t) + \dot{v}_t - \dot{z}_{vlt}. \tag{3.53}$$

Where, $\tilde{k}_{lv}^t$ is a positive factor. The force field $\tilde{f}_l^t$ is given as:

$$\tilde{f}_l^t = \begin{cases} \left(\left(\dfrac{1}{\tilde{d}_l^t} - \dfrac{1}{r^\tau} \right) \dfrac{\tilde{k}_{l2}^t}{(\tilde{d}_l^t)^2} - \dfrac{\tilde{k}_{l1}^t(\tilde{d}_l^t - r^\tau)}{(r^t - r^\tau)} \right) \tilde{n}_l^t, \text{ if } \tilde{d}_l^t < r^t \\ -\tilde{k}_{l1}^t \tilde{n}_i^t \text{ , } otherwise. \end{cases} \tag{3.54}$$

In this equation, $\tilde{k}_{l1}^t = k_{l1d}^t + \varepsilon_3 \|\tilde{v}_t\|$, $\tilde{k}_{l2}^t$, and $\tilde{d}_l^t = \|\tilde{p}_l - \tilde{p}_t\|$ are the positive factors, and the Euclidean distance, respectively. The unit vector $\tilde{n}_l^t$ is computed as $\tilde{n}_l^t = (\tilde{p}_l - \tilde{p}_t) / \|\tilde{p}_l - \tilde{p}_t\|$.

Let $\tilde{x}_1$ be the position error between the vector $(\tilde{p}_l - \tilde{p}_t)$ and the vector $r^\tau(\tilde{p}_l - \tilde{p}_t) / \|\tilde{p}_l - \tilde{p}_t\|$, and $\tilde{x}_2$ be the relative velocity between the leader and the target. We get: $\tilde{x}_1 = (\tilde{p}_l - \tilde{p}_t) - r^r(\tilde{p}_l - \tilde{p}_t) / \|\tilde{p}_l - \tilde{p}_t\| = \lambda_l^t(p_l - p_t + z_{plt})$, $\tilde{x}_2 = (\tilde{v}_l - \tilde{v}_t) = (v_l - v_t + z_{vlt})$,here $\lambda_l^t = 1 - r^\tau / \|\tilde{p}_l - \tilde{p}_t\|$, and the error dynamic of the system is given as:

$$\begin{aligned} \dot{\tilde{x}}_1 &= \lambda_l^t \tilde{x}_2 \\ \dot{\tilde{x}}_2 &= \dot{v}_l - \dot{v}_t + \dot{z}_{vlt}. \end{aligned} \tag{3.55}$$

Proposition 3.2. *Consider the leader l is described by the model (3.1) and controlled by the control law (3.53) in noisy environment when* $d_l^t \le r^t$. *If the velocity of the target is smaller than the maximum velocity of the leader, and the noise is bounded by a radius* $r_n = min(\lambda/2, \lambda^*/2)$ *as given in theorem 2, then system (3.55) is stable at the equilibrium state* $(\tilde{v}_l - \tilde{v}) \approx 0$ *and* $(\tilde{p}_l - \tilde{p}_t) \approx r^\tau(\tilde{p}_l - \tilde{p}_t) / \|\tilde{p}_l - \tilde{p}_t\|$.

Proof of proposition 3.2

Similar to the proof of theorem 2, to analyze the stability of the system (3.55) we choose the Lyapunov function as:

$$V_\xi = \tilde{V}_l^t + \frac{1}{2} \tilde{x}_2^T \tilde{x}_2 \text{ ,} \tag{3.56}$$

Where $\tilde{V}_i^t = V_l^t(\tilde{d}_l^t)$ with $\tilde{d}_l^t = \|\tilde{p}_l - \tilde{p}_t\|$. Let $\tilde{x}_1^* = (p_l - p_t + z_{plt}) = \tilde{x}_1 / \lambda_l^t$ be the position error between the leader and the target, and note that the relation as follows: $\left(\partial \tilde{V}_i^t / \partial p_l\right)^T =$

$=\left(\partial\tilde{V}_i^t/\partial(p_l-p_t+z_{plt})\right)^T$. Taking the time derivative of equation (3.56) along the trajectory of the system (3.55) we obtain:

$$\begin{aligned}\dot{V}_\tau(t) &= \left(\nabla\tilde{V}_i^t\right)^T\dot{\tilde{x}}_1^* + \tilde{x}_2^T\dot{\tilde{x}}_2 = \frac{1}{\lambda_l^t}\left(\nabla\tilde{V}_i^t\right)^T\dot{\tilde{x}}_1 + \tilde{x}_2^T\dot{\tilde{x}}_2 \\ &= \tilde{x}_2^T(\nabla\tilde{V}_i^t + \dot{\tilde{x}}_2) = -\tilde{k}_{lv}^t\tilde{x}_2^T\tilde{x}_2 \le 0.\end{aligned} \tag{3.57}$$

So, this equation guarantees that system (3.55) is stable at the equilibrium position when using the control law (3.53).

3.4.5 Simulation Results

In this sub-section, we present the simulation results of the above proposed control algorithms. For these simulations, we assume that the initial velocities of the robots and the target are zero. The initial positions of the robots are random. Each robot is able to sense the position of other robots as well as the position of the target and obstacles. The target moves on a sine wave trajectory as $p_t=(0.9t+640,\ 160\sin(0.01t)+250)^T$. The general parameters of the simulations are listed in table 3.1.

TABLE 3.1: PARAMETER VALUES

Parameter	**Definition**	**Value**
N	Number of robots	9
φ_d	Desired formation angle	2π/3 (rad)
r^τ	Desired radius of circular formation	60 (m)
r^t	Target approach radius	100 (m)
r_a	Radius around each active node	25 (m)
r_r	Collision radius around each robot	45 (m)
ε_1, ε_2, ε_3	Positive constants	1, 0.5, 0.7
k_{l1}^t, k_{l2}^t	Factors for approaching to target	9, 0.6
k_{i1}^k, k_{i2}^k	Positive factors for fast repulsion	80, 12
k_{i1}^o, k_{i2}^o	Constants for fast obstacle avoidance	90, 15
k_{i1}^j, k_{i2}^j, k_{i3}^j	Positive constants	3, 4, 9
k_{iv}^j, k_{iv}^t, k_{iv}^k	Damping factors	1.4

A. Test the intelligence of the swarm

For this simulation, the formation angle is selected. Firstly, we test the proposed algorithms to generate the desired formations (V-shape and circular), and the proposed algorithm to control the robots towards the virtual nodes in the desired formation (Algorithm 3.1). Moreover, the stability of a swarm following the desired formations under the influence of the environment, in which there are stationary obstacles, is also tested.

The results of the simulations in figure 3.12 show that the desired formations are easily created. Robots, which have the random initial positions, have quickly achieved the desired positions in these desired formations while tracking a moving target without collisions. The position permutations among the members in the formation occurred, but they did not influence on the structure of the formation during the target tracking. At initial time, one robot is chosen as the leader, and then it is saved in order to drive its formation towards the target in a V-shape formation. At time *t=70s*, the V-shape formation was made, and it was kept until the square robot detected the obstacle O_1. At time *t=160s*, while avoiding the obstacle O_1, the virtual node, which was owned by the square robot, became a free node. Then, this virtual node attracted the triangular robot to become the active node at time *t=200s*. After escaping the obstacle, the square robot quickly approached to the remaining free node of the desired formation as shown in Fig.11. Similarly, the rhombus robot was permuted with another robot in the formation while avoiding the obstacle O_2. At time *t=250s*, the V-shape formation changed to the circular formation in order to encircle the target. In this situation, the member robots became the free robots, and then they approached the desired circular formation to become the active robots in this formation. Figure 3.13 shows that this circular formation was kept around the target at the desired radius r^τ at time *t=320s*. In other words, the leader's position is stable at the equilibrium point, at which $||p_l - p_t|| = r^\tau$.

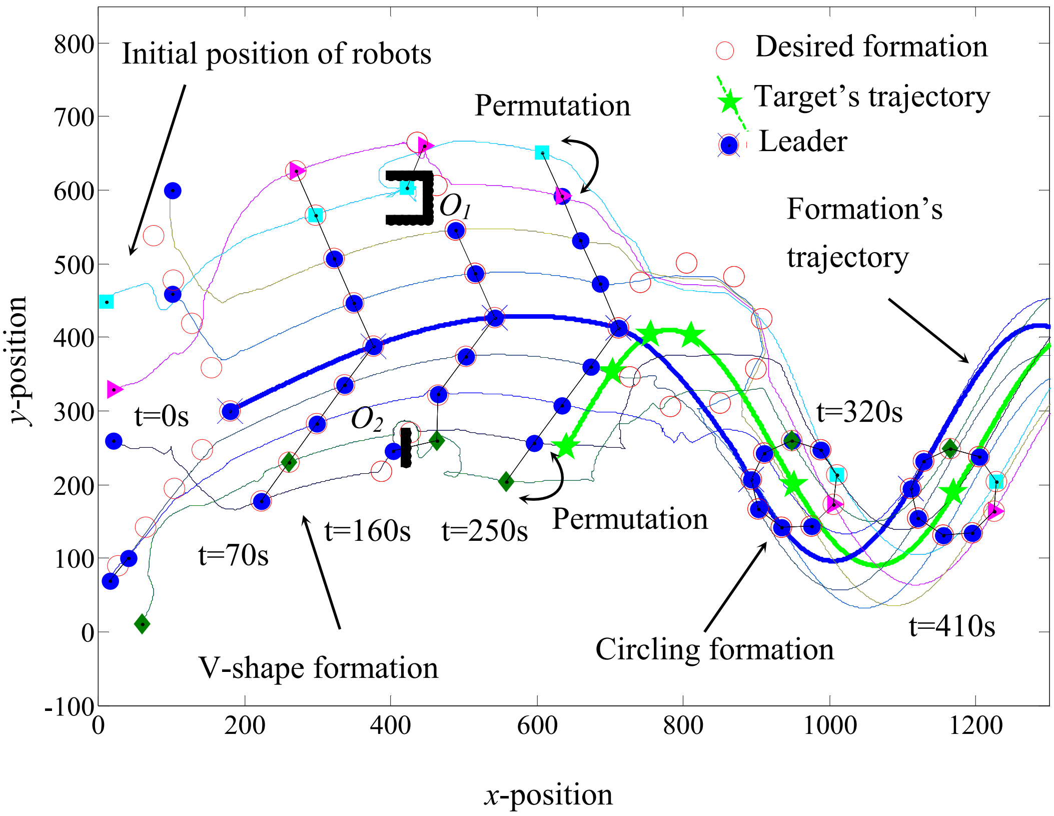

Figure 3.12: Path planning for a swarm following the desired formations under the influence of the obstacles while tracking a moving target.

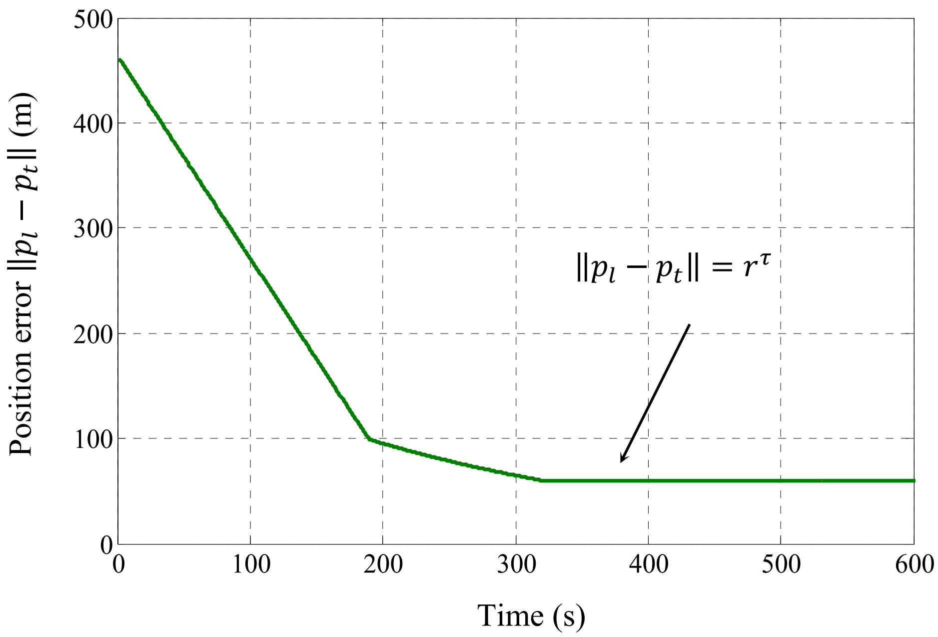

Figure 3.13: Position error $\|p_l - q_t\|$ in case the leader is not hindered while tracking a moving target.

Secondly, we also test the intelligence of a swarm when the leader is trapped in the complex obstacle (for example U-shape obstacle, see figure 3.14). In this situation, the actual leader has to transfer its leadership to another member in the swarm, and then it has to escape this obstacle. The simulation results in figure 3.14 show that, at time *t=0s*, the square robot is chosen as the leader, and its leadership was kept until it was trapped in the U-shape obstacle at time *t=200s*. While avoiding the obstacle, the square leader transferred its leadership to the triangular robot, which was not hindered and closest to the target. Then, this square leader became a free robot. It automatically found a way (lilac way) to escape this U-shape obstacle in order to continue following its formation. After receiving the leadership, the triangular robot reorganized a new formation, and continued to lead this formation in the target tracking. The distance between the new leader and the target was always shrunk until it achieved the active radius of the circular desired formation $\|p_l - p_t\| = r^\tau$. Then, this distance is maintained to encircle the moving target, see figure 3.15. Moreover, figure 3.14 shows that the position permutation between the square leader and the triangular leader does not influence on the desired structure of the formation.

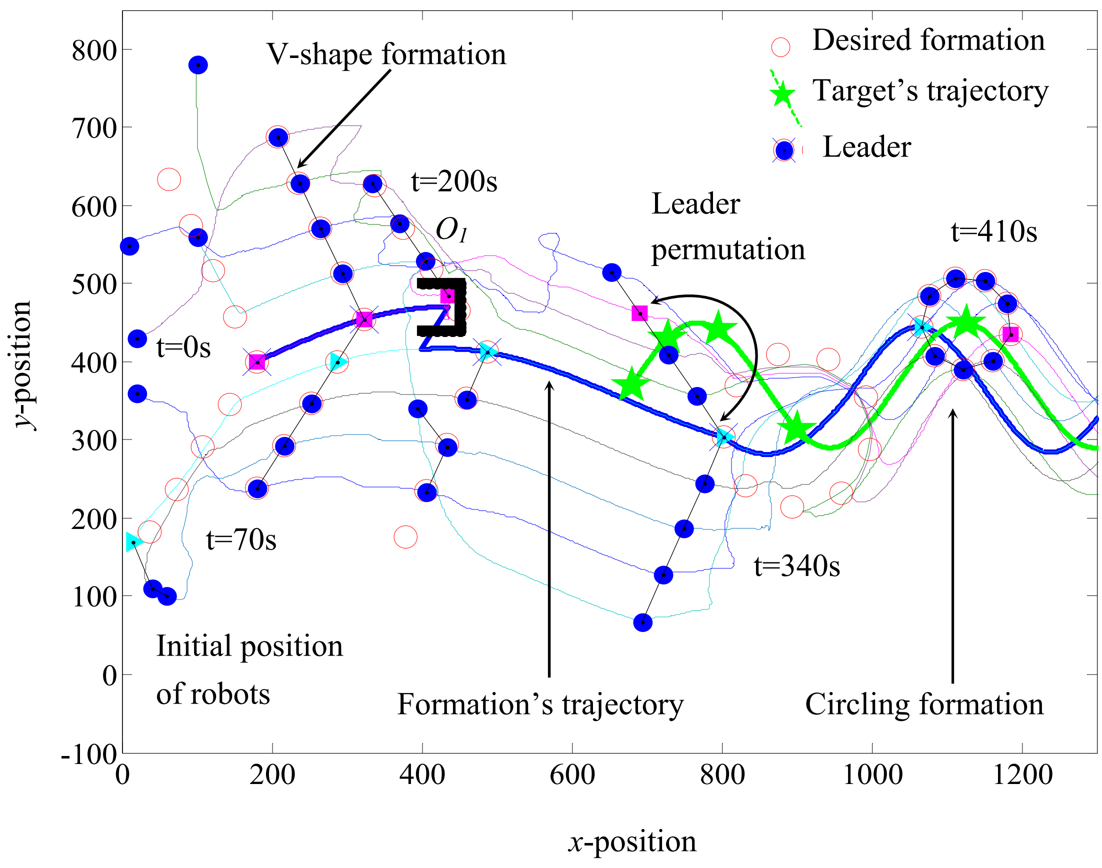

Figure 3.14: Path planning for a swarm following the desired formations while tracking a moving target with the leader permutation.

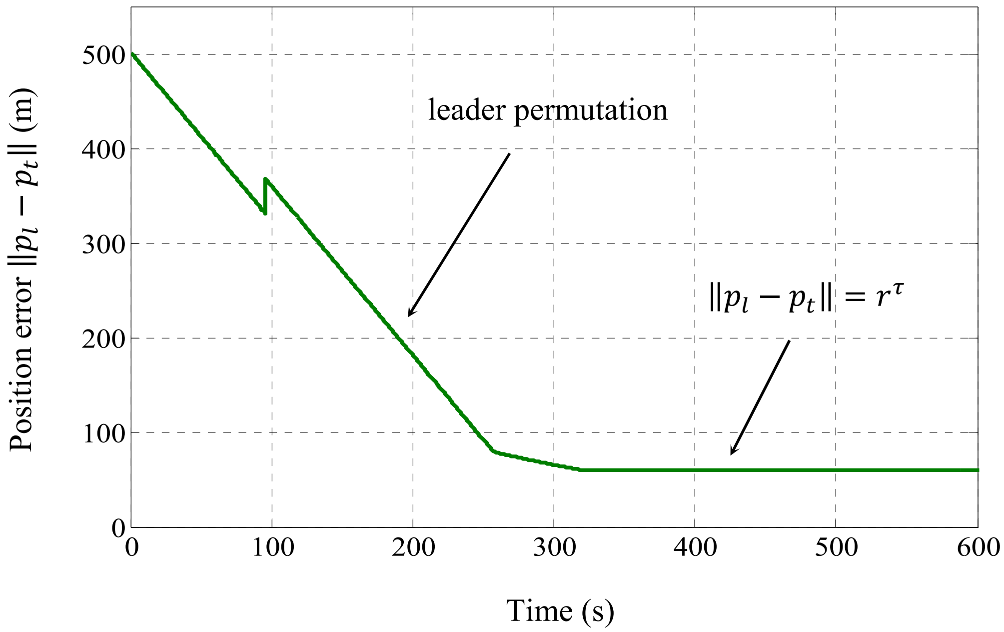

Figure 3.15: Position error $\|p_l - q_t\|$ while tracking a moving target in case the leader is permuted.

B. Test the stability of the formation under the influence of the noise

In this sub-section, we test the stability of the formation under the influences of noise and the change of the formation angle. The noises used in this simulation are Gaussian function with zero mean, variance of 1 and standard deviation of 1, see figure 3.16.

The formation angle is used for simulations as depicted in figure 3.17. The results of the simulations show that the robot i is always close to the active node j in the desired formation, and its formation was maintained following the desired formations (V-shape and circular formation) although there are the effects of the noisy environment and the changes of the formation angle φ(t)/2. The position error between each robot i and the active node j, at which this robot i was occupying, is small, see figure 3.18. The simulation results in figure 3.19 also show that from the random initial positions, the free robots have quickly found their desired position on the desired V-formation. Then, they tracked a moving target in a stable V-formation. At time *t=70s*, under the influence of the sudden change of the formation angle from 2π/3 to φ=(*π-0.6*) the stability of the formation was broken, and then the stability of this formation was quickly redesigned to continue tracking the moving target. In contrast, when the formation angle φ(t) changed slowly the formation of robots was always maintained following the desired V-formation with small position errors, see figure 3.18 and figure 3.19. Moreover, the simulation results also show that the noise had influences on the position error between the actual formation of robots and the desired formation, but this influence only caused the small changes in the formation of robots, see figure 3.18 and figure 3.19.

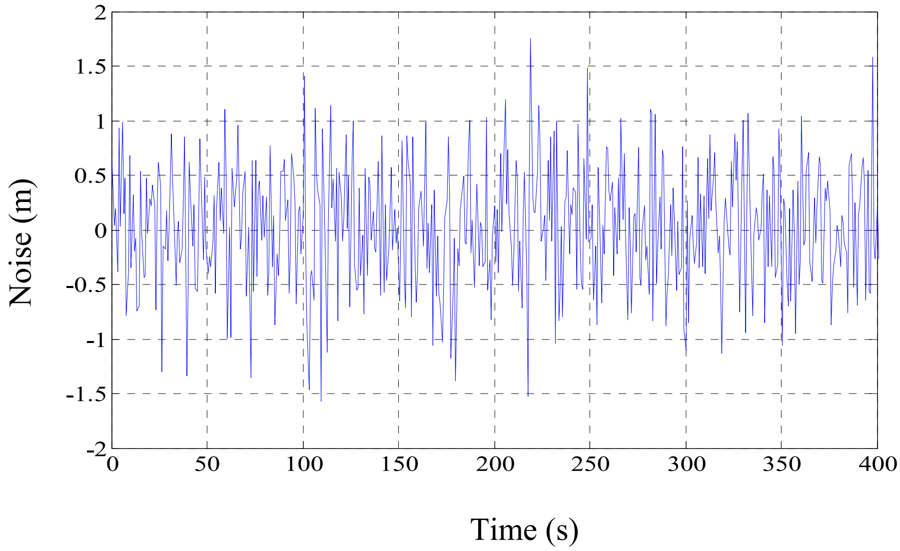

Figure 3.16 Noise effects on the system.

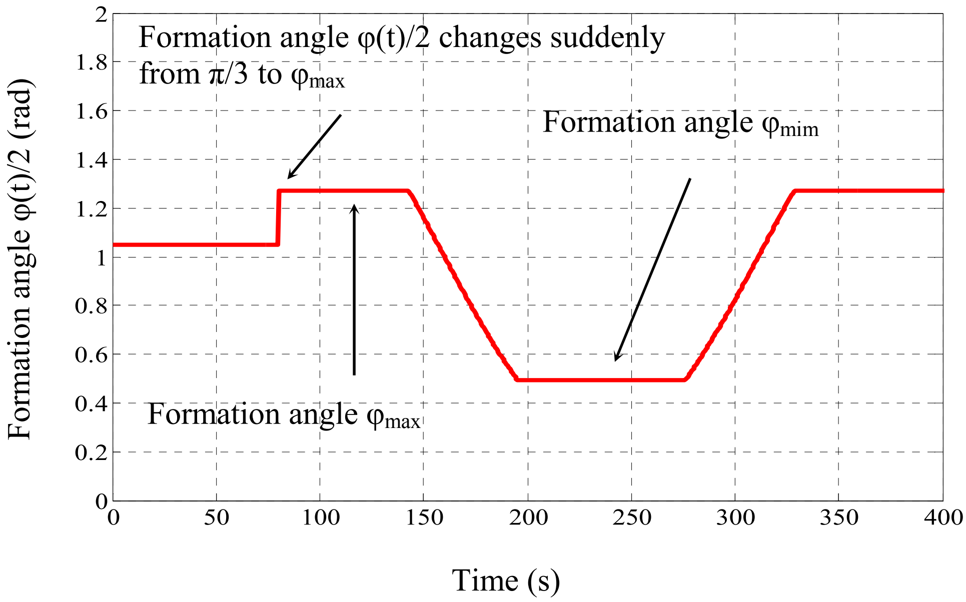

Figure 3.17: Formation angle $\varphi(t)/2$ while tracking a moving target.

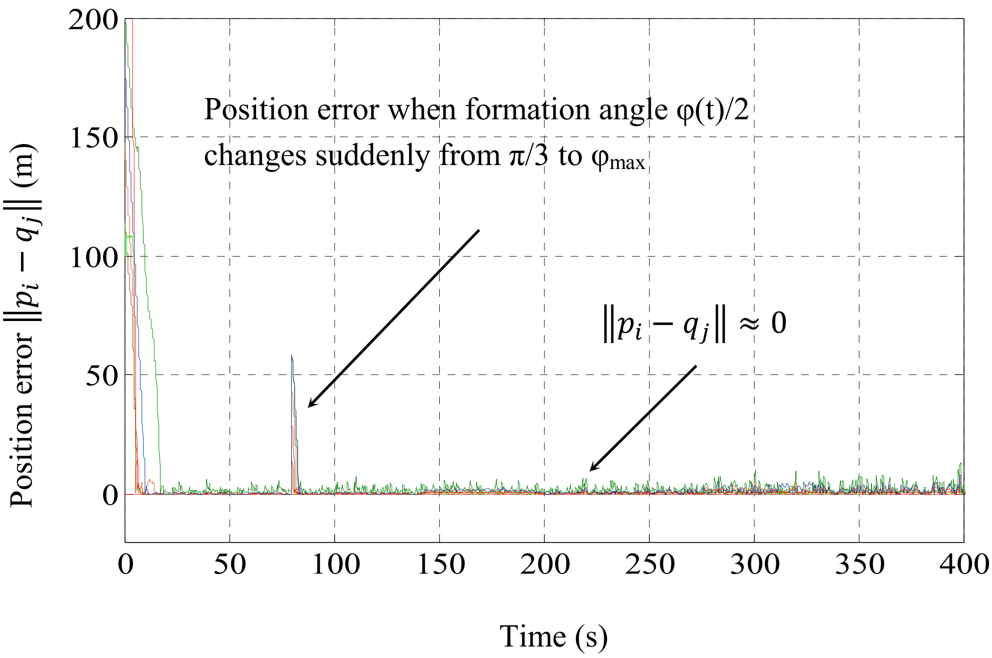

Figure 3.18: Position errors $\|p_i - q_j\|$ when there is an effect of the noise.

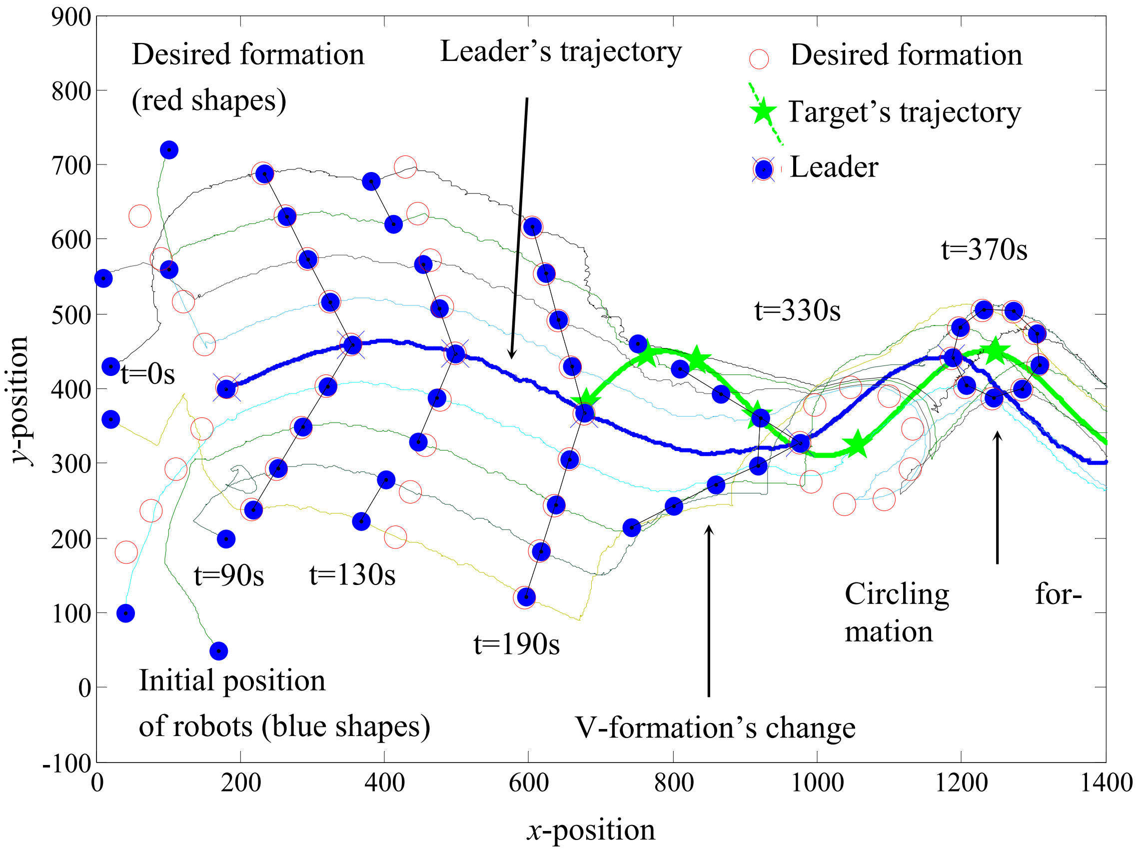

Figure 3.19: The influences of noises and the different formation angles on the swarm's trajectory while tracking a moving target.

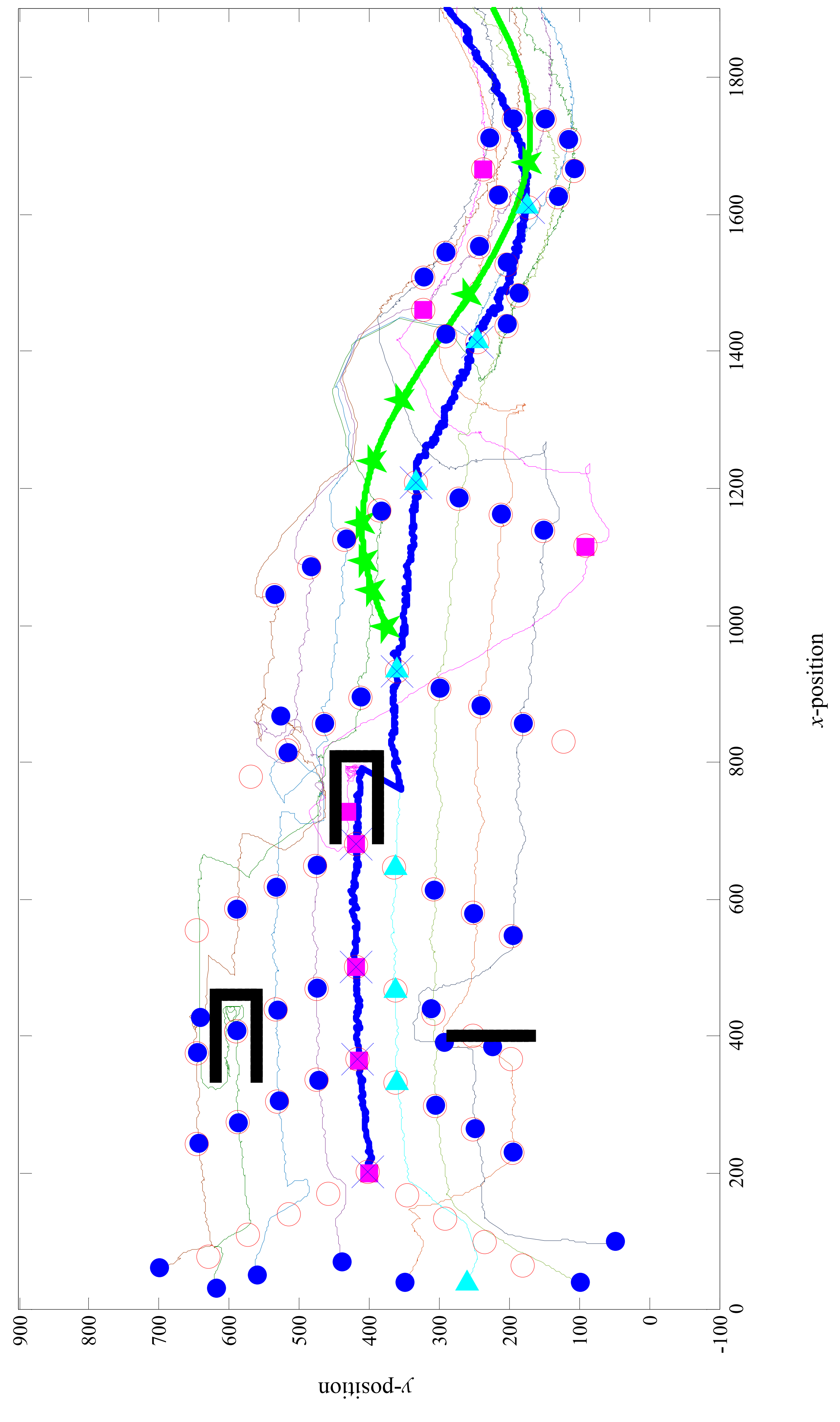

Figure 3.20: Path planning for a swarm following the desired formations while tracking a moving target under the influence of the noise

The simulation results in figure 3.20 also show that the robots can easily escape obstacles, and always converge to the designed virtual nodes in the desired formations although there are effects of the noise and the obstacles.

3.4.6 Conclusion

This section has proposed a novel approach to formation control of autonomous robots following the desired formations to track a moving target in a dynamic environment. The robot team is able to form V-shape formation to track the target efficiently and then changes its formation to the circular shape to better monitor the target. The stability analysis of the proposed formation control is given. The rotational force field combining with the repulsive force can drive the robot to quickly escape from the obstacle, more importantly is to avoid the local minimum problems when the sum of the attractive and repulsive forces of the potential field is equal to zero in the case of concave obstacle shapes.

The results of the simulations have proved the intelligence of the swarm while tracking a moving target. This swarm's intelligence expresses when the free robots automatically find the desired positions at the virtual nodes in the desired formation and occupy these positions while tracking a moving target. Furthermore, when robots are trapped in obstacles they can quickly find out the way to escape these obstacles, and continue to track the moving target with their swarm. In addition, the leader's intelligence is also expressed when it transfers its leadership to other free member in the swarm in order to avoid obstacles efficiently. The simulation results also show that formation's stability is maintained during movement, although there are influences of the noises and obstacles from the environment as well as the changes of the formation angle.

3.5 Direction control for collinear formation

This section considers an approach for the collinear formation control of autonomous robots while reaching the target position in a dynamic environment. This approach is built and developed based on the formation shape control method combined with the artificial vector field method. In this section, we focus on the control algorithm u_i^t for the target tracking. This control law has to guarantee that the movement direction of the formation towards the target position is kept stable in the given invariant orientation. The other controllers (the formation connection controller u_i^j, collision avoidance controller u_i^k, obstacle avoidance controller u_i^o) are built similarly to the section 3.4.

3.5.1 Target tracking control algorithm

Firstly, one robot, which has the closest distance to the target $d_l^t = \|p_l - p_t\| = \min\{\|p_i - p_t\|, \mathrm{i} = 1,...N\}$, is selected as the leader in order to control the motion of the formation. The target reaching controller, which is designed based on the relative position between the leader and the target, has to guarantee that the formation's motion is maintained in the stable direction to the target. This control law is designed as follows:

$$u_l^t = F_l^t(p_l)n_l^t - k_{l2}^t(p_l - p_\beta) - k_{lv}^t(v_l - v_t) + \dot{v}_t \tag{3.58}$$

This equation shows that the first component $F_l^t(p_l)n_l^t$ is used to control the target tracking with the value of the attractive force $F_l^t(p_l)$ and the unit vector $n_l^t = (p_l - p_t)/\|p_l - p_t\|$. The attractive force $F_l^t(p_l)$ is computed as follows:

$$F_l^t(p_l) = \begin{cases} -\dfrac{k_{l1}^t}{r^\tau}\|p_l - p_t\|, & if \;\; d_l^t < r^\tau \\ -k_{l1}^t, & otherwise. \end{cases} \tag{3.59}$$

Here, k_{l1}^t and r^τ are the positive factor and the radius to reach the target, respectively. The second component $-k_{lv}^t(v_l - v_t)$ is added as a damping term. The remaining component $-k_{l2}^t(p_l - p_\beta)$ works as the orientation controller in order to maintain the formation's motion in the stable direction towards the target. In other words, this controller guarantees that $\lim_{t\to\infty}(p_l(t) - p_\beta(t)) = 0$ or $\lim_{t\to\infty}(\alpha(t)) = 0$. Here, k_{l2}^t is a positive constant, and $(p_l - p_\beta)$ is the relative position vector between the leader and the desired leader position $p_\beta = (x_\beta, y_\beta)^T$. This desired leader position is dependent on the rotation direction of the desired orientation angle β_d, see figure 3.21.

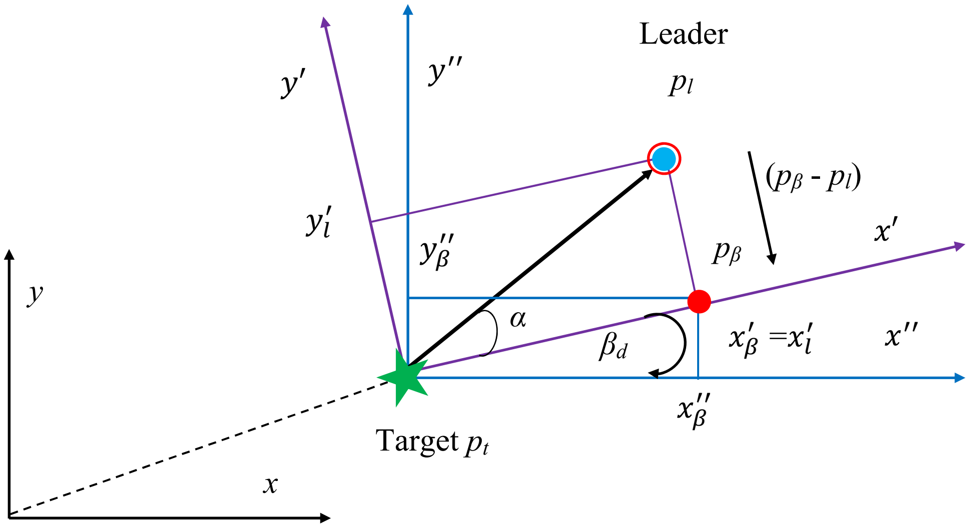

Figure 3.21: The desired leader position $p_\beta = (x_\beta, y_\beta)^T$ in case the positive desired angle.

Assume that the desired orientation angle rotates clockwise is positive, and rotates counter-clockwise is negative. Consider the case that the desired orientation angle is positive, the desired leader position is calculated as follows: We build the coordinate tem $x'y'$ based on this positive desired orientation angle β_d as depicted in figure 3.21. The desired coordinates on the coordinate system $x'y'$, at which the leader has to reach in order to guarantee that the movement direction towards the target is stable, is determined as $(x_\beta',\ y_\beta')^T = (x_l',\ 0)^T$. On the coordinate system xy, the coordinates of this desired leader position are determined as follows:

$$\begin{pmatrix} x_\beta \\ y_\beta \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \end{pmatrix} + \begin{pmatrix} \cos\beta_d & -\sin\beta_d \\ \sin\beta_d & \cos\beta_d \end{pmatrix} \begin{pmatrix} x_l' \\ 0 \end{pmatrix}. \tag{3.60}$$

On the other hand, from figure 3.21 the coordinates of the leader is determined as follows:

$$\begin{pmatrix} x_l \\ y_l \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \end{pmatrix} + \begin{pmatrix} \cos\beta_d & -\sin\beta_d \\ \sin\beta_d & \cos\beta_d \end{pmatrix} \begin{pmatrix} x_l' \\ y_l' \end{pmatrix}. \tag{3.61}$$

By equating (3.61), we obtain x_l' as follows:

$$x_l' = (x_l - x_t)\cos\beta_d + (y_l - y_t)\sin\beta_d\,. \tag{3.62}$$

In case the desired angle is negative, we will find x_l' as follows:

$$x_l' = (x_l - x_t)\cos\beta_d - (y_l - y_t)\sin\beta_d\,. \tag{3.63}$$

3.5.2 Simulation results

In this section, we give the simulation results in order to verify the function of the above presented control algorithms. The target is assumed stationary. The general parameters for simulations are listed in the table 3.1.

Firstly, we test the algorithms to generate the collinear desired formation, and the algorithm to drive the robots towards the collinear desired formation. The results of the simulations in figure 3.22 show that the collinear desired formation can easily be created with the different formation angles δ_d. Robots, which have the random initial positions, have achieved the desired positions in this desired formation while reaching the target.

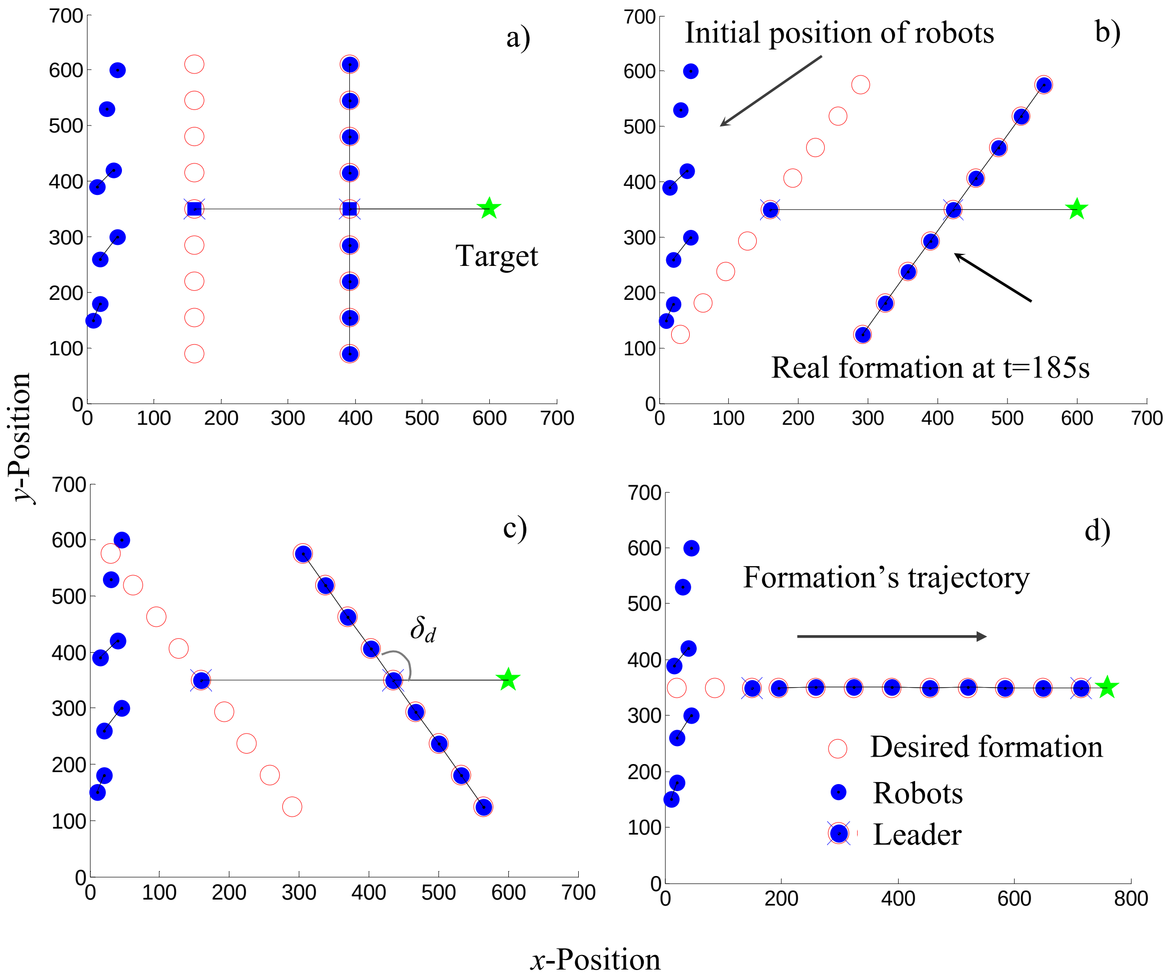

Figure 3.22: Simulations on the angle δ_d change of the collinear desired formation and the desired formation follower of robots in order to reach the target. Plots a), b), c), d) depict the formation's motion with angle $\delta_d=\pi/2$, $\delta_d=\pi/3$, $\delta_d=2\pi/3$, $\delta_d=0$, respectively.

Secondly, the stability of the swarm following the desired collinear formation in an invariable direction towards the target under the influence of the environment is tested. For this simulation, the target's position, the formation angle, and the desired orientation angle are chosen as $p_t = (700, 350)^T$, $\delta_d = \pi/2$, and $\beta_d = 0$, respectively. Obstacles o_1, o_2 and the initial position of the robots are depicted in figure 3.23, and figure 3.24.

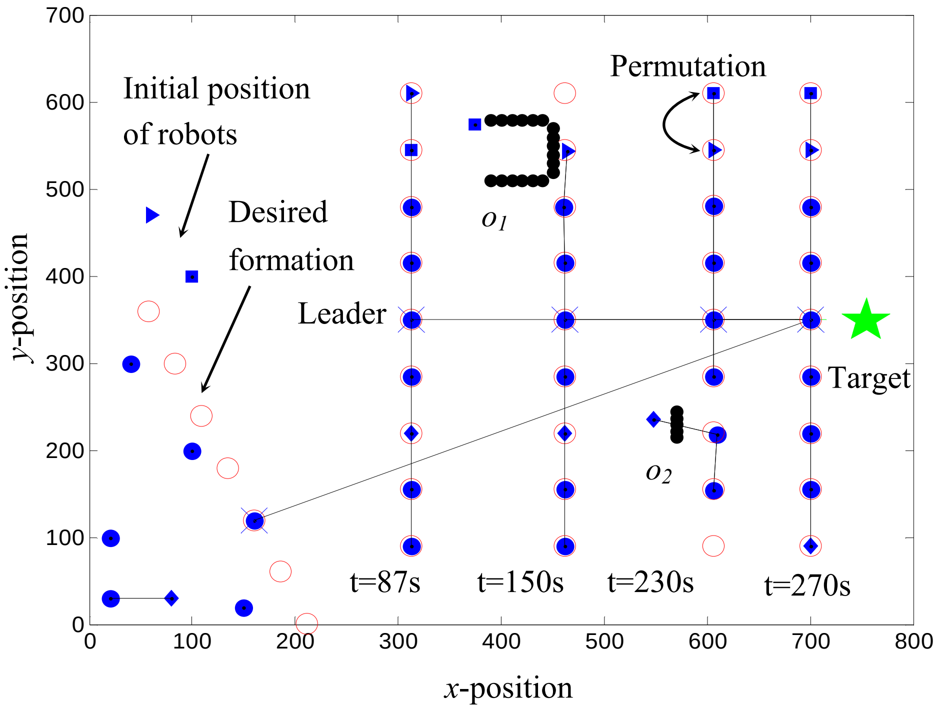

Figure 3.23: The influence of the environment on the collinear formation of robots during reaching towards the target. The red circles, blue shapes, and black shapes are the desired formation, robots, and the obstacles, respectively.

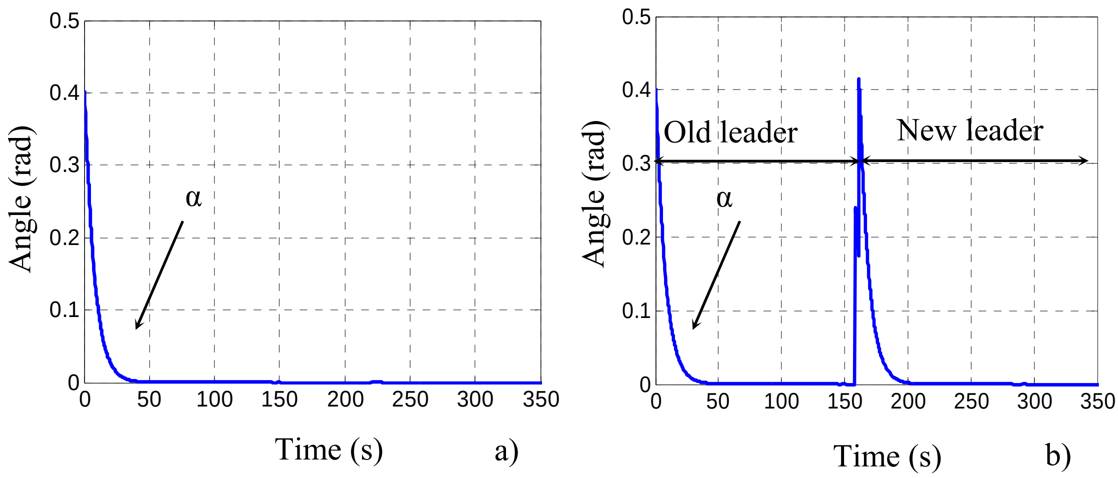

Figure 3.24: Simulation on the stability of the motion direction $\alpha = \angle(p_l - p_t),(p_\beta - p_t)$ to the target of the collinear formation when the environment changes, plot a) and the leader is permuted, plot b).

Case 1. The position permutation between member robots while avoiding obstacles.

The simulation results in figure 3.23 and figure 3.24 show that the collinear formation of robots, which follows a desired formation, is maintained during flight towards the target without collisions. At time t=0s, one robot is chosen as the leader of the swarm, and then it is saved in order to drive its formation to the target in the stable motion direction α, see figure 3.24a. At time t=87s, the formation of a swarm is made based on the desired structure, and it is kept until the squares robot detects the obstacle o_1. While avoiding obstacle o_1, the virtual node, which the square robot has owned, became a free node, and it attracted the triangle robot to become the active node. After escaping the obstacle, the square robot quickly reached the remaining free node of the desired formation, see figure 3.23. Similarly, at time t=230s, the rhombus robot is permuted with other robots in the swarm. In this simulation, the obstacles of the environment can permute the position of the member robots in the formation, but they do not influence on the formation angle δ_d and the motion direction α of the swarm.

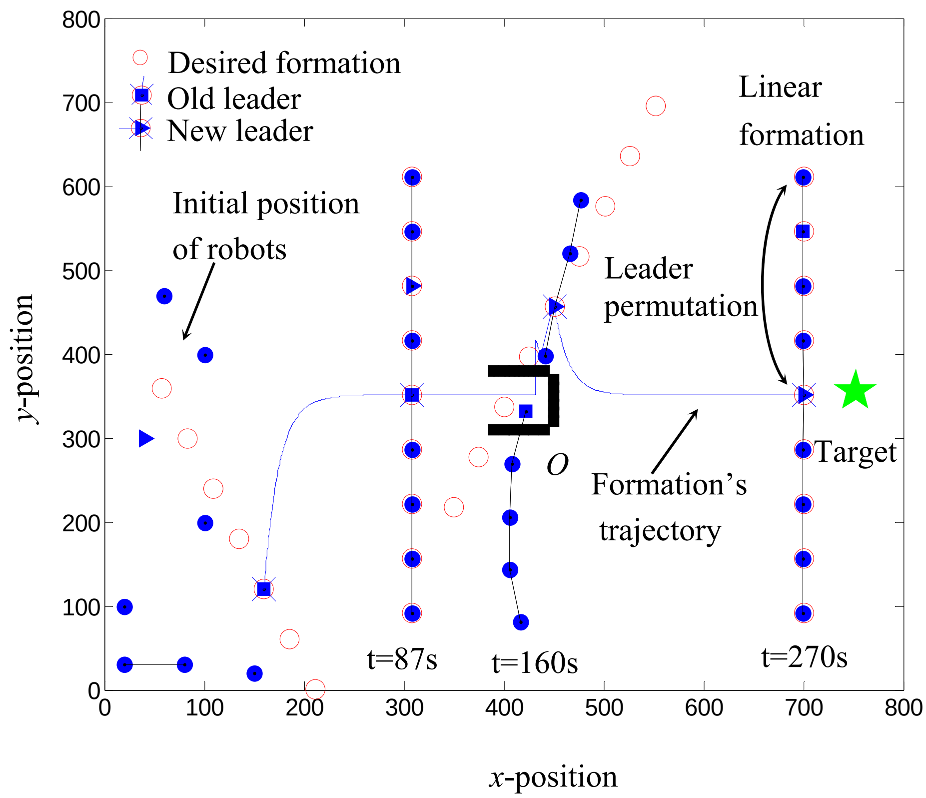

Figure 3.25: Simulation on the leader permutation of a collinear formation while moving towards the target in a stationary environment. The red circles, blue shapes, and black shapes are the desired formation, robots, and the obstacles, respectively.

Case 2. The leader permutation while avoiding obstacles.

The leader change influences on the organization of the formation during movement towards the target. Figure 3.25 shows that, at time t=0, the square robot is chosen as the leader and it also saved to lead the swarm to the target. Using the orientation controller $-k_{l2}^{t}(p_l - p_\beta)$ as given in (3.58), this leader quickly achieved the desired direction β_d =0 to the target (t=50s to t=150s), see figure 3.24b. The formation's organization is changed when the square leader (old leader) is trapped in the U-shape obstacle at time t=160s. In this situation, the square leader lost the leader role. It became a free robot and automatically found a way to escape this obstacle and continued to track its formation. The triangle robot, which is a free robot and closest to the target, is used as a new leader in order to continue to lead the swarm towards the target. Figure 3.24b shows the new leader has quickly led its formation in the desired direction before reaching the target.

3.5.3 Conclusion

In this section, we have proposed a novel approach to formation control of autonomous robots following a desired collinear formation to reach a stationary target. The desired formation with the different formation angles is built on the relative position between the target and the leader of the swarm. The trajectory of member robots is driven by the artificial force fields from the virtual attractive nodes of the desired formation. The mission of the leader is to lead the formation towards the target in a desired direction. Furthermore, the repulsive force fields between robots are used to guarantee that there are no collisions in the swarm during movement. Moreover, in order to avoid obstacles of the environment, the repulsive and rotating force fields are also added. The results of the simulations have shown that using the proposed control algorithms the member robots have quickly achieved the desired positions in the desired formation. In some cases, such as obstacle avoidance, the position of some robots in formation can be permuted, but the structure and the motion direction of the formation are kept.

4 Cooperative Control for Multi Robot Systems

4.1 Introduction

Formation control of multi-robot systems has been one of the interesting research topics in the control community all over the world in recent years. Its potential applications in many areas, such as search and rescue missions, forest fire detection and surveillance, is the motivation and reason for this attraction. In the formation control of multi-robot systems, the moving trajectory determination of each member-robot and the control of its motion along this determined trajectory are crucial problems. One of the effective and interesting methods to solve these problems is the artificial vector field method as presented in chapter 2. In this method, the motion of the robots is controlled by the artificial force fields that are built based on the relative positions of the robots, target and obstacles of the environment.

In recent years, the artificial potential field method has been widely studied and used to control the formation of multi-agent systems to reach the position of the goal in a dynamic environment, see [14]-[33]. One of the main issues in the formation control of multi-agents to track a moving target is that all the member robots have to move together without collisions among them in an ordered swarm. Moreover, the whole swarm must avoid obstacles along its trajectory, which has a big influence on the target reaching. In order to solve these problems, the motion of each member robot is controlled by a total force field which includes the interactive forces between neighboring robots, the repulsive forces from the environment obstacles, and the attractive force of the target. Under the effect of this total force, the formation of the swarm is stably maintained while the swarm reaches the target position in the free environment. In contrast, this stability is broken when the swarm avoids the obstacles of the dynamic environment. In this situation, the agents of the swarm are split, and each agent will itself determine its direction to avoid the obstacle. After the swarm has overcome the obstacles, its organization is redesigned; however, its formation is possibly changed, see [34]-[50]. This problem can be resolved, if the swarm maintains its formation in a smaller size. This is an interesting topic that attracts the attention from researchers in recent years. Some research results around this topic are presented in [74]-[76]. In this approach, each agent can cooperatively learn the network's parameters to decide the size and the split of the network in a decentralized fashion so that the connectivity, formation and tracking performance can be improved when avoiding obstacles.

In this chapter, we present a control method for the cooperation among the members of a swarm under the effect of the dynamic environment. The stability and robustness of the formation of a swarm are maintained while avoiding obstacles and tracking a moving target. Moreover, in a complex environment where the space among obstacles is narrow, we suggest that the swarm's size can change, so the swarm can easily pass through this space. In addition, in order to avoid collisions and maintain the stability in the formation of a swarm, the neighboring robots will be connected to each other by the attractive and repulsive vector field between them. Information about obstacles in the environment will be sent to all other member robots in the swarm. Therefore, the velocity of the robots in a formation while avoiding obstacle is the same.

The main contributions of this chapter are as follows:

- "Connection between neighboring robots" is presented in section 4.3. In this content, we propose an approach for the connection between neighboring robots in a swarm. The main aim of this approach is to generate the stable and robust links among the neighboring robots in the formation of a swarm. These connections are controlled by the attractive/repulsive forces among them. Hence, using the attractive/repulsive forces, the neighboring robots will quickly approach the equilibrium position, at which the distances among them are constant (that is the sum of the attractive/repulsive forces are equal to zero).
- "Adaptive formation control in a dynamic environment" is presented in section 4.4. In this content, we consider an approach for the adaptive formation control of the autonomous robots while tracking a moving target in a dynamic environment. The main aim of this approach is to control the formation of a swarm that can easily and quickly escape the obstacles of the environment without the collisions (especially, in an environment, in which the space between the obstacles is narrow). While avoiding the obstacles, the formation structure of the swarm can change, but it is not broken. In order to perform this idea, an applied active method is to shrink the swarm's size into the smaller size. Hence, the adaptive formation control algorithm is designed such that the swarm's size is inversely proportional to the sum of the repulsive forces from the obstacles acting on the swarm.
- "Cooperative formation control in a dynamic environment" is presented in section 4.5. In this content, we consider an approach for the cooperative formation control of the autonomous robots while tracking a moving target in a dynamic environment. The main aim of this approach is to control the formation of a swarm to quickly avoid and

overcome the obstacles of the environment following the direction of the target's trajectory while keeping the formation. This approach is built based on the repulsive force combined with the rotational force surrounding the obstacles.

The rest of this chapter is organized as follows: The problem statement is given in the next section. Section 4.3 presents the cooperative control algorithm for two neighboring robots. In section 4.4, the adaptive control method for the formation of a swarm while avoiding obstacles to track a moving target is presented. Finally, section 4.5 presents the cooperative control algorithm for the neighboring robots in the formation of a swarm while tracking a moving target in a dynamic environment.

4.2 Problem statement

In this section, we consider a swarm of *N* robots ($N \geq 2$) that moves in a two-dimensional Euclidean space $\{R^2\}$ with *M* obstacles of the environment. Each robot's motion, which is assumed as a moving point in the space, is described by the dynamic model as follows:

$$\begin{aligned} \dot{p}_i &= v_i \\ \dot{v}_i &= u_i, \qquad i = 1,..,N. \end{aligned} \tag{4.1}$$

Where $(p_i,\ v_i,\ u_i) \in \{R^2\}$ are the position, the velocity, and the control input of the robot *i*, respectively.

In the formation of a desired swarm, the neighboring robots have to link with each other to generate the constant distances among them (example in figure 4.1). Let $N_i^\alpha(t)$ be the set of the robots in the neighborhood of the robot *i* at time (*t*), such that:

$$N_i^\alpha(t) = \left\{ \forall j : d_i^j \leq r^\alpha, j \in \{1,..N\}, j \neq i \right\}. \tag{4.2}$$

Where $r_0^\alpha > 0$, $r^\alpha > 0$, and $d_i^j = \left\| p_i - p_j \right\|$ are the collision range surrounding each robot, the interaction range (radius of neighborhood circle, see figure 4.1), and the Euclidean distance between the robot *i* and the robot *j*, respectively. For example, in figure 4.1, the robot R_1 has three neighbors: R_2, R_3, R_4.

Similarly, let $N_i^\beta(t)$ be the set of the obstacles in the neighborhood of the robot *i* at time (*t*), such that:

$$N_i^\beta(t) = \left\{ \forall o : d_i^o \leq r^\beta, o \in \{1,..M\}, o \neq j \right\}. \tag{4.3}$$

Here $r^{\beta} > 0$ and $d_i^o = \|p_i - p_o\|$ are the obstacle detection range and the Euclidean distance between the robot i and the obstacle o, respectively.

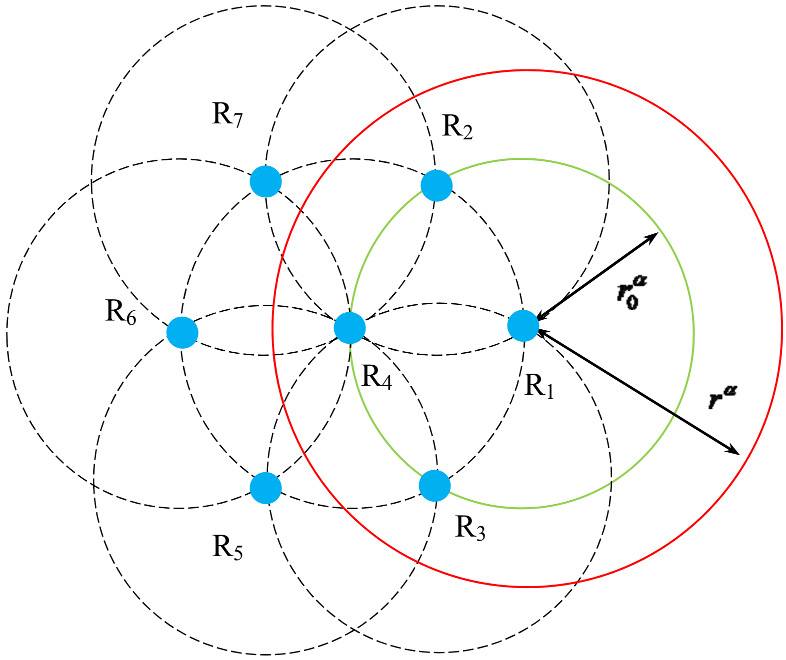

Figure 4.1: Configuration of a desired swarm of seven member-robots.

4.3 Connection between neighboring robots

Consider robot (p_i, v_i) and robot (p_j, v_j) that move with the dynamic model as described in (4.1). The control input u_i is proposed as

$$u_i = f_i^j(p_i) - k_{iv}^j c_i^j (v_i - v_j) + \dot{v}_j . \tag{4.4}$$

In this equation, the relative velocity $(v_i - v_j)$ between the robot i and its neighbor j is used as damping term with the damping scaling factor k_{iv}^j. The scalar c_i^j is used to determine if the robot j is a neighbor of robot i. It is defined as:

$$c_i^j = \begin{cases} 1 & \text{if } j \in N_i^{\alpha}(t) \\ 0 & \text{if } j \notin N_i^{\alpha}(t). \end{cases} \tag{4.5}$$

To create the attractive/repulsive force field $f_i^j(p_i)$ between the robot i and its neighbor j, a respective potential function is proposed as:

$$V_i^j(p_i) = \frac{c_i^j}{2}\left(k_{ip}^{1j}\left(\frac{1}{d_i^j} - \frac{1}{r_0^\alpha}\right)^2 + k_{ip}^{2j}\left(d_i^j - r_0^\alpha\right)^2\right). \tag{4.6}$$

Taking the negative gradient of this potential function at p_i (see in the Appendix), we obtain the attractive/repulsive force, which is depicted in figure 4.2, as follows

$$\begin{aligned} f_i^j(p_i) &= -\nabla V_i^j(p_i) \\ &= c_i^j\left(\left(\frac{1}{d_i^j} - \frac{1}{r_0^\alpha}\right)\frac{k_{ip}^{1j}}{(d_i^j)^2} - k_{ip}^{2j}\left(d_i^j - r_0^\alpha\right)\right)n_i^j. \end{aligned} \tag{4.7}$$

Where $n_i^j = \left(p_i - p_j\right)/\left\|p_i - p_j\right\|$ is a unit vector along the line connecting p_i *to* p_j, and d_i^j is the Euclidean distance as shown in equation (4.2). The positive constants ($k_{ip}^{1j}, k_{ip}^{2j},$) are used to regulate the fast collision avoidance and the stability in the set of the α neighbors of the robot i.

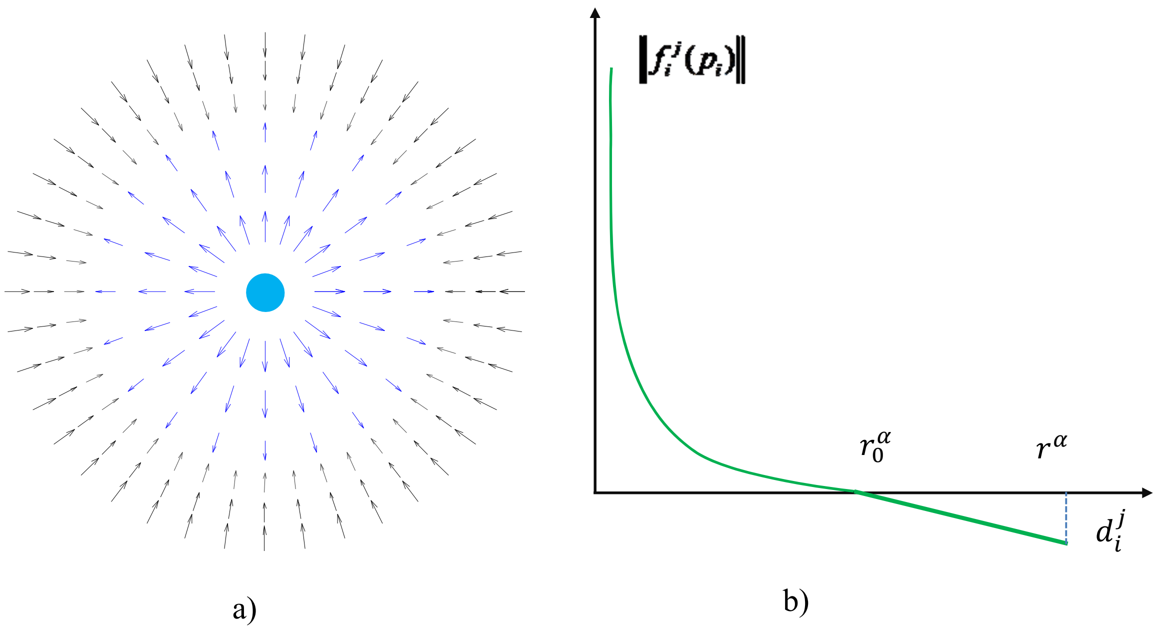

Figure 4.2: The description of the attractive/repulsive force field surrounding the robot j ($j \in N_i^\alpha(t)$) that acts on the robot i (a) and its amplitude (b).

The interacting ranges ($r_0^\alpha, r^\alpha > 0,$ shown in figure 4.1) describe the influence of the force $f_i^j(p_i)$ on the robot *i*. When $0 < d_i^j < r_0^\alpha$, then robots *i* and *j* repel each other to avoid the collision between them. Otherwise, when $r_0^\alpha < d_i^j \le r^\alpha$, they attract each other to achieve the equilibrium position $(d_i^j = r_0^\alpha)$ in the set of α neighbors of robot *i*. In case $d_i^j > r^\alpha$ there is no interaction between these members. As depicted in figure 4.2, the effect of the attractive/repulsive force $f_i^j(p_i)$ of the robot *j* ($j \in N_i^\alpha(t)$) on the robot *i* is depending on the relative position between robot *j* and robot *i*. Under the effect of this attractive/repulsive force, the neighboring robots will quickly approach the equilibrium position $(d_i^j = r_0^\alpha)$, at which the force $f_i^j(p_i)$ is equal to zero.

Theorem 4.1. *Consider the robot i is described by the model (4.1), and controlled by the control law (4.4). If the robot* $j \in N_i^\alpha(t)$, *then robot i will approach to the equilibrium position, at which* $v_i = v_j$ *and* $\|p_i - p_j\| = r_0^\alpha$.

Proof of theorem 4.1

Consider a point p_{ij} lies on the line connecting p_i to p_j, and satisfies $\|p_{ij} - p_j\| = r_0^\alpha$, see figure 4.3. The position of this point p_{ij} is calculated as follows:

$$(p_{ij} - p_j) = r_0^\alpha (p_i - p_j) / \|p_i - p_j\| \ . \tag{4.8}$$

This equation can be rewritten as

$$p_{ij} = \lambda_i^j p_i + (1 - \lambda_i^j) p_j \ . \tag{4.9}$$

Where $\lambda_i^j = r_0^\alpha / \|p_i - p_j\|$. Now, in order to analyze the stability of the robot *i* at the equilibrium position, at which $v_i = v_j$ and $\|p_i - p_j\| = r_0^\alpha$, we let $x_1 = p_i - p_{ij} = (1 - \lambda_i^j)(p_i - p_j)$ and $x_2 = v_i - v_j$. The error dynamic of the system is described as follows:

$$\begin{aligned} \dot{x}_1 &= (1 - \lambda_i^j) x_2 \\ \dot{x}_2 &= \dot{v}_i - \dot{v}_j. \end{aligned} \tag{4.10}$$

Substitute (4.4) into (4.10) we obtain:

$$\begin{aligned} \dot{x}_1 &= (1 - \lambda_i^j) x_2 \\ \dot{x}_2 &= -\nabla V_i^j(p_i) - k_{iv}^j x_2. \end{aligned} \tag{4.11}$$

Choose the Lyapunov function as follows:

$$V = V_i^j(p_i) + \frac{1}{2} x_2^T x_2. \tag{4.12}$$

Let $x^*_1 = (p_i - p_j) = x_1 / (1-\lambda_i^j)$ be the position error between the robot i and the robot j, and note that the relation in (4.6): $\left(\partial V_i^j(p_i) / \partial p_i\right)^T = \left(\partial V_i^j(p_i) / \partial (p_i - p_j)\right)^T$. Taking the time derivative of equation (4.12) along the trajectory of the system (4.11) we obtain:

$$\begin{aligned} \dot V(t) &= \left(\nabla V_i^j(p_i)\right)^T \dot x_1^* + x_2^T \dot x_2 \\ &= \frac{1}{(1-\lambda_i^j)} \left(\nabla V_i^j(p_i)\right)^T \dot x_1 + x_2^T \dot x_2 \\ &= x^T{}_2 \left(\nabla V_i^j(p_i) + \dot x_2\right) \\ &= -k_{iv}^j x_2^T x_2 \le 0. \end{aligned} \tag{4.13}$$

So, this equation guarantees that the system (4.11) is stable at the equilibrium position, at which $p_i = p_j$ and $v_i = v_j$ when the control law (4.4) is used. In other words, using the controller (4.4) the robot i will approach the equilibrium position, at which $\|p_i - p_j\| = r_0^\alpha$ and $v_i = v_j$.

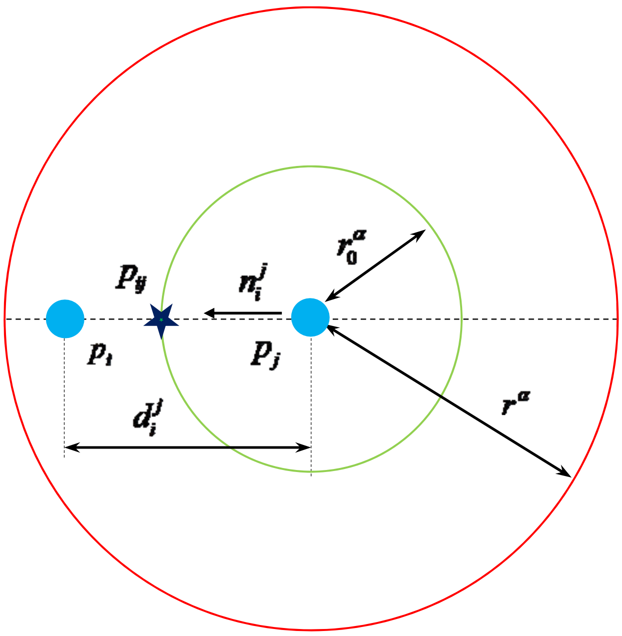

Figure 4.3: The description of the approach of the robot i towards the equilibrium position, at which $\|p_i - p_j\| = r_0^\alpha$ along the direction of the attractive force field from its neighbor j.

4.4 Adaptive formation control in a dynamic environment

4.4.1 Problem formulation

This chapter presents an approach for the adaptive formation control of multi-agent systems while tracking a moving target in a dynamic environment. In this approach, while the swarm reaches the target position, if it detects obstacles on the way, its size will change in order to quickly avoid these obstacles, but its formation is maintained. In special cases, such as when the space among the obstacles is narrow, then the swarm's size of the robots will automatically shrink into a smaller size. Hence, the swarm can easily pass through this space, while the swarm's connection is still kept, see figure 4.4. However, when the swarm reaches a minimum desired size, at which the swarm cannot overcome the obstacles, its formation will be broken. The robots will automatically split from their swarm in order to escape the obstacles as the free robots, and avoid the collisions with each other. After the swarm has overcome the obstacles, its formation is reorganized to continue tracking the target. In other words, these free robots will themselves find their swarm and they link to each other in a new formation. In our approach, the information, which is obtained from the changing environment, is concurrently sent to all robots in the swarm. Therefore, the swarm's size will quickly be adapted to the changes of the environment. In order to handle these problems, the adaptive formation control algorithm is built based on the change of the desired distance between the neighboring robots. This desired distance depends on the sum of the repulsive forces from obstacles acting on the swarm.

The idea for the formation adaptation of a swarm in a complex environment while tracking a moving target is depicted in figure 4.4. When some robots in the swarm detect obstacles then the swarm's size will automatically shrink into smaller size so that the swarm can easily pass through these obstacles, but the formation of the swarm is maintained (example figure 4.4c). However, in order to avoid collisions among the robots in a formation the swarm's size is only allowed to reduce to a minimum desired size. Then, the member robots in the formation will automatically split from its formation to become the free robots to avoid obstacles in the direction toward the target (example figure 4.4d). After the robots have overcome the obstacles, the swarm's structure will be redesigned in a new desired swarm. The changing of the swarm's size to adapt to a complex environment is controlled based on the desired radius r_0^α (see figure 4.1 and figure 4.2) between the neighboring robots. The control algorithm for this adaption is presented in section 4.2.2.

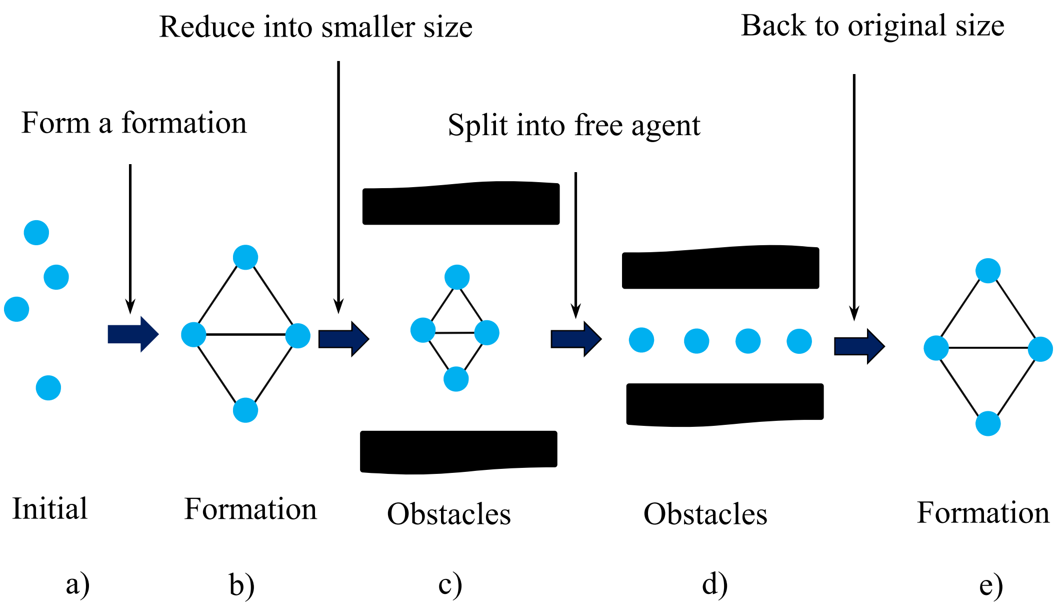

Figure 4.4: The description of the adaptive formation control of a swarm while avoiding obstacles in a narrow space: the original formation (b) and (e), the smaller reduced formation (c), and the formation splitting (d).

4.4.2 Adaptive formation control algorithm

This sub-section presents the adaptive control algorithm for the formation of a swarm of *N* robots, which pass through *M* obstacles of the environment to track a moving target. The control law for each robot *i* (*i=1,..,N*) is given as follows:

$$u_i = u_i^o + u_i^j + u_i^t. \tag{4.14}$$

A. Obstacle avoidance control

The first component u_i^o of (4.14) is used to control the obstacles avoidance for the robot *i* of the swarm while tracking a moving target. This component is proposed as:

$$u_i^o = \sum_{o=1}^{M} \left(f_i^o(p_i) - k_{iv}^o c_i^o (v_i - v_o) \right). \tag{4.15}$$

Where the relative velocity vector $(v_i - v_o)$ between the robot *i* and its neighboring obstacle o ($o \in N_i^\beta(t)$) is used as a damping term with the damping scaling factor k_{iv}^o. The scalar c_i^o, which is used to determine that an obstacle o is a neighboring obstacle of robot i or not, is defined as follows:

$$c_i^o = \begin{cases} 1 & \text{if } o \in N_i^\beta(t) \\ 0 & \text{if } o \notin N_i^\beta(\text{t}). \end{cases} \tag{4.16}$$

The repulsive force $f_i^o(p_i)$ is created surrounding the obstacles to drive the robot i away from these obstacles. It is designed as:

$$\begin{aligned} f_i^o(p_i) &= -\nabla V_i^o(p_i) \\ &= c_i^o\left(\left(\frac{1}{d_i^o} - \frac{1}{r^\beta}\right)\frac{k_{ip}^o}{(d_i^o)^2} - k_{ip}^\delta\left(d_i^o - r^\beta\right)\right)\text{n}_i^o \end{aligned}. \tag{4.17}$$

In this equation, the positive constants k_{ip}^o, k_{ip}^δ are applied to control the fast obstacle avoidance. $n_i^o = (p_i - p_o)/\|p_i - p_o\|$ is the unit vector from the obstacle to the robot i. The gradient vector field $f_i^o(p_i)$ is characterized by a respective potential function, which is given as follows:

$$V_i^o(\text{p}_i) = \frac{c_i^o}{2}\left(k_{ip}^o\left(\frac{1}{d_i^o} - \frac{1}{r^\beta}\right)^2 + k_{ip}^\delta\left(d_i^o - r^\beta\right)^2\right). \tag{4.18}$$

B. Swarm-connection control

The second component u_i^j of (4.14) is used to control the connection of the neighboring robots to avoid collisions and to keep the constant distances among them in an ordered swarm as depicted in figure 4.1. This control component is designed as:

$$u_i^j = \sum_{j=1, j\neq i}^{N}\left(f_i^j(p_i) - k_{iv}^j c_i^j\left(v_i - v_j\right) + \dot{v}_j\right). \tag{4.19}$$

In this equation, the relative velocity $(v_i - v_j)$ between the robot i and its neighbor j is used as damping term with the damping scaling factor k_{iv}^j. The scalar c_i^j is also defined similar to (4.5). To create the attractive/repulsive force field $f_i^j(p_i)$ between the robot i and its neighbor j, a corresponding potential function is proposed as:

$$V_i^j(p_i) = \frac{c_i^j}{2}\left(\left(\frac{k_{ip}^{1j}}{d_i^j} + k_d\right)^2 + k_{ip}^{2j}\left(d_i^j - r_1^\alpha\right)^2\right). \tag{4.20}$$

Taking the negative gradient of this potential function at p_i, we obtain the attractive/repulsive force as follows:

$$\begin{aligned} f_i^j(p_i) &= -\nabla V_i^j(p_i) \\ &= c_i^j\left(\left(\frac{k_{ip}^{1j}}{d_i^j} + k_d\right)\frac{k_{ip}^{1j}}{(d_i^j)^2} - k_{ip}^{2j}\left(d_i^j - r_1^\alpha\right)\right)n_i^j. \end{aligned} \tag{4.21}$$

where $n_i^j = \left(p_i - p_j\right)/\left\|p_i - p_j\right\|$ is a unit vector along the line connecting p_i *to* p_j, d_i^j is the Euclidean distance shown in equation (4.2). The positive constants k_{ip}^{1j}, k_{ip}^{2j} are used to regulate the fast collision avoidance, and the stability in the set of the α neighborhood of the robot i. The distance r_1^α is a minimum desired distance at which the attractive/repulsive forces are equal. The positive factor k_d is used as an adaptive control element to control the balance position between the attraction and the repulsion. Hence, when the swarm's size changes the formation of the swarm will be maintained.

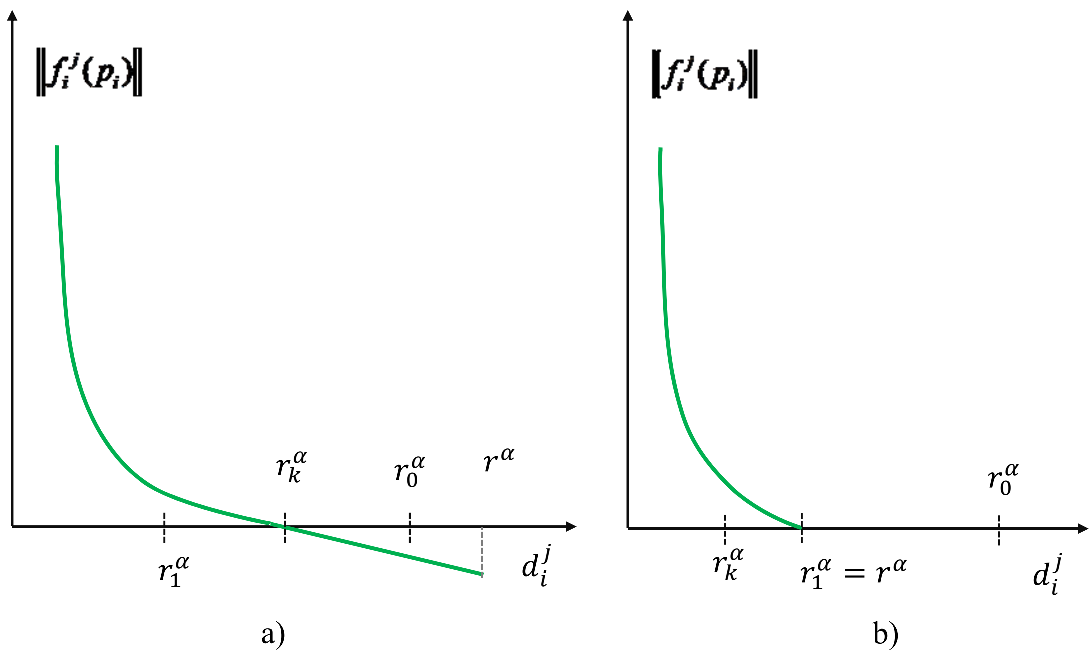

Figure 4.5: The amplitude of the force of the robot j acting on the robot i when $r_1^\alpha \le r_k^\alpha \le r_0^\alpha$ (a) and when $r_k^\alpha < r_1^\alpha$ (b).

By equating $\left(\left(k_{ip}^{1j}/d_i^j + k_d\right)k_{ip}^{1j}/(d_i^j)^2 - k_{ip}^{2j}\left(d_i^j - r_1^\alpha\right)\right) = 0$, one can find a value $d_i^j = r_k^\alpha$ at which the sum of the attractive force and the repulsive is equal to zero. In other words, if there is a given value $r_k^\alpha \geq r_1^\alpha$ and the line $-k_{ip}^{2j}\left(d_i^j - r_1^\alpha\right)$ is not changed, then the adaptive control element k_d is determined as a function of the r_k^α. This function is calculated as

$$k_d = \frac{k_{ip}^{2j}\left(r_k^\alpha - r_1^\alpha\right)\left(r_k^\alpha\right)^2}{k_{ip}^{1j}} - \frac{k_{ip}^{1j}}{r_k^\alpha} \ . \tag{4.22}$$

This equation shows that when the desired distance r_k^α changes from the minimum desired value r_1^α to the maximum desired value r_0^α then the adaptive control element k_d will automatically change in order to find the balance position, at which the connection between robot j and i is stable (see figure 4.5a). When $0 < d_i^j < r_k^\alpha$, then the robots i and j repel each other to avoid the collisions between them. Otherwise, when $r_k^\alpha < d_i^j \leq r^\alpha$, then they attract each other to achieve the equilibrium position $(d_i^j = r_k^\alpha)$ in the set of α neighborhood of robot i. When $d_i^j > r^\alpha$ there is no interaction between these members.

As described in figure 4.1, the swarm's size depends on the links between neighboring robots in the ordered swarm. Hence, when changing these links, that is, changing the desired distance r_k^α, the swarm's size will also change. Furthermore, the formation of the swarm has to shrink into a smaller formation in order that the swarm can easily pass through the narrow space between the obstacles. Therefore, the desired distance r_k^α is designed by an adaptive control force that is the average of the sum of the repulsive forces from obstacles to the swarm. This desired distance is proposed as:

$$r_k^\alpha = r_0^\alpha - c_o \sum_{i=1}^{N} \sum_{o \in N_i^\beta(t)} \left\| f_i^o\left(p_i\right) \right\|. \tag{4.23}$$

Where, the component c_o is defined as $c_o = k_c \Big/ \sum_{i=1}^{N} \sum_{o \in N_i^\beta(t)} c_i^o$, here k_c is a positive constant. The repulsive force $f_i^o\left(p_i\right)$ from the obstacle o *(o=1...M)* to the robot i is presented in (4.17). The equation (4.23) shows that when the swarm does not detect any obstacle (that is $c_o \sum_{i=1}^{N} \sum_{o \in N_i^\beta(t)} \left\| f_i^o\left(p_i\right) \right\| = 0$) then $r_k^\alpha = r_0^\alpha$ (that is, the original size of the swarm is not changed). If this adaptive control force increases, that is, the swarm is hindered more, then the r_k^α will decrease into the smaller size until the swarm can pass through these obstacles. However, if $r_k^\alpha < r_0^\alpha$, then the connection of the swarm must be broken to avoid the collisions among the member robots in the swarm, that is, in this situation there is only the repulsive force among the member robots, see figure 4.5b. In this situation, the member robots will split from their swarm to avoid obstacles. Therefore, to solve these problems, the radius of neighborhood circle r^{α} is chosen as:

$$r^{\alpha} = \begin{cases} \dfrac{3}{2} r_k^{\alpha}, & \text{if } r_1^{\alpha} \le r_k^{\alpha} \le r_0^{\alpha} \\ r_1^{\alpha}, & \textit{otherwise}, \end{cases} \tag{4.24}$$

and the adaptive control element k_d is also redesigned as follows:

$$k_d = \begin{cases} \dfrac{k_{ip}^{2j}\left(r_k^{\alpha} - r_1^{\alpha}\right)\left(r_k^{\alpha}\right)^2}{k_{ip}^{1j}} - \dfrac{k_{ip}^{1j}}{r_k^{\alpha}}, & \text{if } r_1^{\alpha} \le r_k^{\alpha} \le r_0^{\alpha} \\ -\dfrac{k_{ip}^{1j}}{r_1^{\alpha}}, & \textit{otherwise.} \end{cases} \tag{4.25}$$

Finally, from equations (4.21), (4.24), (4.25) we see that when r_k^{α} reduces from r_0^{α} to r_1^{α}, then the swarm's size also shrinks into a smaller size, but the robust connections between the neighboring robots are further maintained (see figure 4.5a). Otherwise, when $r_k^{\alpha} < r_1^{\alpha}$, then the link of the swarm is broken, there is only the repulsive force among the neighboring robots to avoid the collisions among them (see figure 4.5b).

C. Target tracking control

In order to control the robot *i* to reach the target position, the third component u_i^t in (4.14) is proposed as:

$$u_i^t = f_i^t\left(p_i\right) - k_{iv}^t\left(v_i - v_t\right) + \dot{v}_t. \tag{4.26}$$

Where, $(v_i - v_t)$ is the relative velocity vector between the robot *i* and the target with a positive constant k_{iv}^t. Under the effect of the attractive force $f_i^t\left(p_i\right)$ of the target, the robot *i* will always track the target until it approaches this target position. This attractive force is proposed as follows:

$$f_i^t(p_i) = \begin{cases} -\dfrac{k_t}{r^\tau}(p_i - p_t), & \text{if } d_i^t < r^\tau \\ -k_t \dfrac{(p_i - p_t)}{\|p_i - p_t\|}, & \textit{otherwise.} \end{cases} \tag{4.27}$$

Here $r^\tau > 0$ is the target approaching range, $(p_i - p_t)$ is the relative position vector between robot *i* and the target, and $d_i^t = \|p_i - p_t\|$ is the Euclidean distance between the robot *i* and the target. In order to adaptively control a swarm that can better avoid obstacles, this attractive force also plays an important role. The magnitude of this force is decided by the control factor k_t, which is proposed as follows:

$$k_t = \begin{cases} \dfrac{k_{ip}^t}{r_k^\alpha}, & \text{if } r_1^\alpha \le r_k^\alpha \le r_0^\alpha \\ \dfrac{k_{ip}^t}{r_1^\alpha}, & \text{if } r_k^\alpha < r_1^\alpha \ . \end{cases} \tag{4.28}$$

Where k_{ip}^t is a positive factor. Equation (4.28) shows that when the adaptive control force increases, that is, the desired distance r_k^α decreases, then the attractive force of the target is also increased. However, when $r_k^\alpha < r_1^\alpha$, the gain of this attractive force is limited by a maximum desired value that corresponds to $k_t = k_{ip}^t / r_1^\alpha$, so the swarm will avoid damage.

D. **Simulation results**

This sub-section presents the results of the simulations of the adaptive formation control algorithm of multi-robots while avoiding obstacles. For the simulations, we assume that the initial velocities of the robots and the target are set to zero, and obstacles of the environment are stationary. All robots know the position of other robots as well as the position of the obstacles and the target. The general parameters for the simulations are listed in table 4.1.

TABLE 4.1: PARAMETER VALUES

Parameter	Definition	Value
r_1^{α}	Minimum desired distance for neighbors	10
r_0^{α}	Maximum desired distance for neighbors	20
r^{β}	Obstacle detecting range	30
r^{τ}	Distance of approach to target position	50
k_{ip}^{t}	Constant for fast approach to target position	3,6
k_{iv}^{t}	Damping factor for approach to target position	1,3
k_{iv}^{j}	Damping factor for approach to balance point	1,5
$k_{ip}^{1j};k_{ip}^{2j}$	Constants for fast link between neighbors	80; 6
$k_{ip}^{o}; k_{ip}^{\delta}$	Constants for fast obstacle avoidance	95; 7
c_1	Constant	0,6
c_2	Constant	3,5

Firstly, we test the control algorithm for the robust connections in a swarm of four robots, while tracking a moving target. The target moves along the trajectory $p_t = (0.3\text{t}+400,\ -0.2\text{t}+300)^{\mathrm{T}}$. For this simulation, the initial positions of robots and obstacle are chosen as follows: $p_1 = (40, 70)^{\mathrm{T}}$, $p_2 = (30, 50)^{\mathrm{T}}$, $p_3 = (80, 10)^{\mathrm{T}}$, $p_4 = (20, 30)^{\mathrm{T}}$, $p_{o1} = (250, 140)^{\mathrm{T}}$.

The results of the simulations in figure 4.6 and figure 4.7 show that the formation of a swarm is maintained while the swarm tracks a moving target. At initial time, all robots move freely, but after a time of circa 70s they link to each other to reach the stable positions in a desired swarm. The swarm's size is kept until the swarm meets the obstacle. At time t=125s until t=220s the swarm's size is shrunk into a smaller size to avoid the obstacle, but the formation is not broken. After overcoming the obstacle the swarm's size is immediately recovered, and further maintained in an original size while tracking a moving target.

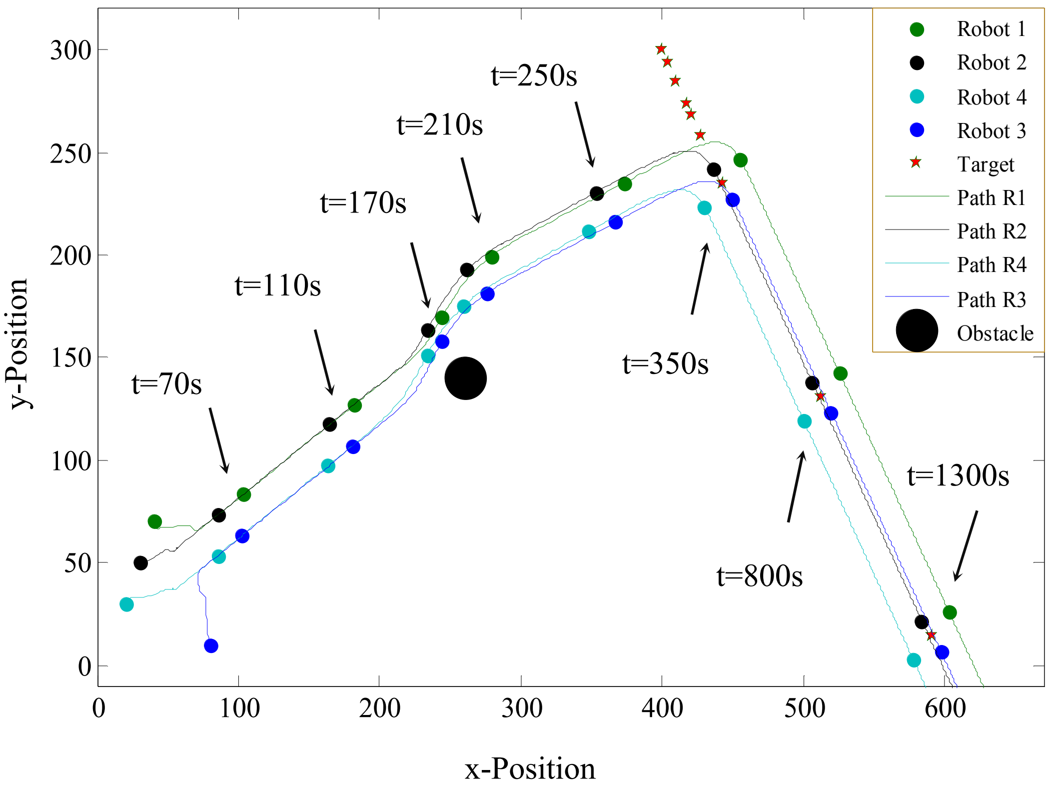

Figure 4.6: The robust connection in the formation of a swarm of four robots is maintained while tracking a moving target.

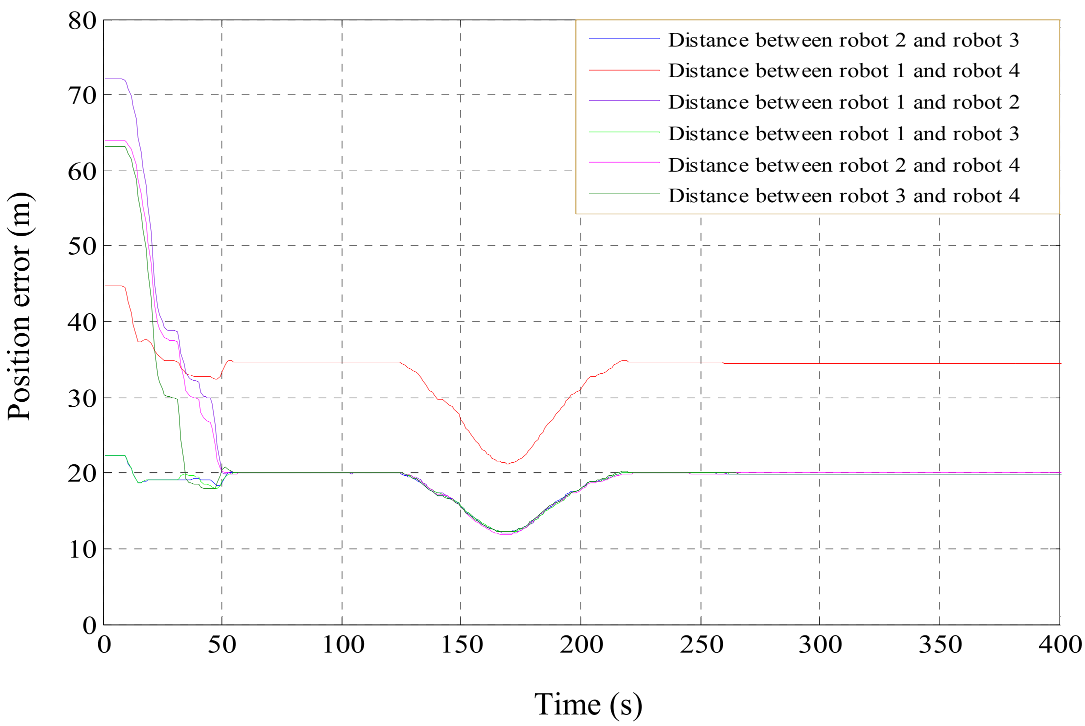

Figure 4.7: The size of a swarm of four robots simulated in figure 4.6 changes at time *t*.

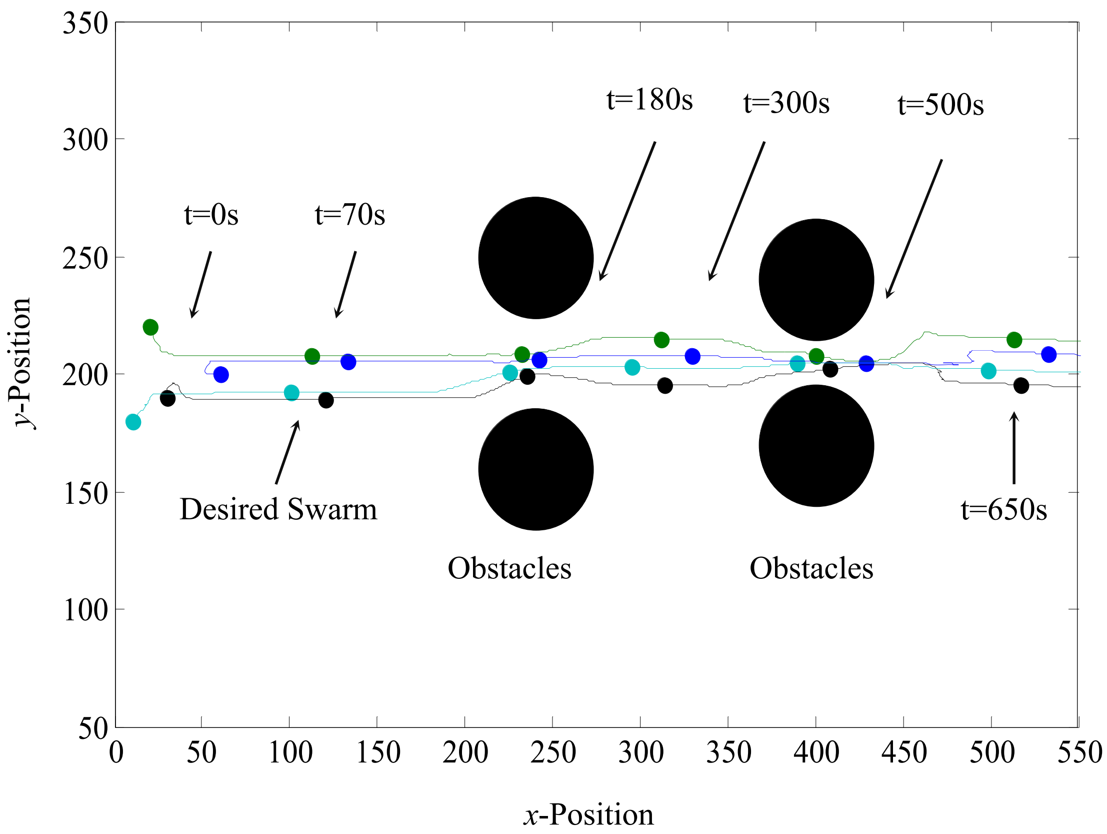

Figure 4.8: The formation adaptation of a swarm of four robots when the space between obstacles changes.

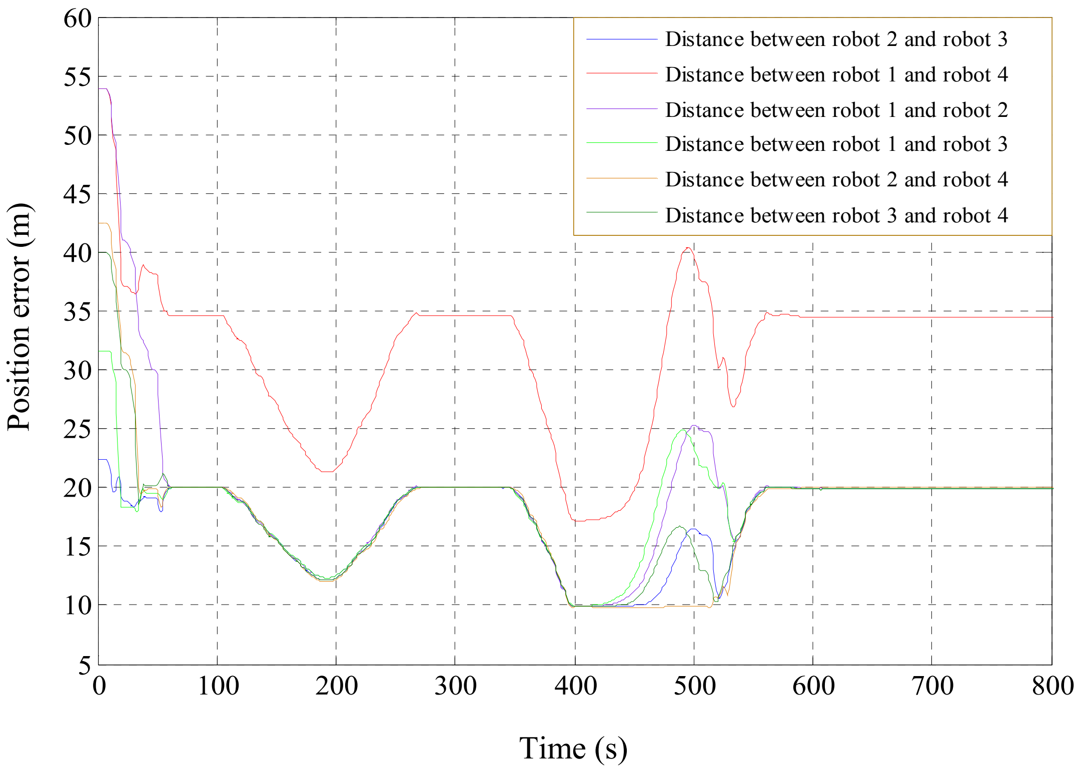

Figure 4.9: The size of a swarm of four robots simulated in figure 4.8 changes at time *t*.

Secondly, the adaptive formation control is tested while a swarm passes through the different narrow spaces between obstacles. Figure 4.8 and figure 4.9 depict the results of the simulations for the adaptation of a swarm of four robots while moving in a narrow space between the obstacles. During the period from 100s to 260s the distances between neighboring robots are reduced to smaller values (see figure 4.8). Hence, the swarm can easily overcome the spaces between these obstacles, while the swarm's structure is maintained. From t=400s to t=560s the swarm's link is broken and the robots become free robots in order to avoid obstacles. In this case, the obstacle avoidance of the free robots is successful and there are no collisions among robots (the smallest distance between neighbors is $d_i^j = r_1^\alpha$, see Figure 4.9).

Similar to the case for a swarm of four robots, figure 4.10 also shows that a swarm of seven robots adapts to the changing of the environment while reaching the position of the target. At time t=200s, the formation is shrunk in order to pass through the narrow space between obstacles, and then it is recovered as the original formation at t=200s. At time t=620s the formation is broken, and then it is redesigned in a different stable structure in order to continue to move towards the target.

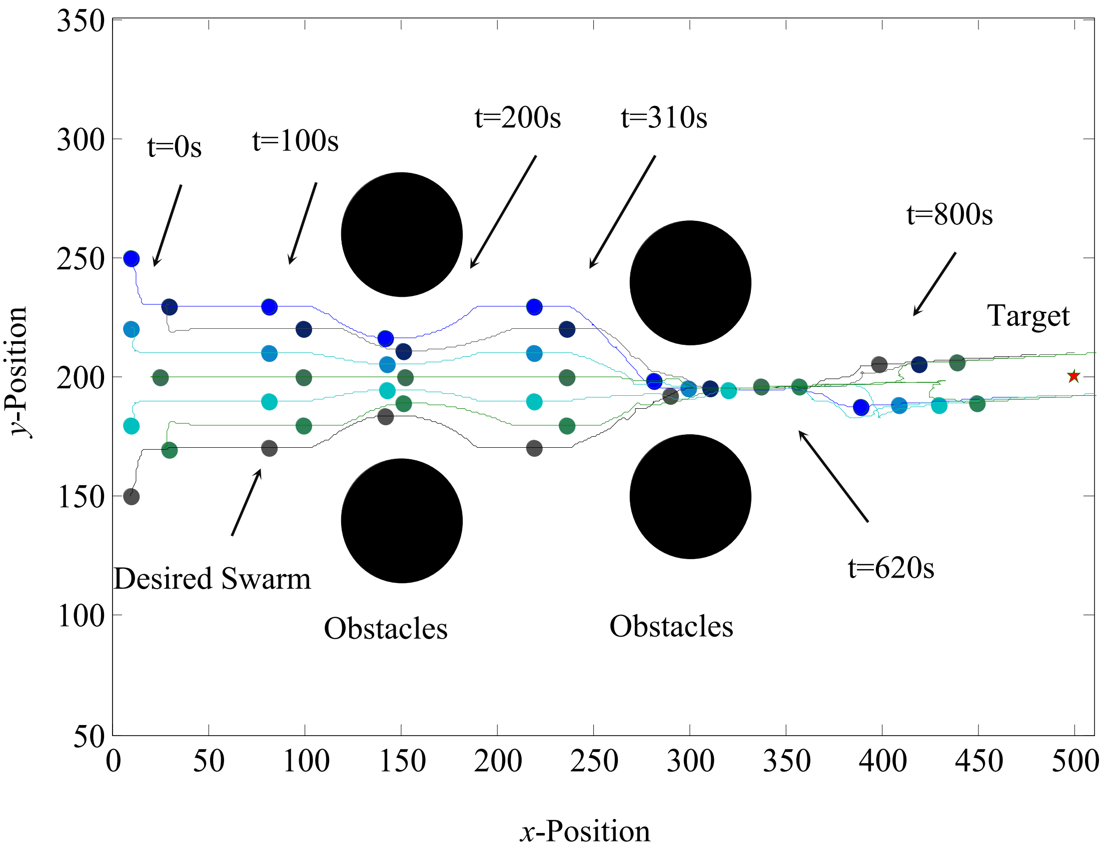

Figure 4.10: The formation adaptation of a swarm of seven robots while exiting the different narrow space between the obstacles.

4.4.3 Conclusion

This section has presented an approach for the adaptive formation control of a swarm of autonomous robots that pass through the obstacles of the dynamic environment to reach the target position. The adaptation of a swarm to the environment is built based on the change of the desired distance between the neighboring robots in the swarm. This desired distance is inversely proportional to the sum of the repulsive force from obstacles to the swarm. Information about obstacles, which each member robot detects from the environment, will be sent to all other member robots in the swarm. Therefore, the swarm's size will immediately change to adapt to the changing environment. The results of the simulations have shown that under the proposed adaptive control algorithm, a swarm of autonomous robots can easily escape the obstacles of the environment in order to reach the target. While avoiding the obstacles, the size of the swarm is automatically changed into the smaller size, so the swarm can quickly avoid the obstacles with the formation is kept. Moreover, the swarm's size reduction will help swarm can easily escape the narrow space among obstacles, but the structure of the formation of this swarm is not broken. In addition, the formation distribution to the free robots in order to avoid the collisions among the members in the swarm while escaping the narrow space between the obstacles also proves the success of our proposed approach.

4.5 Cooperative formation control in a dynamic environment

4.5.1 Problem formulation

This sub-section presents an approach to cooperative control for the formation of a swarm of autonomous robots to track a moving target in an unknown environment. This approach is based on the traditional potential fields combined with the rotational vector field. The repulsive potential field is used to repel the robots away from obstacles while the rotational vector field is added to drive the robots so as to overcome obstacles in the direction of the target's trajectory. The target's direction is determined based on the relative position between the current position and the future position of the target with the preselected time-step Δt, see figure 2.8 in chapter 2. Under the effect of the blended vector field, the autonomous robots can easily escape obstacles in order to quickly reach the target. In this approach, the neighboring robots in a swarm will be linked to each other by the attractive and the repulsive vector field between them in order to generate the stability and robustness of a formation. Moreover, information about obstacles, which each member robot detects from the environment, will be sent to all other member robots in the swarm. Hence, the formation of the swarm is maintained while tracking a moving target and the velocity of the neighboring robots in a formation is also matched while avoiding obstacle simultaneously.

The formation control for a swarm of the autonomous robots to track a moving target in an unknown environment is shown in figure 4.11. This swarm must overcome the U-shaped obstacle in order to reach the moving target. In an unknown environment, it is very difficult to determine the desired motion direction for the robots to easily escape obstacles and simultaneously reach the target quickly. The best way to solve this problem is to control the robots to avoid obstacles in the direction of the target's trajectory. Moreover, while these robots avoid obstacles, the stability and robustness of their formation must be maintained. Therefore, in order to execute this idea, each robot will be controlled by a total force that consists of the attractive force $f_i^t(p_i)$ of the target, the sum of the repulsive forces $\sum_{o \in N_i^\beta(t)} f_i^{op}(p_i)$ of the obstacles, the rotational force $f_i^{or}(p_i)$ surrounding the obstacles, the connecting force $f_i^j(p_i)$ between this robot with its neighbors, and the obstacle avoidance forces from other member robots send to this robot.

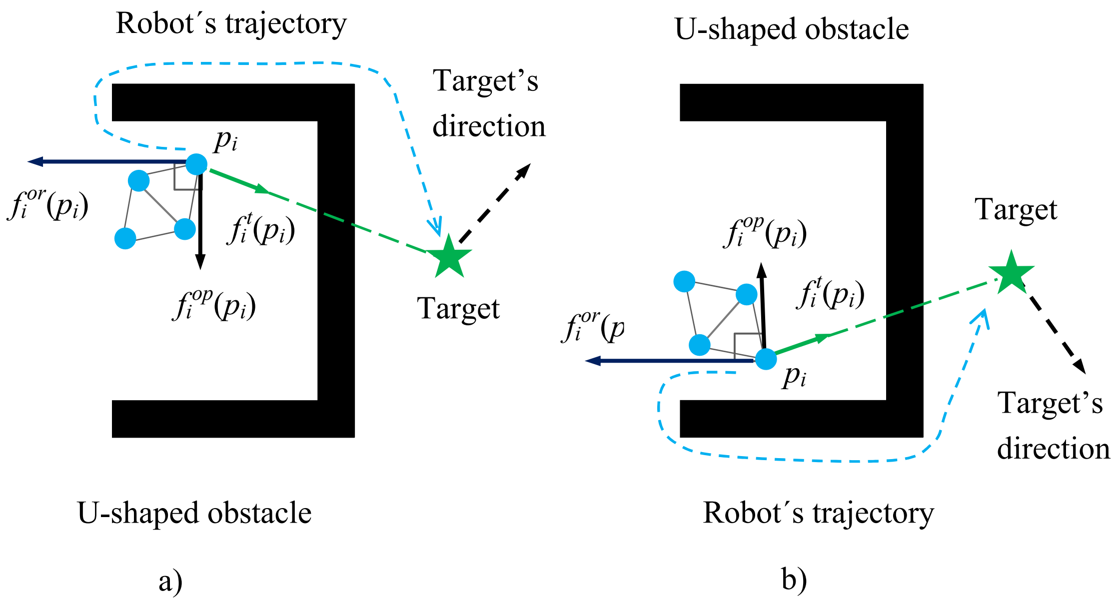

Figure 4.11: The geometric description of the obstacle avoidance and escape for a swarm of four robots while tracking a moving target: clockwise (a) and counter-clockwise (b).

4.5.2 Formation control algorithm

This section presents the formation control algorithm for a swarm of N robots, which passes through M obstacles to track a moving target. As stated above, the final aim of the member robots in a swarm is to escape obstacles of the environment to reach the target while staying together. Accordingly, the control algorithm for each member robot i of a swarm is also given as follows:

$$u_i = u_i^j + u_i^o + u_i^t. \tag{4.29}$$

In this control law, the first component u_i^j is used to control the robust connection between the neighboring robots in the formation. As shown in section 4.4, this connection is controlled based on the combination of the attractive vector field and the repulsive vector field among the neighboring robots. Furthermore, in order to obtain the quick stability at the balance point, at which the distance among the neighboring robots is constant, the relative velocity vector ($v_i - v_j$) between them is added as a damping term with the scaling factor k_{iv}^j.

The second component u_i^o of (4.29) is used to control the obstacle avoidance for each member robot of the swarm. As presented above, the final aim of this sub-section is to control the formation of a swarm to quickly escape the obstacles, but this formation is always maintained while avoiding these obstacles. In order to solve this problem, information about obstacles, which each member robot detects from the environment, will be sent to all other member robots in the swarm. Therefore, the obstacle avoidance control law is projected for each robot i as follows:

$$u_i^o = \sum_{k=1}^{N} \sum_{o \in N_k^{\beta}(t)} \left(f_k^{op}(p_k) + f_k^{or}(p_k) - k_{kv}^o c_k^o (v_k - v_o) \right). \tag{4.30}$$

Where the relative velocity vector ($v_k - v_o$) between the robot k ($k=1,2,..N$) and its neighbor-obstacle $o \in N_k^{\beta}(t)$ is used as a damping term with the damping scaling factor $k_{kv}{}^o$. The local repulsive force surrounding the obstacle $f_k^{op}(p_k)$ is used to drive the robot away from this obstacle. The local rotational force surrounding the obstacle $f_k^{or}(p_k)$ is used to drive the robot to escape this obstacle in the direction of the target's trajectory. As presented in section 2.3, the combination of these forces will help that the robot can easily and quickly exit the obstacles to continue tracking a moving target.

The third component u_i^t of (4.29) is used to control the robot to reach the target position as represented in section 4.4. However, in order to control the formation cooperation of a swarm while avoiding obstacles better, the attractive force surrounding the target must have the smaller magnitude. Hence, this attractive force is proposed as follows:

$$f_i^t(p_i) = \begin{cases} -\dfrac{k_t^\xi}{r^\tau}(p_i - p_t), & \text{if } d_i^t < r^\tau \\ -k_t^\xi \dfrac{(p_i - p_t)}{\|p_i - p_t\|}, & \text{otherwise.} \end{cases} \tag{4.31}$$

Where, the scaling factor k_t^ξ is designed, such that:

$$k_t^\xi = \begin{cases} k_t^{\xi 1} & \text{if } N_i^\beta(t) = \varnothing \text{ (empty set)} \\ k_t^{\xi 2} & \text{otherwise.} \end{cases} \tag{4.32}$$

Here, the selected positive constants $k_t^{\xi 1}$ and $k_t^{\xi 2}$ satisfy that: $k_t^{\xi 1} > k_t^{\xi 2}$.

Finally, the control law for each member robot i of a swarm is summarized as follows

$$\begin{aligned} u_i \;=\;& \sum_{j \in N_i^\alpha(t)} \left(f_i^j(p_i) - k_{iv}^j c_i^j \left(v_i - v_j \right) + \dot{v}_j \right) \\ &+ \sum_{k=1}^{N} \sum_{o \in N_k^\beta(t)} \left(f_k^{op}(p_k) + f_k^{or}(p_k) - k_{kv}^o c_k^o (v_k - v_o) \right) \\ &+ f_i^t(p_i) - k_{iv}^t (v_i - v_t) + \dot{v}_t \end{aligned} \tag{4.33}$$

4.5.3 Simulation results

This sub-section presents the simulation results of the cooperative formation control of the autonomous robots to track a moving target in a dynamic environment. The general parameters for the simulations are listed in table 4.1.

Case 1: The keeping of the formation of a swarm of four robots while avoiding the U-shaped obstacle to track a moving target is tested. For these simulations, the initial positions of the robots are chosen as follows: $p_1 = (20, 180)^T$, $p_2 = (30, 230)^T$, $p_3 = (40, 170)^T$, $p_4 = (10, 210)^T$.

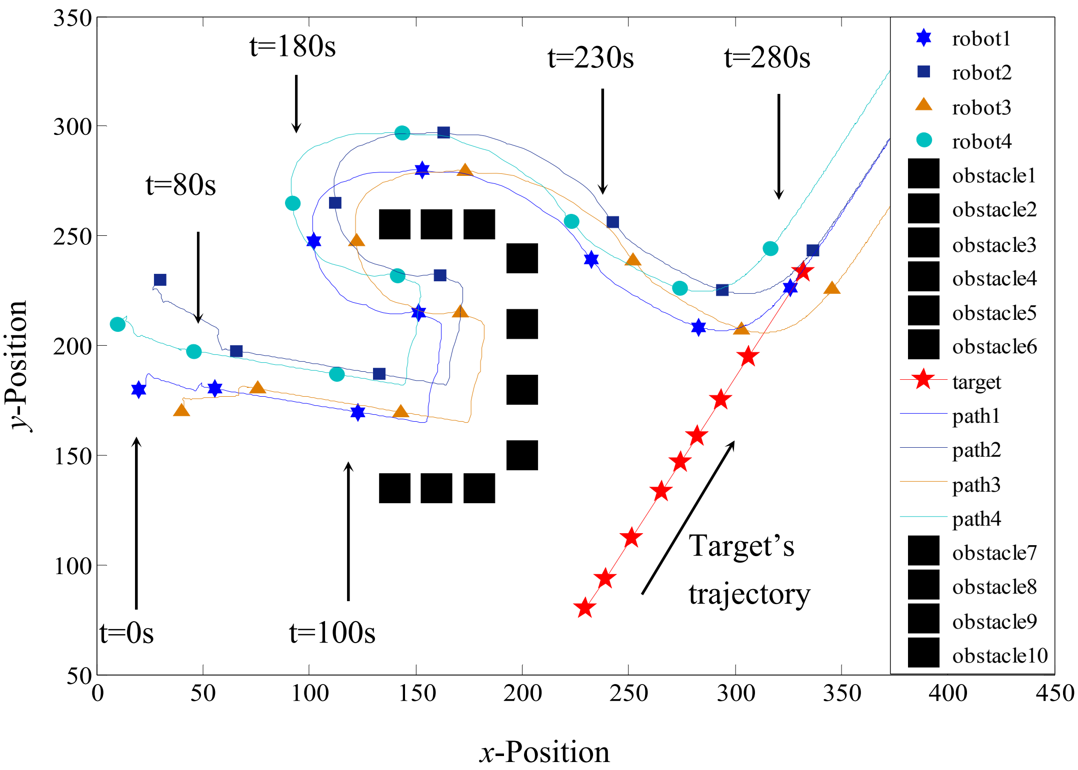

Figure 4.12: The keeping of the formation of a swarm of four robots while avoiding the U-shaped obstacle in order to track a moving target, which moves along the trajectory p_{t1}.

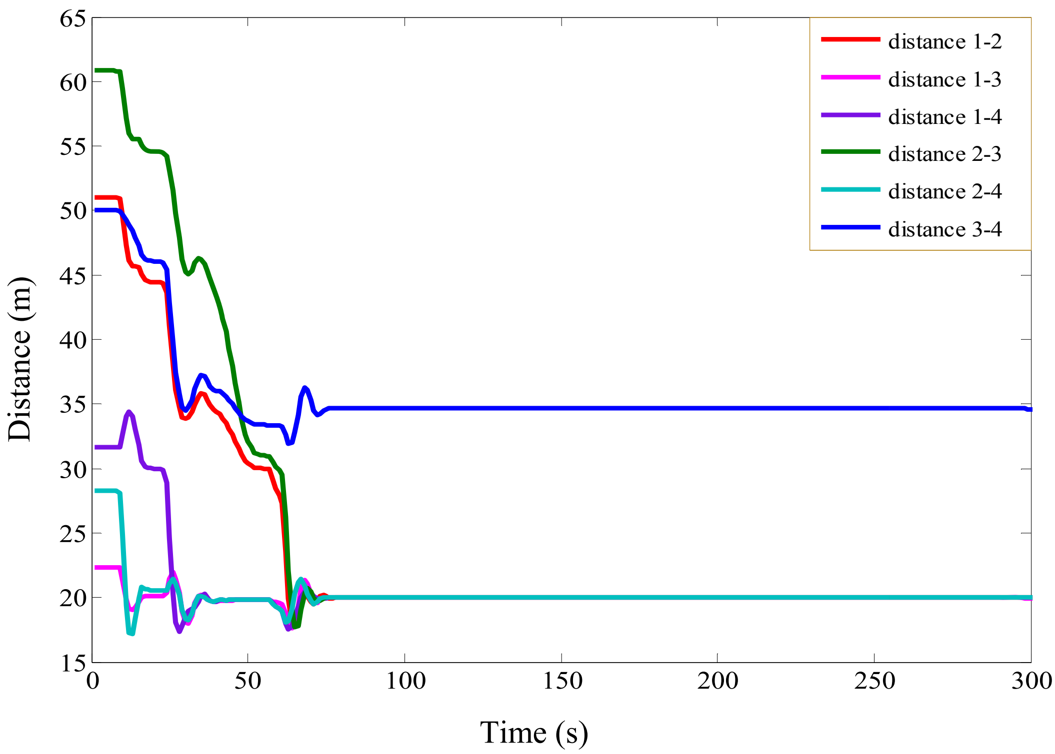

Figure 4.13: The distance between robots in the swarm of four robots simulated in figure 4.12 at time *t*.

The first situation, in which the target moves along the trajectory $p_{t1} = (0.6t+230,\ 0.9t+80)^T$, is depicted in figure 4.12 and figure 4.13. The results of the simulations in figure 4.12 and figure 4.13 show that the formation of a swarm of four robots is maintained while the swarm tracks a moving target. At initial time, all robots move freely, but after a time of circa 80s they are linked to each other in order to generate a desired formation. Then, this formation moves towards the target position by the attractive force field from the target until it meets the obstacles. When the swarm detects the obstacle it changes its moving direction to avoid collision with this obstacle and searches the new path towards the target. Figure 4.12 shows that the obstacle avoidance of the swarm according the moving direction of the target is successful. The robots can easily escape the U-shaped obstacle without breaking the formation. The distance between the neighboring robots in the swarm is kept constant, see figure 4.13. After the robots overcome the obstacle, they continue to chase the target until the swarm reaches this target at time t=280s, see figure 4.12.

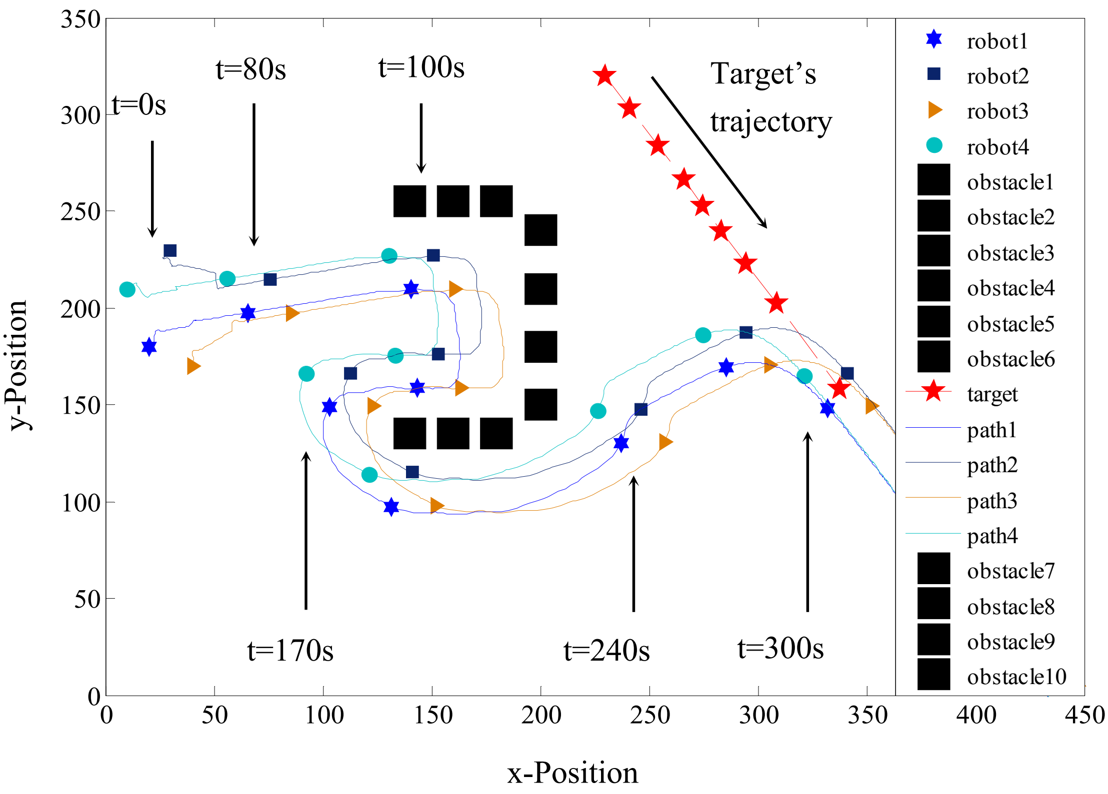

Figure 4.14: The keeping of the formation of a swarm of four robots while avoiding the U-shaped obstacle to track a moving target that moves along the trajectory as $p_{t2} = (0.6t + 230,\ -0.9t + 320)^T$.

The second situation, in which the target moves along the trajectory as $p_{t2} = (0.6t+230,\ -0.9t+320)^{T}$, is simulated in figure 4.14. The simulation result in this case shows the intelligence of a swarm while pursuing a moving target. The robots in the swarm move according to the direction of a moving target in order to quickly catch the target. While avoiding the obstacles, the connections among the neighboring robots are stably kept.

Case 2: The keeping of the formation of a swarm of four robots while avoiding the wall-shaped obstacle to track a moving target is tested. In this case, the target's trajectory is selected as $p_{t3} = (-0.4t+150,\ -0.9t+250)^{T}$. For this simulation, the initial positions of the robots are chosen as follows: $p_1 = (270, 320)^{T}$, $p_2 = (300, 300)^{T}$, $p_3 = (310, 340)^{T}$, $p_4 = (330, 310)^{T}$. The simulation result depicted in figure 4.15 shows that the swarm of four robots successfully escapes the wall-shaped obstacle. The robots can quickly exit this wall-shaped obstacle in order to move towards the target. The directional movement of these robots is driven towards the right of the wall-shaped obstacle (clockwise direction) at time t=120s. The change in movement direction helps the robots avoid the collisions with the obstacles, and find the fastest way to chase the target. Moreover, the formation of the robots is not broken while avoiding the obstacles.

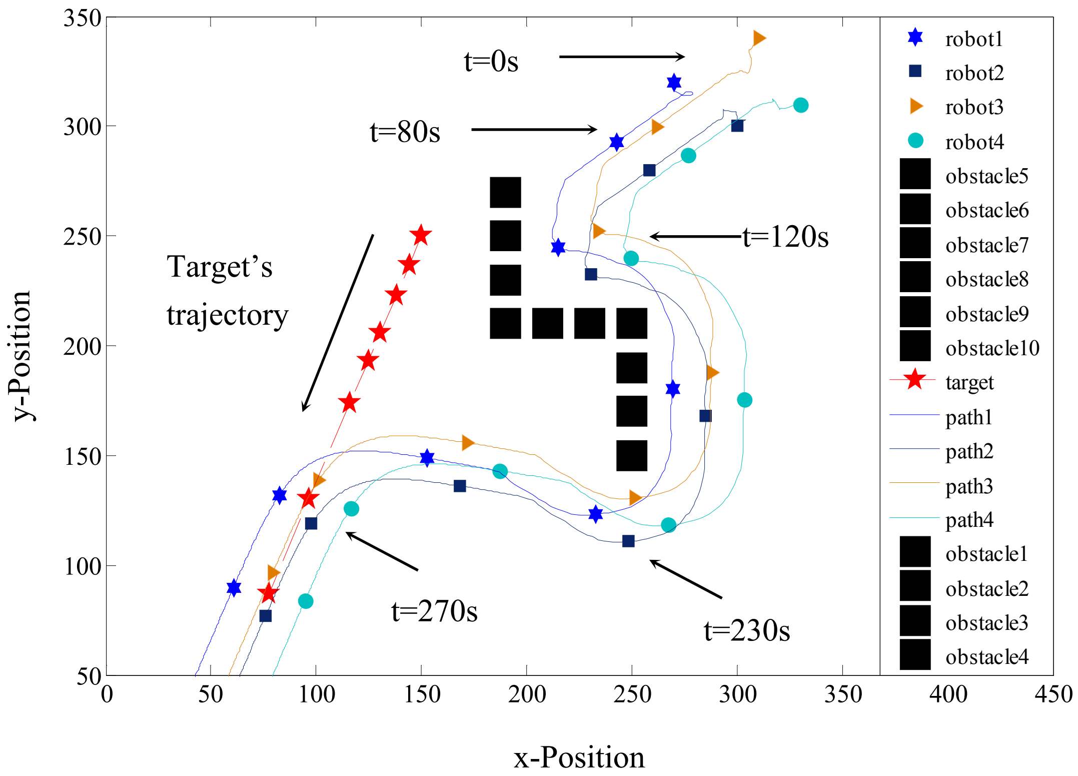

Figure 4.15: The keeping of the formation of a swarm of four robots while avoiding the wall-shaped obstacle to track a moving target, which moves along the trajectory p_{t3}.

4.5.4 Conclusion

This section has presented an approach to the cooperative control for the formation of a swarm of autonomous robots to track a moving target in a dynamic environment based on the combination of the potential force fields and the rotational force field. The rotational force field is added to help the robots to quickly escape obstacles. The movement direction for the robots to avoid obstacles is designed to be in the moving direction of the target, such that the robots can easily escape the obstacles and find the fastest path towards the target. The results of the simulations have shown that, under the effect of the blended force field, a swarm of autonomous robots can easily find a path to track a moving target in a environment, in which there are the different obstacles. Using the added rotational force field, the obstacle avoidance of this swarm is successfully achieved. Moreover, the robots in a swarm are connected to each other and they obtain the information about the obstacles of the environment from other member robots. Thus, the formation of a swarm is maintained while avoiding the obstacles in order to track a moving target.

5 Merging and Splitting in a Mobile Sensor Network

5.1 Introduction

In recent years, the mobile sensor network has been an interesting research topic in the control community all over the world [39]-[59]. Its potential applications in many areas, such as search and rescue missions, and forest fire detection and surveillance, is the motivation for this attraction.

One of the interesting issues in the mobile sensor network to reach the target position is flocking control [34]-[38]. The sensors in the network have to link with each other in order to avoid collision and maintain their velocity during tracking. Obstacle avoidance [16, 17] is also an interesting topic in path planning for autonomous mobile sensors to reach the target. The artificial potential field is known as a positive method in order to solve these problems. Recently, the artificial vector field method has been widely studied and powerfully applied to formation control of a swarm of multi-agents to reach a target in a dynamic environment, see [23]-[50].

Furthermore, the control of a mobile sensor network is important and presents two main issues: sensor splitting and sensor merging. Sensor splitting arises when the new targets appear, automatically splitting from the main group of the moving sensors into the subgroups in order to track the new targets. In contrast, sensor merging can occur when targets disappear, forcing those subgroups to redistribute to the remaining sensor groups still tracking their targets. Additionally, sensor merging is required at operation start time due to initial sensor placements. Although this topic is very interesting, and has potential applications in military area as well as in civilian area, but the research results in this field are still very limited. The published literature has mainly focused on the control for the mobile sensor network to reach a single target. Using the artificial attractive potential field, which is generated from the target and has decreasing amplitude to the target's position, the free sensors will automatically meet and connect to each other during tracking [39]-[50]. However, in practice, it is very difficult to execute this method, as the velocity of a free sensor is very high when it is far from the target.

In this chapter, we propose a novel approach to control the sensor merging/spitting in a mobile sensor network while tracking the moving targets in a dynamic environment. This approach is developed based on the traditional potential field method [14]-[22] combined with the geometry method and the energy level partitioning method. In this approach, the invariable global attractive force field surrounding the target is used to drive all sensors in a group towards their target. Simultaneously, a constant global attractive force field is also generated from the virtual leader of a group in order to control the free-sensors to combine into their group. In a group, a member-sensor, which has the shortest distance to the target of this group, is chosen as the virtual leader. Under the effect of the total force field, which contents the attractive force field of the target and the virtual leader, the free-sensors will usually move towards their group during tracking because the attractive force of the virtual leader is always designed larger than the global attractive force of the target. In contrast, sensor splitting, wherein a sensor group is broken into subgroups in order to track the new targets, is performed by the geometry method. In this method, when a new target appears, a boundary line through the target position and the center of the main group is generated as the basis for the sensor splitting. The sensors that lie together on one side of this boundary line will form to a new subgroup with a new virtual leader

The rest of this chapter is organized as follows: The sensor merging control method based on the energy level partitioning surrounding the virtual leader is given in the next section. Section 5.3 presents the sensor splitting control method. In section 5.4, the general controller for each sensor in a mobile sensor network is given. The results of the simulations are given in section 5.5. Finally, section 5.6 concludes this chapter.

5.2 Sensor merging control

5.2.1 Problem statement

In this section we consider a network of N mobile sensors ($N \geq 2$) that track a moving target in a two-dimensional Euclidean space $\{R^2\}$ with M obstacles in the environment. The free-sensors have to merge into the formation of a group and become new members. The member-sensors in a swarm will connect with their neighboring sensors in order to generate a stable formation without collisions. Each sensor, which is assumed as a moving point in the space, is also described by the dynamic model as:

$$\begin{aligned} \dot{p}_i &= v_i \\ \dot{v}_i &= u_i, \qquad i = 1,..,N. \end{aligned} \tag{5.1}$$

Where $(p_i,\ v_i,\ u_i) \in \{R^2\}$ are the position, the velocity, and the control input of the robot i, respectively.

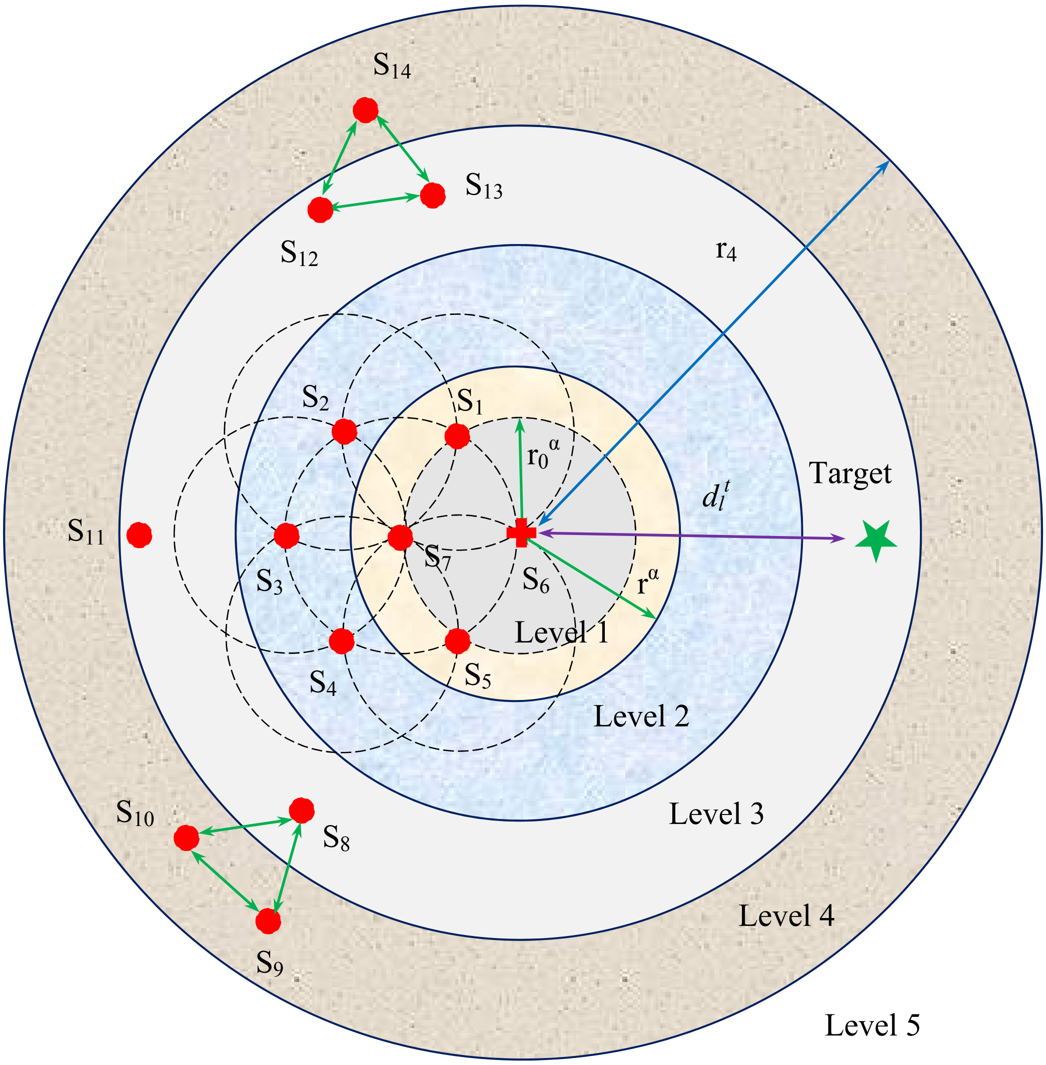

Figure 5.1: The description of a group of the mobile sensors in the energy level partitioning surrounding the virtual leader while tracking a moving target.

First of all, in order to simplify the analysis, we make the following remarks, definitions, and assumptions:

Remark 5.1. In a mobile sensor network, each subgroup has the mission to track a respective moving target. In each subgroup, a sensor which has the shortest distance to the target will be selected as the virtual leader of this group. This virtual leader has the mission to attract free-sensors towards the formation of this group. Under the attractive force of this leader, these free-sensors will quickly combine into the formation and become the new members. Moreover, the formation of a group will be maintained by the connection between neighboring member-sensors in this formation during tracking without collisions. A sensor is determined as a free-sensor or member-sensor in a group which depends on the relative position between this sensor and the leader through energy levels as described on figure 5.1.

Definition 5.1. A sensor *j* is the neighbor of the sensor *i* if it lies in the limited communication range r^α of the sensor *i* (radius of neighborhood circle, shown in figure 5.1). During the motion of the sensors in a group the relative position between them can change, hence the neighbors of each sensor can also change. Therefore, in general we can define the set of the sensors in the neighborhood of sensor *i* at time *t* as follows:

$$N_i^\alpha(t) = \left\{ j : d_i^j \le r^\alpha, j \in \{1,..,N\}, j \ne i \right\}. \tag{5.2}$$

Where $d_i^j = \left\| p_i - p_j \right\|$ is the Euclidean distance between sensor *i* and sensor *j* in the space. For example, in figure 5.1, the sensor S_1 has three neighbors: S_2, S_6 and S_7. As depicted in figure 5.1, in order to generate a stable and robust formation each member sensor in a group is only allowed to connect with its neighboring sensor. Thus, to perform this problem the radius r^α of neighborhood circle can be chosen, such that: $r_0^\alpha < r^\alpha < \sqrt{3} r_0^\alpha$.

Definition 5.2. A sensor *i* (*i=1,..,N*) is the member of a group if it lies in the energy level 1 of the virtual leader. In other words, it is the neighbor of the leader (the distance from this sensor to the leader is smaller than the radius of the neighboring circle r^α). For example, in figure 5.1, sensor S_6 is selected as the virtual leader of a group, hence the sensors S_1, S_5, S_6 and S_7 are member-sensors of this group, because these sensors lie in the energy level 1 of the virtual leader. In case, sensor *i* lies in the energy level n $(n \ne 1)$, it will be a member-sensor of a group if it has at least two neighbors which lie in the smaller energy level corresponding to (*n-1*), or it has at least one neighbor that lies in the energy level (*n-1*) and at least two neighbors that have the same energy level n with sensor *i.* For example, in figure 5.1, sensor S_2 (S_2 lies in the energy level 2) is a member sensor, because it has two neighbors (S_1, S_7) that lie in the energy level 1. Similarly, sensors S_4 and S_{14} also are the member sensors in the group of the virtual leader S_6. Furthermore, sensor S_3 (S_3 lies in the energy level 2) is a member sensor, because it has one neighbor that lies in the energy level 1 and two neighbors (S_2, S_4) that lie in the energy level 2 . In other cases, sensor *i* is a free-sensor, for example sensors S_{11}, S_{12} and S_{13} in figure 5.1. Furthermore, sensors S_8, S_9 and S_{10} also are free-sensors, although they are connected in a sub-formation.

Assumption 5.1. The position $p_i = (x_i, y_i)^T$ and the velocity $v_i = (v_{ix}, v_{iy})^T$ of the robot *i* are known. The robot is equipped with sensors such as cameras, sonars, laser sensors, GPS sensors, and associated algorithms, etc., to estimate the trajectory (position $p_t = (x_t, y_t)^T$ and velocity $v_t = (v_{tx}, v_{ty})^T$) of the target precisely.

Assumption 5.2. The velocity of the moving target is limited by the maximum velocity of the robot $\|v_t\| < v_{r\max}$.

Assumption 5.3. The robot can sense the position $p_o = (x_o, y_o)^T$ and the velocity $v_o = (v_{ox}, v_{oy})^T$ of the obstacles in the environment precisely.

5.2.2 Sensor merging control algorithm

This sub-section presents an approach for the free sensor merging control into a group. This approach is based on the energy level partitioning method surrounding the virtual leader. The main aim of this approach is to control all free robots that can quickly find and reach a group in order to become the members in the formation of this group. According to the above analyses, the sensor merging control algorithm for a group of *N* sensors while tracking a moving target is built by the following steps:

Step 1. Choose a sensor, which has the shortest distance to the target, as the virtual leader of the group. The distance from the virtual leader to the target is computed as:

$$d_l^t = \min\left\{d_i^t = \|p_i - p_t\|, i = 1,...N\right\}. \tag{5.3}$$

Step 2. Partition the energy levels from the selected virtual leader in order to determine that the sensor *i* is a free-sensor or a member-sensor of the group. Let r_n be the radius of the energy level n from the virtual leader. Then, the magnitude of the energy level *n,* (*n*=*{1,2,...}*), is described as $r_n - r_{n-1}$, and it satisfies the equation $r_0^\alpha = r_n - r_{n-1}$, see figure 5.1. Here, the positive constant r_0^α is a minimum desired distance between the neighboring sensors, at which the attractive and repulsive forces between these neighboring sensors balance. In general, the radius of the energy level *n* is built as follows:

$$r_n = \lambda + n r_0^\alpha. \tag{5.4}$$

In this equation, the positive constant λ is used to determine the radius of the neighborhood circle r^α ($r^\alpha = \lambda + r_0^\alpha$), see figure 5.1. Now, we consider a sensor *i* that has the relative distance to the leader, such that: $d_i^l = \left\| p_i - p_l \right\|$. In order to determine the energy level n_i, at which the sensor i is existing, we build an inequality as follows:

$$d_i^l \le \lambda + n_i r_0^\alpha, \; i = 1,..N \; and \; n_i = \{1,2,3...\}. \tag{5.5}$$

From this inequality, the energy level n_i is proposed as follows:

$$n_i = ceil\left((d_i^l - \lambda)/ r_0^\alpha\right). \tag{5.6}$$

Let c_e be the number of the sensors that lie in the energy level $(n_i - 1)$ and are the neighbors of the sensor i at time t. This component is described as follows:

$$c_e = \sum_{j=1, j \ne i}^{N} c_j \,. \tag{5.7}$$

Here, the scaling factor c_j is defined as:

$$c_j = \begin{cases} 1, & if \; n_j = n_i - 1 \; and \; j \in N_i^a(t) \\ 0, & otherwise \end{cases}. \tag{5.8}$$

Where, n_j is the energy level, at which the sensor j $(j = 1,2,..N; \; j \ne i)$ is existing. Similar to equation (5.7), the number of the sensors that lie in the energy level n_i and are the neighbors of the sensor i at time t, is described as follows:

$$c_m = \sum_{k=1, k \ne i}^{N} c_k \,. \tag{5.9}$$

Here, the scalar c_k is also described as follows:

$$c_k = \begin{cases} 1, & if \; n_k = n_i \; and \; k \in N_i^a(t) \\ 0, & otherwise. \end{cases} \tag{5.10}$$

Where, n_k is the energy level which the sensor k $(k = 1,2,..N; \; k \ne i)$ is owning.

Finally, a sensor that is determined to be a free-sensor or member-sensor of a group is presented on the flowing diagram in figure 5.2.

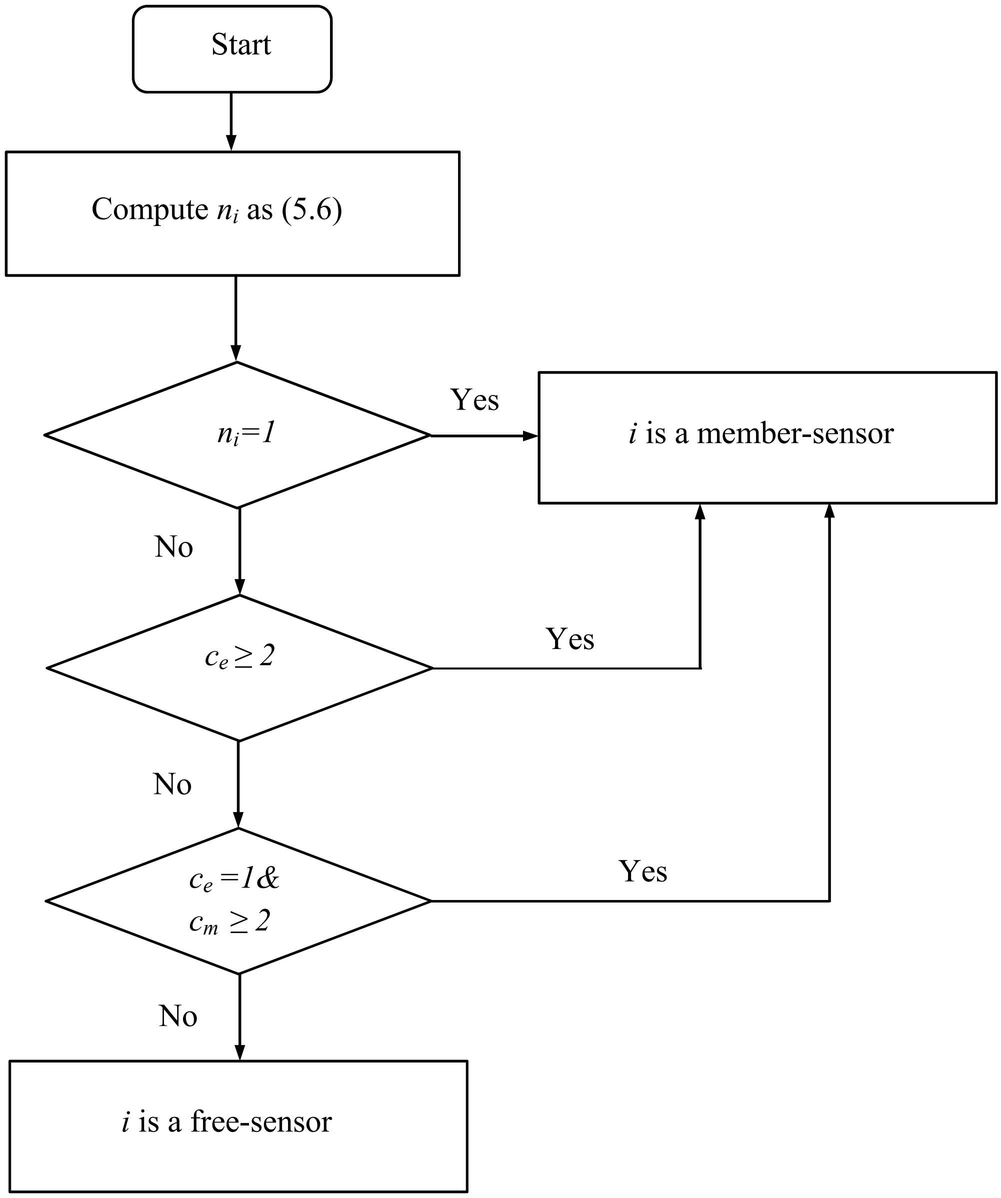

Figure 5.2: The flowing diagram for the determination of the sensor i $(i = 1,..,N)$ that is a free-sensor or a member-sensor in a group.

Step 3. Design the sensor merging control law for each sensor i in a group as follows:

$$u_i^l = -k_{ip}^l c_i^l e_i^l - k_{iv}^l c_i^l (v_i - v_l). \tag{5.11}$$

In this control law, the relative velocity vector ($v_i - v_l$) between the sensor i and the virtual leader is used as a damping term with the damping scaling factor $0 < k_{iv}^l$. k_{ip}^l is a positive gain factor, and $e_i^l = (p_i - p_l) / \|p_i - p_l\|$ is a unit vector from the leader to the sensor i. The scalar c_i^l is described as follows:

$$c_i^l = \begin{cases} 1\,, & if\ i\ is\ free-sensor \\ 0\,, & otherwise \end{cases}. \tag{5.12}$$

Equation (5.12) shows that only free-sensors are affected by the attractive force of the virtual leader. In other words, using the attractive force from the virtual leader, all free sensors will find and approach their group to become the new members in this group. Then, these new members will combine with other members in the group to continue tracking their target.

Remark 5.2. If a target disappears at time *t,* the sensors that are tracking this target will automatically combine with the nearest existing subgroups as the free-sensors of these subgroups.

5.3 Sensor splitting control

This sub-section presents the sensor splitting control method from a group, which is tracking a moving target, into subgroups to track the new targets when these new targets appear. The idea to design this method is depicted in figure 5.3.

As shown in figure 5.3 the line $f(x,y)=0$ through the center A of the main group and the new target splits the main group, which is tracking the old target, into two sub-groups. The first sub-group (subgroup 1) consists of the sensors that lie on the side that contains the old target while the second subgroup (subgroup 2) consists of the sensors that lie on the opposite side. Assume that in the main group there are *N* sensors ($N \geq 2$), and each sensor $i\ (i = 1,..,N)$ is located at the position $p_i = (x_i,\ y_i)^T$. Then, the center of this main group at time *t* is calculated, such that:

$$A(t) = \frac{1}{N}\sum_{i=1}^{N} p_i(\mathrm{t}). \tag{5.13}$$

Let $p_{t_1}(t) = (x_{t_1}(t),\ y_{t_1}(t)\ )^T$, $p_{t_2}(t) = (x_{t_2}(t),\ y_{t_2}(t)\ )^T$, and $A(t) = (x_A(t),\ y_A(t))^T$ be the position of the target 1 (old target), the position of the target 2 (new target), and the position of the center of the main group at time *t*, respectively. According to [96], [97], the boundary line $f(x,y)=0$ is given , such that:

$$f\left(x,y\right) = \frac{x - x_A(t)}{x_{t_2}(t) - x_A(t)} - \frac{y - y_A(t)}{y_{t_2}(t) - y_A(t)}. \tag{5.14}$$

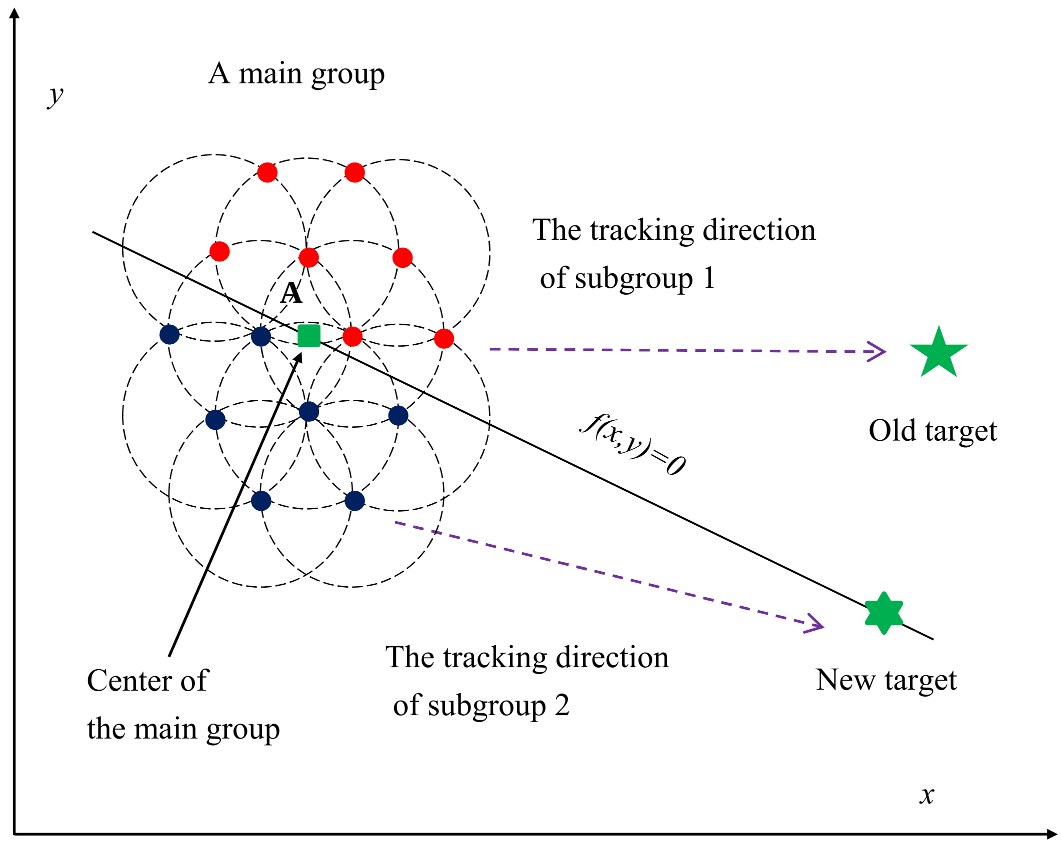

Figure 5.3: The description of the geometry method for the sensor splitting control from a main group, which is tracking the old target, into two subgroups (subgroup 1 and subgroup 2) when the new target appears. The subgroup 1 continues to track the old target while the subgroup 2 is split to track the new target.

The boundary line $f(x,y)=0$ will split the coordinate plane xy into two half-planes. One side of this boundary line consists of all points that satisfy the inequality $f(x,y)<0$. All points on the opposite side satisfy the inequality $f(x,y)>0$. Hence, in order to determine if a sensor i is lying on a particular side of the boundary line $f(x,y)=0$, we use the old target as a test-point as follows: if sensor i lies on the side containing the test-point, then $f(p_{t_l}(t))f(p_i(t))>0$. In contrast, if sensor i and the test-point lie on the different sides of the boundary line $f(x,y)=0$, then $f(p_{t_l}(t))f(p_i(t))<0$. In the special case, if the old target lies on the boundary line $f(x,y)=0$, then the sensors on one side of this boundary line are selected to track a target, while the sensors on the remaining side of this boundary line will continue to track the remaining target. In order to solve this problem, we can choose a random test-point (for example a test-point $D(t)=(x_{t_1}(t)+\Delta,\ y_{t_1}(t)+\Delta\ )^T$, here Δ is a positive constant) to determine that the sensor i is lying on which side of the boundary line $f(x,y)=0$.

If the sensors i $(i = 1,..,N)$ of the main group satisfy the inequality $f(D(t))f(p_i(t)) \geq 0$, then they are selected as the members of a subgroup (for example subgroup 1) to track the old target. The remaining sensors of the main group will be redesigned into a new group (for example subgroup 1) to track the new target. Finally, after the sensor partitioning, the mission of each sensor is to reach its target, and this mission is maintained until this target disappears or some new targets appear.

Based on the above analysis, the control law for each sensor i to track its target is proposed as follows:

$$u_i^t = f_i^{t_m}(p_i) - k_{iv}^t(v_i - v_{t_m}) + \dot{v}_{t_m}. \tag{5.15}$$

Similar to equation (5.11), in this equation, the relative velocity vector $(v_i - v_{t_m})$ between the sensor i and its target is added as a damping term with the positive factor k_{iv}^t. The force field $f_i^{t_m}(p_i)$ is designed as an attractive potential field surrounding each target t_m with $m=\{1,2,3...\}$. Therefore, under the effect of this force field, each sensor i will always be attracted towards its target until it reaches this target. This attractive force field is designed as follows:

$$f_i^{t_m}(p_i) = \begin{cases} -\dfrac{k_{ip}^t}{r^\tau}\left\| p_i - p_{t_m} \right\| e_{it_m}, & if \;\; d_i^t < r^\tau \\ \\ -k_{ip}^t e_{it_m}, & otherwise. \end{cases} \tag{5.16}$$

Here $r^\tau > 0$ is the close range surrounding the target, at which the sensor's speed is reduced before reaching the target, and $e_{it_m} = (p_i - p_{t_m}) / \left\| p_i - p_{t_m} \right\|$ is the unit vector directing from the target (p_{t_m}, v_{t_m}) to the sensor (p_i, v_i). Equation (5.16) shows that the magnitude of the attractive force $f_i^{t_m}(p_i)$ is depending on the positive control factor k_{ip}^t and the distance $d_i^{t_m} = \left\| p_i - p_{t_m} \right\|$ between sensor i and its target.

Example: The sensor splitting control algorithm from a main group into two sub-groups to track two moving targets is presented in Algorithm 5.1.

Algorithm 5.1: Sensor splitting control

Update data: The actual state of the robots $(p_i,\ v_i$ with $i = 1,..N)$, the targets $(p_{t_m},\ v_{t_m}$ with $m = 1;2)$, and the actual position A of the center of the main group.

Build the boundary line $f(x,y)=0$ that satisfies equation (5.14), and calculate $f(p_{t_1})f(p_i)$.

for $i=1:N$

if $f(p_{t_1})f(p_i) > 0$ **then**

Sensor i is split into the sub-group 1 to track the target 1, and the target tracking control law is given as

$$u_i^t = f_i^{t_1}(p_i) - k_{iv}^t(v_i - v_{t_1}) + \dot{v}_{t_1};$$

elseif $f(p_{t_1})f(p_i) < 0$ **then**

Sensor i is split into the sub-group 2 to track the target 2, and the target tracking control law is given as

$$u_i^t = f_i^{t_2}(p_i) - k_{iv}^t(v_i - v_{t_2}) + \dot{v}_{t_2};$$

elseif $f(p_{t_1})f(p_i) = 0$ **then**

Choose a test-point $D = (x_{t_1} + \Delta,\ y_{t_1} + \Delta\ )^T$, and calculate $f(D)f(p_i)$.

if $f(D)f(p_i) \geq 0$ **then**

Sensor i is the member of the sub-group 1 that will track the target 1, and the target tracking control law is given as

$$u_i^t = f_i^{t_1}(p_i) - k_{iv}^t(v_i - v_{t_1}) + \dot{v}_{t_1};$$

else

Sensor i is the member of the sub-group 2 that will track the target 2, and the target tracking control law is given as

$$u_i^t = f_i^{t_2}(p_i) - k_{iv}^t(v_i - v_{t_2}) + \dot{v}_{t_2}$$

end

end

end

5.4 General controller

This subsection presents the general controller for each sensor i $(i = 1,..,N)$, which has the described dynamic model as equation (5.1), in a mobile sensor network. This controller is proposed as follows:

$$\begin{aligned} u_i \;=\; & \sum_{j\in N_i^\alpha(t)} \left(f_i^j(p_i) - k_{iv}^j(v_i - v_j) + \dot{v}_j \right) \\ + \; & \sum_{o\in N_i^\beta(t)} \left(f_i^o(p_i) - k_{iv}^o(v_i - v_o) \right) \\ + \; & c_i^l \left(-k_{ip}^l(p_i - p_l)/\|p_i - p_l\| - k_{iv}^l(v_i - v_l) \right) \\ + \; & f_i^{t_m}(p_i) - k_{iv}^t(v_i - v_{t_m}) + \dot{v}_{t_m}. \end{aligned} \tag{5.17}$$

In this controller, the first component $\sum_{j\in N_i^\alpha(t)} \left(f_i^j(p_i) - k_{iv}^j(v_i - v_j) + \dot{v}_j \right)$ is used to control the connection between sensor $(p_i,\ v_i)$ with its neighbors in the formation of a group. The pair (p_j, v_j) is the position and velocity of the sensor j, $j \in N_i^\alpha(t)$, see Definition 1. As presented in chapter 4, the attractive/repulsive force field $f_i^j(p_i)$ is given such that $f_i^j(p_i) = \left(\left((d_i^j)^{-1} - (r_0^\alpha)^{-1} \right) k_{ip}^{1j}(d_i^j)^{-3} - k_{ip}^{2j}\left(d_i^j - r_0^\alpha \right)(d_i^j)^{-1} \right)\left(p_i - p_j \right)$. Here, k_{iv}^j, k_{ip}^{1j}, k_{ip}^{2j} are the positive factors. $r_0^\alpha > 0$, and $d_i^j = \|p_i - p_j\|$ are the collision range surrounding each robot, and the Euclidean distance between the robot i and the robot j, respectively. The second component $\sum_{o\in N_i^\beta(t)} \left(f_i^o(p_i) - k_{iv}^o(v_i - v_o) \right)$ is used to drive the sensor i to avoid the obstacles (p_o, v_o) of the environment. Similar to the Definition 1, the set of the obstacles in the β neighborhood of sensor i at time t is also defined as $N_i^\beta(t) = \left\{ o, d_i^o \le r^\beta, o = 1,..M, o \ne j \right\}$, here $r^\beta > 0$ and $d_i^o = \|p_i - p_o\|$ are the obstacle detection range and the Euclidean distance between the robot i and the obstacle o, respectively. The repulsive force field $f_i^o(p_i)$ is also given such that $f_i^o(p_i) = \left(\left((d_i^o)^{-1} - (r^\beta)^{-1} \right) k_{ip}^{1o}(d_i^o)^{-3} - k_{ip}^{2o}\left(d_i^o - r^\beta \right)(d_i^o)^{-1} \right)\left(p_i - p_o \right)$. Here, $k_{iv}^o, k_{ip}^{1o}, k_{ip}^{2o}$ are the positive factors. $(p_i - p_o)$, $(\mathrm{v}_i - v_o)$ are the relative position and velocity between sensor i and its neighboring obstacle o, respectively. The third component $c_i^l \left(-k_{ip}^l(p_i - p_l)/\|p_i - p_l\| - k_{iv}^l(v_i - v_l) \right)$, which is presented in section 5.2.2, is used to control the sensor i merging into its group if it is a free-sensor. The fourth component $f_i^{t_m}(p_i) - k_{iv}^t(v_i - v_{t_m}) + \dot{v}_{t_m}$ is used to control the sensor i tracking its target as presented in section 5.2.3.

5.5 Simulation results

This section presents the results of the simulations for the sensor merging and splitting control algorithm in different cases. For the simulations, we assume that the initial velocities of the robots and target are set to zero, and the obstacles in the environment are stationary. The general parameters for the simulations are listed in table 5.1.

TABLE 5.1: PARAMETER VALUES

Parameter	Definition	Value
r_0^{α}	Desired distance between robots	20
λ	Positive constant	0.42
r^{β}	Obstacle detecting range	25
r^{τ}	Target approach radius	100
k_{ip}^{t}, k_{ip}^{l}	Factors for approaching to target and leader	6; 7
k_{ip}^{1j}, k_{ip}^{2j}	Positive factors for fast connection	80; 7
k_{ip}^{1o}, k_{ip}^{2o}	Constants for fast obstacle avoidance	90; 8
k_{iv}^{j}, k_{iv}^{t}, k_{iv}^{o}, k_{iv}^{l}	Damping factors	1,8

A. Test the sensor merging control algorithm

First of all, we test the sensor merging control algorithm for a mobile sensor network of four sensors while tracking a moving target in a stationary environment, in which there are the different obstacles. For this simulation, the initial positions of the sensors and the obstacles are selected as : $p_1 = (10, 200)^T$, $p_2 = (30, 210)^{\mathrm{T}}$, $p_3 = (60, 220)^{\mathrm{T}}$, $p_4 = (400, 10)^{\mathrm{T}}$, $p_{o1} = (230, 150)^{\mathrm{T}}$, $p_{o2} = (300, 100)^{\mathrm{T}}$. The target moves along the trajectory as follows: $p_t = (0.3t + 200, \ -0.2t + 480)^T$. The

The results of the simulations in figure 5.4 and figure 5.5 show that three free sensors (S_1, S_2, S_3) have quickly found and connected each other in order to generate the robust formation of a group. In other words, these free sensors have quickly approached to a group and become the first members in the formation of this group at time t=200s (see figure 5.5). Then, these members have continued to track the moving target in the stability and robustness of a formation. Moreover, their motion towards the target position is not affected by the attractive force of the virtual leader (here sensor S_3 plays the role as a virtual leader to attract the free sensors towards it). On the other hand, under the effect of the attractive force from the virtual leader, the free sensor S_4 passes through obstacles O_1, O_2 without the collision to achieve the group that was generated by the combinations among three sensors S_1, S_2, S_3 (see figure 5.4). As shown in figure 5.4, the free sensor S_4 has easily and quickly moved towards the group that is tracking the moving target under the leadership of the virtual leader. After approaching the group, the free sensor S_4 has linked with other members in order to become a new member in this group at time t=630s while tracking the moving target. Furthermore, figure 5.5 also shows that while tracking a moving target the formation of the sensors S_1, S_2, S_3, S_4 is maintained, and the position errors among the members of this formation is very small, see figure 5.5.

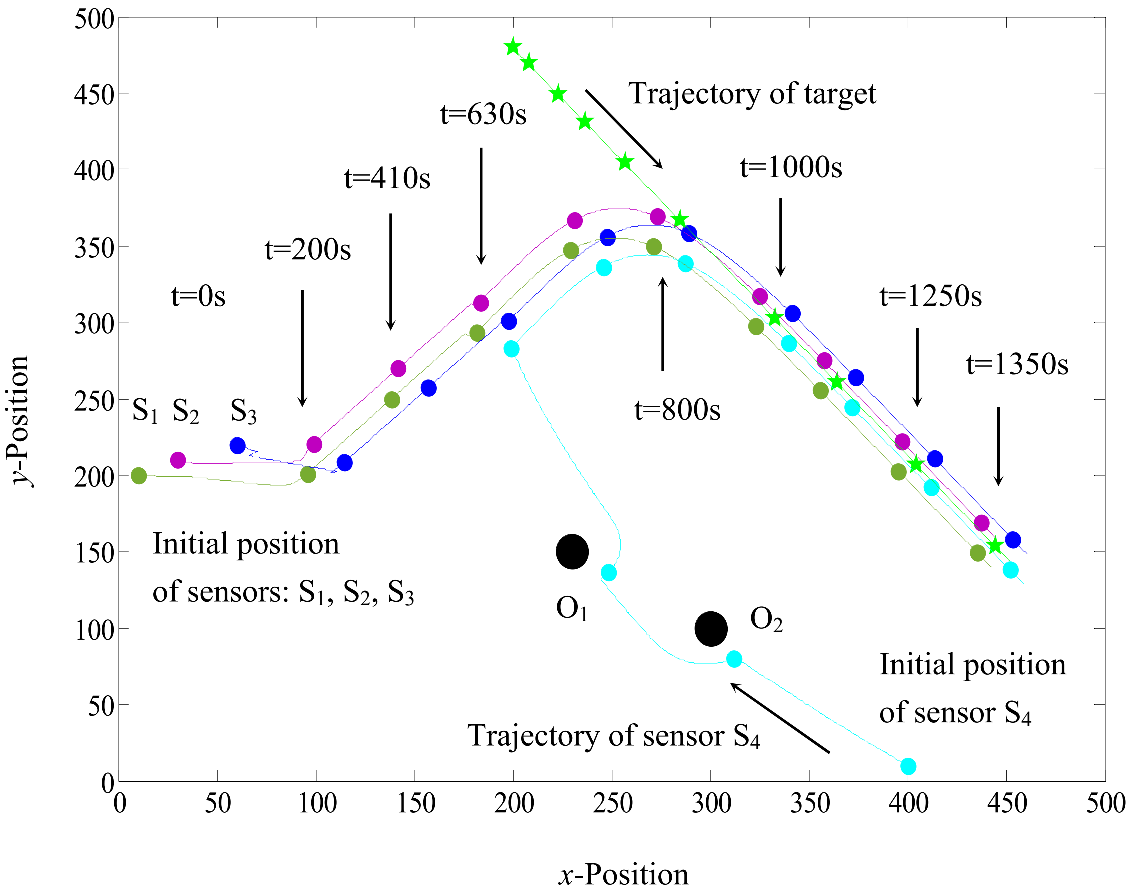

Figure 5.4: Snapshots of a mobile sensor network of four sensors that is tracking a moving target in a dynamic environment, and controlled by the formation control algorithm combined with the sensor merging control algorithm and the obstacle avoiding control algorithm. The target moves along the trajectory $p_t = (0.3t + 200,\ -0.2t + 480)^T$ (green trajectory) at time t=0s.

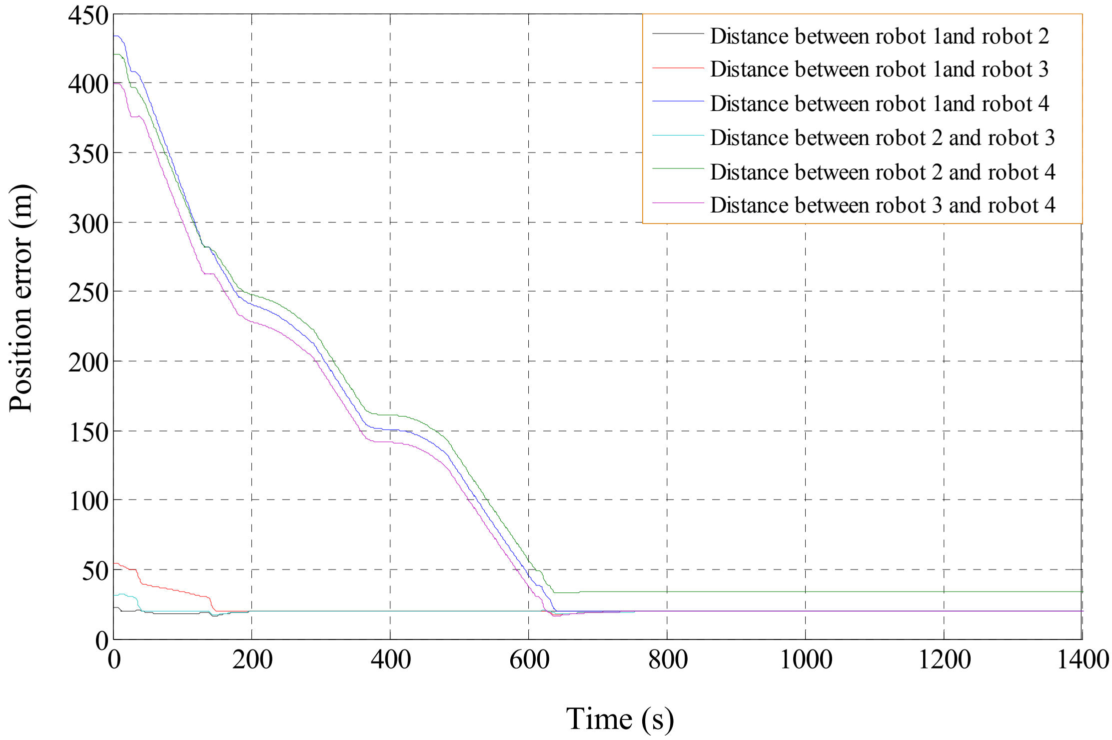

Figure 5.5: Position errors among the sensors in a mobile sensor network of four sensors at time *t* while tracking a moving target as simulated in figure 5.4.

Secondly, we test the control algorithm for a mobile sensor network of seven sensors while tracking a moving target in a stationary environment. For this simulation, the target moves along the trajectory $p_t = (0.06\text{t}+200,\ -0.08\text{t}+480)^{\text{T}}$. The initial positions of sensors and obstacles are selected as: $p_1 = (10, 20)^{\text{T}}$, $p_2 = (20, 10)^{\text{T}}$, $p_3 = (30, 20)^{\text{T}}$, $p_4 = (40, 50)^{\text{T}}$, $p_5 = (5, 80)^{\text{T}}$, $p_6 = (300, 10)^{\text{T}}$, $p_7 = (450, 10)^{\text{T}}$, $p_{o1} = (350, 100)^{\text{T}}$, $p_{o2} = (280, 200)^{\text{T}}$.

The results of the simulations in figure 5.6 show that the swarm-finding of two sensors (S_6 and S_7) is successfully achieved, and this swarm-finding does not change the general trajectory of the swarm. As shown in figure 5.6, at beginning all sensors are directed to the position of the virtual leader. The free-sensors $S_{1,2\ldots5}$ quickly find their neighbors, and immediately they connect together to become the first member-sensors of a basic group. The approach of the sensor S_6 into this basic group is not difficult. Furthermore, the sensor S_7 also successfully reaches the basic group, although it is hindered by some obstacles of the environment while tracking. The organization of the basic group is changed when the free-sensors S_6 and S_7 become the new members of this basic group. The constant distances among member-sensors are maintained while these sensors track the moving target.

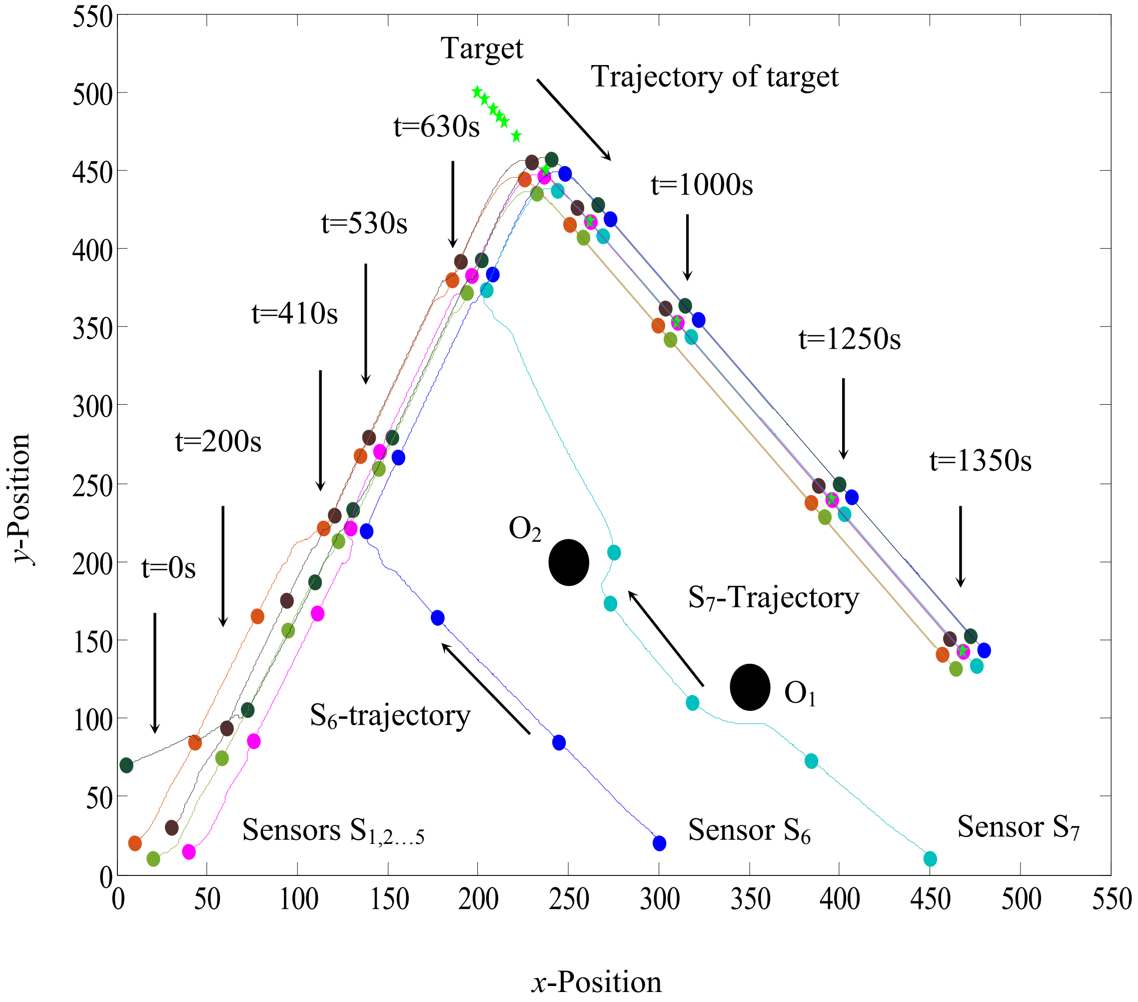

Figure 5.6: Snapshots of a mobile sensor network of seven sensors that is tracking a moving target in a stationary environment, and controlled by the formation control algorithm combined with the sensor merging control algorithm and the obstacle avoiding control algorithm. The target moves along the trajectory $p_t = (0.06t + 200,\ -0.08t + 480)^T$ (green trajectory) at time t=0s.

B. Test the sensor merging and splitting control algorithm

Firstly, we test the sensor merging/splitting algorithm in the case a new target appears while a mobile sensor network is tracking a moving old target. The old target moves in the sine wave trajectory $p_{t1} = (7\text{t}+20,\ 100\sin(0.06\text{t} - \pi/2)+350)^{\text{T}}$ at time t=0s. The new target appears at time t=25s and it moves along the trajectory $p_{t2} = (5{,}4\text{t}+148,\ -2{,}5\text{t}+269)^{\text{T}}$. The positions of obstacles are chosen as $p_{o1} = (470,\ 370)^{\text{T}}$, $p_{o2} = (510,\ 440)^{\text{T}}$, $p_{o3} = (400,\ 175)^{\text{T}}$.

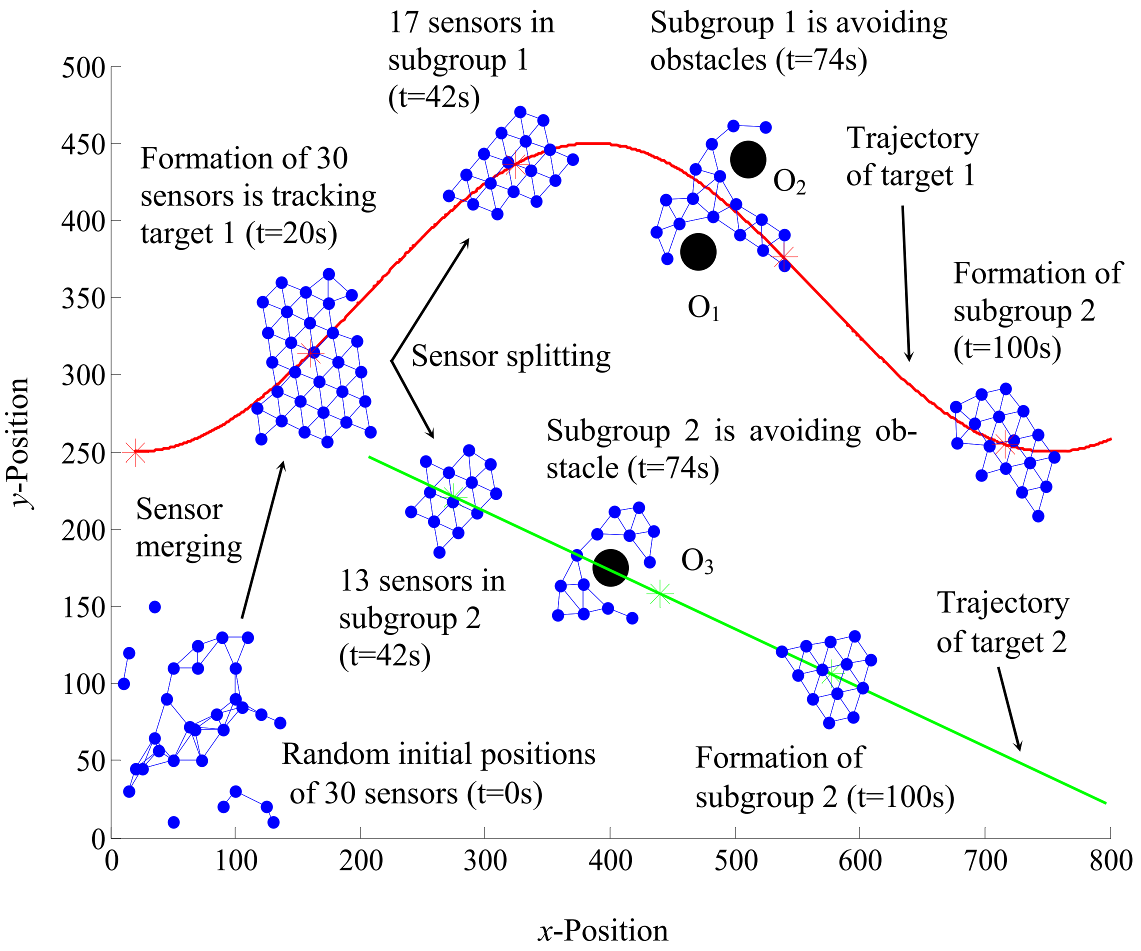

Figure 5.7: Snapshots of a mobile sensor network of 30 sensors that is tracking the moving targets in a stationary environment, controlled by the formation control algorithm combined with the sensor merging/splitting algorithm. The target 1 moves in the sine wave trajectory $p_{t1} = (7t+20,\ 100\sin(0.06t - \pi/2)+350)^T$ (red trajectory) at time t=0s, and the target 2 moves along the line trajectory $p_{t2} = (5{,}4t+148,\ -2{,}5t+269)^T$ (green trajectory) at time t=25s.

The results of the simulations in figure 5.7 show that the sensor merging/splitting control algorithm combined with the formation control algorithm is well functioning. At initial time, all sensors have the random position (see figure 5.7), but after a period of circa 20s they linked to each other in order to generate a desired formation, in which the distance between the neighboring sensors is constant. In other words, these sensors are contributed into a formation by the sensor merging control algorithm before they reach towards the old target (target 1). At time t=25s, when a new target (target 2) appears, some sensors are split from the generated main formation in order to track this new target, and the remaining sensors continue to track the old target. In figure 5.7, we can see that after sensors are split they have quickly combined to each other into a new formation. At time t=74s, the formation of subgroups is broken while avoiding obstacles. Then, these formations are automatically redesigned while tracking the moving targets at time t=100s, see figure 5.7.

As shown in figure 5.7, the sensor splitting from a main group into subgroups based on the geometry method has been successful. However, the number of the sensors in subgroups is possibly unequal (for example, in figure 5.7 there are 17 sensors in the first subgroup while in the second subgroup there are only 13 sensors). So, to solve this problem the sensor splitting algorithm is developed as follows:

Step 1. Split sensors into subgroups when a new target appears as represented above.

Step 2. Determine the number of the sensors in each subgroup. Assume that the number of the sensors in the first subgroup and the second subgroup are ξ and ζ $(\zeta+\xi=N)$, respectively. Let δ be the necessary number of the sensors to add into a subgroup in that the number of the sensors is unequal to $N/2$. If $\xi<\zeta$, $(\xi=N/2-\delta)$ and $(\zeta=N/2+\delta)$, then δ sensors of the second subgroup that have the smallest distance to the first target will split into the first subgroup. In contrast, if $\xi>\zeta$, $(\xi=N/2+\delta)$ and $(\zeta=N/2-\delta)$, then δ sensors of the first subgroup that have the smallest distance to the second target will split into the second subgroup.

The developed sensor splitting algorithm is tested in the second case: a new target appears in the interval time from t=25s to t=95s while a main group is tracking a moving old target. The results of the simulations in figure 5.8 show that the developed sensor splitting algorithm is well working, the sensors are split equally into subgroups (as shown in figure 5.8, each subgroup has 15 sensors) when the new target appears at time t=25s. In contrast, when the second target disappears at time t=95s, the second subgroup, which is tracking this target, merges into the first subgroup and continues to track the first target as the members of this subgroup. This sensor merging is successful and without the collisions. However, the structure of the formation of the whole group after merging can be redesigned. Each member sensor quickly finds its neighbors, and they connect to each other in order to generate and maintain the stable and robust connections in their formation during tracking.

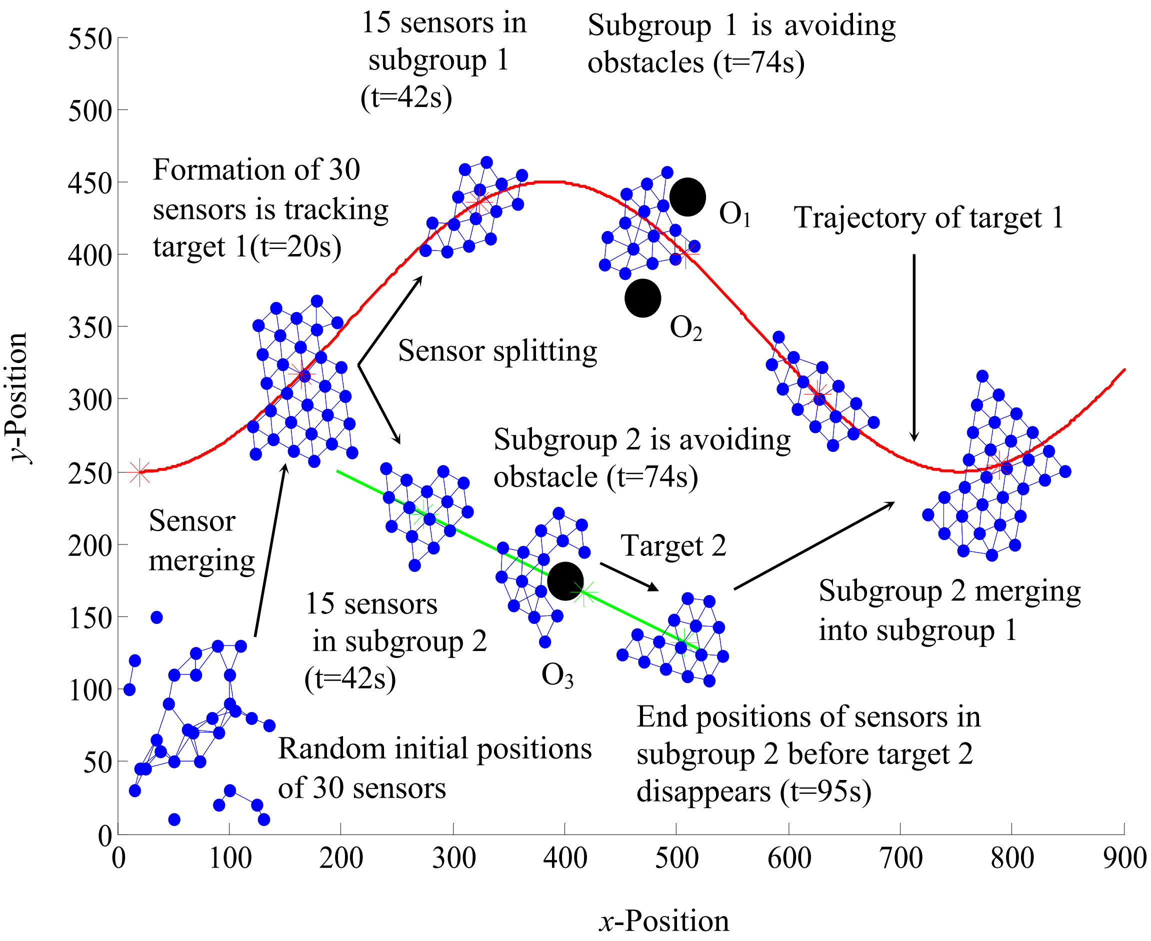

Figure 5.8: Snapshots of a mobile sensor network of 30 sensors that is tracking the moving targets in a dynamic environment, controlled by formation control algorithm combined with the sensor splitting/merging algorithm. The first target moves in the sine wave trajectory (red trajectory) at time t=0s, and the second target appears at time t=25s and runs along line trajectory (green trajectory) until it disappears at time t=95s. The sensors in a main group are split equally into two subgroups when the second target appears.

In the general case, if we want to split α sensors ($\alpha = 1,2,..,\beta-1$) from a group of β sensors to track a new target, then we perform as follows: similar to the above developed sensor splitting method, firstly we also have to determine the number ζ of the sensors in the split subgroup. Then, we compare ζ and α: if $\zeta < \alpha$, then ∂ sensors ($\zeta + \partial = \alpha$), which are close to the new target, are added into the split subgroup. On the other hand, if $\zeta > \alpha$, then ε sensors ($\zeta - \varepsilon = \alpha$), which are close to old target, are returned into the old group. The results of the simulations in figure 5.9 have verified the effect of this approach. In this situation, we can see that four sensors are split from a group of 30 sensors in order to track a new target. This target tracking is occurring with a kept formation of four sensors until this target disappears at time t=95s, and then these sensors automatically rejoin their old group (subgroup 1) at which they were split when the new target appears.

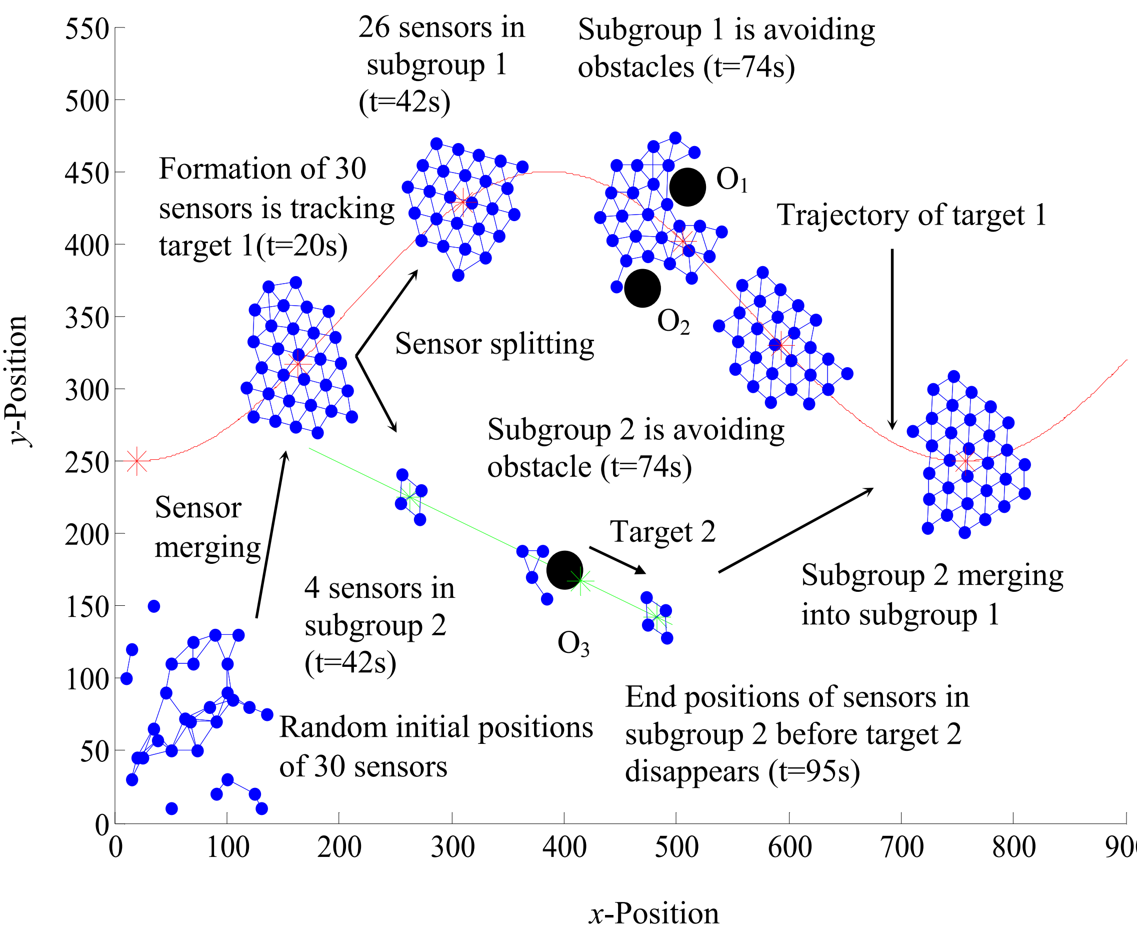

Figure 5.9: Snapshots of a subgroup of four sensors which are split from a main group of 30 sensors to track a new target in the interval time 25s< t <95s, and return to their main group after the new target disappears at time t=95s to continue to track the old target.

5.6 Conclusion

This paper has presented an approach to control a mobile sensor network to track the moving targets in a dynamic environment. In this approach, we solve two main issues: the sensor splitting and merging control when the number of the targets is changed. The free sensor merging into a group is controlled by the invariable attractive force field from the selected virtual leader in this group. In other words, under the effect of the attractive force field from the selected virtual leader, the free-sensors can easily reach the main group and become the new members in formation of this group. In contrast, subgroups will be split from the existing groups to track new targets when these new targets appear. The sensor splitting algorithm is built based on the geometry relationship between the targets and the center of the group. The members in a split subgroup will connect with their neighbors in order to generate a robust formation without collision while tracking their target. In addition, when a target disappears, the sensors that are tracking this target will automatically contribute into the nearest existing subgroup. The effectiveness of this proposed approach has been verified in simulations.

The results of the simulations have shown that this approach is one of the good control methods that can be applied to control the contribution and distribution of a mobile sensor network while tracking the dynamic targets. The sensor merging algorithm, which is built based on the energy level partitioning in the invariable attractive force field of the virtual leader, works very well. Using this control algorithm, the free-sensors can easily find and approach their group while reaching the target, and then these free-sensors will become the new member-sensors in the formation of the group. Especially, this method solves the speed problem that occurs when the free-sensors are away from their target. Furthermore, the swarm-finding of the free sensors does not influence on the target tracking of the swarm. On the other hand, the sensor splitting algorithm based on the geometry method shows that this is an active method for the sensor distribution from a main group into subgroups to track the new targets. The sensor partitioning is quickly performed at the time when a new target appears, and then the formation of these split sensors is maintained while tracking their target in a free environment until this target disappears. However, while avoiding the obstacles the formation connection of the groups are broken. In this situation, the member-sensors of these groups are distributed to the free sensors in order to easily escape the obstacles without the collision among them. After exiting the obstacles, the formation of the distributed sensors is reorganized to continue tracking their target. Moreover, the results of the simulations in figure 5.9 have also shown that the development and application of the proposed approach for the exact sensor splitting from a main group into subgroups, such as the splitting α sensors from a group of β sensors, ($\alpha = 1, 2 \ldots \beta - 1$, and $\beta \leq N$) into a subgroup, are successful.

6 Conclusion and Future Work

6.1 Conclusion

This work has presented a method for controlling the formation of autonomous robots while tracking the dynamic targets under the influence of the dynamic environment. This approach is based on the developed artificial vector fields. In this approach, the proposed artificial vector fields, which consist of the attractive, repulsive, and rotational force field, are combined with the damping term in the formation control laws in order to control the velocity, heading, connectivity, as well as the obstacle avoidance for a swarm of autonomous robots during movement. In addition, using the added rotational force field in the obstacle avoiding controller, robots can easily escape the obstacles while moving towards the target. In the preceding chapters, focuses on two main issues of formation control following the desired formations and cooperative formation control have been given.

In the first approach (formation control following the desired formations), a desired formation (for example V-shape, line or circular shape), which includes the coordinated equidistant virtual nodes, is first generated based on the relative position between the swarm's virtual leader and the moving target. The virtual leader of the swarm is randomly chosen from a member that is closest to the target. This virtual leader plays an important role in creating and driving the formation towards the target in a stable trajectory. Hence, in undesired cases, for example, when the leader is broken or trapped in the obstacles (U-shape obstacle), a new leader is immediately chosen, so that the swarm will continue to track the target. The selected virtual leader is controlled by a global attractive potential field generated from the target in order to drive its formation on a specific trajectory towards the target. Furthermore, the motion of each robot is controlled by the artificial force fields, which include the local and global attractive potential fields surrounding the generated virtual nodes in the desired formation and the local repulsive potential field surrounding each robot. Hence, the robot always converges to a certain virtual node in the desired formation and avoids collisions with other robots simultaneously. In addition, using the repulsive vector field combined with the rotational vector field in the obstacle avoiding controller, robots can easily escape the obstacles while tracking. Using the desired position finding algorithm, robots can quickly find their desired position at the virtual nodes in the desired formation. Moreover, the influence of the noisy environment on the stability of the formation, as well as the formation adaptation of a swarm while reaching a moving target are also studied in this topic. The results of the simulations have proved the success of the approach. Results show that the robots quickly approached the coordinated virtual nodes in the desired formation and maintained at these virtual nodes, although affected by a noisy environment. Especially, upon the use of the desired position finding algorithm, the formation of a swarm was not discrete. On overcoming the obstacle, the robots continued to approach their formation and then they found the other desired positions in the formation.

In the second approach (cooperative formation control), robots are only allowed to communicate within their neighboring relationship; however, the swarm's cohesion must be maintained coincidentally in tracking the moving target and as well as while avoiding obstacle. Therefore, in order to perform the cooperative formation control for a swarm, the neighboring robots are first linked to each other by the proposed formation connection controller. These neighboring connections created the robust formation cohesion and avoided collisions among robots in the swarm simultaneously. Further, for the formation adaptation while passing through a narrow space among obstacles, the adaptive formation control algorithm is built based on the size change of formation. Moreover, the splitting/merging control algorithm is used to help the free robots to easily and quickly find their formation as well as to split some roots from a main group into smaller subgroups when some new targets appear. The simulations have shown that our proposed algorithms for controlling the cooperative formation have well worked, namely: the free robots have easily merged into the formation of their swarm; the formation connectivity was maintained while tracking; the formation's size was shrunk into a smaller size while overcoming the narrow space among obstacles to maintain formation.

Finally, our proposed control algorithms are not only to control autonomous robots moving along a desired trajectory, but also to hold these robots in a specified formation without collisions during movement. Especially, using the added rotational force field in the obstacle avoiding controller, the local minima problems that still exist in the traditional potential field method have been solved.

6.2 Future work

Further studies and researches can be expanded to cooperative control in a hybrid system, cooperative learning and sensing. The first research objective is to develop a method for formation control of autonomous robots in three-dimensional space (3-D) to model swarms and allow heterogeneous swarms of aerial, ground, and underwater vehicles to combine formation together in a certain task while avoiding collisions; for example, the cooperation of the unmanned aerial vehicles with autonomous mobile robots while pursuing the moving objectives. In addition, learning and sensing can also be developed and utilized in the environment where the connectivity, the size, as well as the structure of a swarm can be improved by environmental parameters. In a multi robot network, members can learn to avoid the moving obstacles, while maintaining network connectivity and topology, aside to find a desired configuration of network. Moreover, members in a network can also sense the environmental parameters in order to perform the task of the environment estimation and mapping.

7 Appendix

7.1 Vector field

According to [96, 97], a domain in which each point is attached a vector is called a vector field. A more precise mathematical definition of a vector field is summarized below.

Definition 7.1. Let *D* be a domain in $\mathbb{R}^n$ (n-dimensional space with n=2 or 3). A vector field on *D* is a function F that assigns to each point $(\xi_1,\ \xi_2,..\xi_n)$ in *D* an n-dimensional vector $\mathrm{F}(\xi_1,\ \xi_2,..\xi_n)$. That is

$$\begin{aligned}\mathrm{F}(\xi_1,\ \xi_2,..\xi_n) &= \left(f_1(\xi_1,\ \xi_2,..\xi_n), f_2(\xi_1,\ \xi_2,..\xi_n),..f_n(\xi_1,\ \xi_2,..\xi_n)\right)^T \\ &= f_1(\xi_1,\ \xi_2,..\xi_n)e_1 + f_2(\xi_1,\ \xi_2,..\xi_n)e_2 + ...f_n(\xi_1,\ \xi_2,..\xi_n)e_n.\end{aligned} \tag{7.1}$$

Where $f_1(\xi_1,\ \xi_2,..\xi_n), f_2(\xi_1,\ \xi_2,..\xi_n),..f_n(\xi_1,\ \xi_2,..\xi_n)$ are real-valued functions, and they are called the component functions. $e_1,\ e_2,..e_n$ are the unit vectors.

7.2 Curl of a vector field

Definition 7.2. Consider a vector field $\mathrm{F} = (\mathrm{P},\ \mathrm{Q},\ \mathrm{R})^T$ in three dimensional space. The curl of this vector field at a point $(x,\ y,\ z)^T$ is the vector field defined by

$$\begin{aligned}\text{curl F} &= \nabla\times\mathrm{F} \\ &= \left(\frac{\partial}{\partial x},\ \frac{\partial}{\partial y},\ \frac{\partial}{\partial z}\right)^{\mathrm{T}}\times(\mathrm{P},\ \mathrm{Q},\ \mathrm{R})^{\mathrm{T}} \\ &= \begin{vmatrix} e_x & e_y & e_z \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} \\ \mathrm{P} & \mathrm{Q} & \mathrm{R}\end{vmatrix} \\ &= \left(\frac{\partial \mathrm{R}}{\partial y}-\frac{\partial \mathrm{Q}}{\partial z}\right)e_x + \left(\frac{\partial \mathrm{P}}{\partial z}-\frac{\partial \mathrm{R}}{\partial x}\right)e_y + \left(\frac{\partial \mathrm{Q}}{\partial x}-\frac{\partial \mathrm{P}}{\partial y}\right)e_z \\ &= \left(\frac{\partial \mathrm{R}}{\partial y}-\frac{\partial \mathrm{Q}}{\partial z},\ \frac{\partial \mathrm{P}}{\partial z}-\frac{\partial \mathrm{R}}{\partial x},\ \frac{\partial \mathrm{Q}}{\partial x}-\frac{\partial \mathrm{P}}{\partial y}\right)^{\mathrm{T}}\end{aligned} \tag{7.2}$$

Where e_x, e_y, e_z are the unit vectors in the direction of the *x*, *y*, and *z* axes, respectively. The vector differential operator ∇ (read nabla or del) is described as follows

$$\nabla = \frac{\partial}{\partial x}e_x + \frac{\partial}{\partial y}e_y + \frac{\partial}{\partial z}e_z = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^{\mathrm{T}}. \tag{7.3}$$

Similarly, we can also take curls of the plane vector fields (the vector fields in two dimensional space $\mathrm{F} = (\mathrm{P},\ \mathrm{Q})^T$). Just assume that the first two component functions P and Q are not dependent on the z and the component function R is zero simultaneously. Then curl of plane vector fields is rewritten as follows

$$\begin{aligned} \operatorname{curl}\mathrm{F} &= \nabla \times \mathrm{F} \\ &= \left(0,\ 0,\ \frac{\partial \mathrm{Q}}{\partial x} - \frac{\partial \mathrm{P}}{\partial y}\right)^{\mathrm{T}} = \left(\frac{\partial \mathrm{Q}}{\partial x} - \frac{\partial \mathrm{P}}{\partial y}\right)e_z. \end{aligned} \tag{7.4}$$

Definition 7.3. The vector field F is called irrotational if its curl is zero ($\operatorname{curl}\mathrm{F} = (0, 0, 0)^T$).

Remark 7.1. The irrotational vector field $\mathrm{F} = (\mathrm{P},\ \mathrm{Q},\ \mathrm{R})^T$ has to satisfy the following conditions in the partial derivatives of P, Q and R:

$$\frac{\partial \mathrm{R}}{\partial y} = \frac{\partial \mathrm{Q}}{\partial z},\ \frac{\partial \mathrm{P}}{\partial z} = \frac{\partial \mathrm{R}}{\partial x} \text{ and } \frac{\partial \mathrm{Q}}{\partial x} = \frac{\partial \mathrm{P}}{\partial y} \tag{7.5}$$

Example: Consider the vector field as proposed in (2.21)

$$\begin{aligned} f_r^{or}(p_r) &= w_r^{or} c_r^o n_r^{or} \\ &= w_r^{or} c_r^o \left(\frac{(y_r - y_o)}{\rho(p_r, p_o)},\ \frac{-(x_r - x_o)}{\rho(p_r, p_o)} \right)^{\mathrm{T}} \\ &= \left(\frac{w_r^{or} c_r^o (y_r - y_o)}{\sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}},\ \frac{-w_r^{or} c_r^o (x_r - x_o)}{\sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}} \right)^{\mathrm{T}} \\ &= \left(P(x, y),\ \mathrm{Q}(x, y) \right)^{\mathrm{T}}. \end{aligned} \tag{7.6}$$

According definition 7.2, the curl of this vector field is calculated as follows

$$
\begin{aligned}
\operatorname{curl} f_r^{or}(p_r) &= 0e_x + 0e_y + \left(\frac{\partial Q}{\partial x_r} - \frac{\partial P}{\partial y_r}\right)e_z \\
&= \left(\frac{\partial}{\partial x_r}\frac{(-w_r^{or}c_r^o(x_r - x_o))}{\sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}} - \frac{\partial}{\partial y_r}\frac{w_r^{or}c_r^o(y_r - y_o)}{\sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}}\right)e_z \\
&= \left(\frac{-w_r^{or}c_r^o(y_r - y_o)^2}{\left(\sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}\right)^3} + \frac{-w_r^{or}c_r^o(x_r - x_o)^2}{\left(\sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}\right)^3}\right)e_z \qquad (77) \\
&= \left(\frac{-w_r^{or}c_r^o}{\sqrt{(x_r - x_o)^2 + (y_r - y_o)^2}}\right)e_z.
\end{aligned}
$$

Equation (7.7) shows that $\operatorname{curl} f_r^{or}(p_r) \neq 0 \ \forall (x_r, y_r) \neq (x_o, y_o)$, hence $f_r^{or}(p_r)$ is a rotational vector field.

7.3 Gradient vector field

Definition 7.4. If the vector field F is defined differentiable everywhere and $\operatorname{curl} \mathrm{F} = (0, 0, 0)^T$, then F is a gradient vector field.

Remark 7.2. A gradient vector field F has some corresponding potential function *f* such that

$$
\mathrm{F} = \nabla f = \begin{cases} \left(\dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y}\right)^{\mathrm{T}} & \text{if } f = f(x, y) \\ \left(\dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y}, \dfrac{\partial f}{\partial z}\right)^{\mathrm{T}} & \text{if } f = f(x, y, z). \end{cases} \qquad (7.8)
$$

In other words, the gradient of a differentiable scalar field $f \subset \mathbb{R}^n$ is a vector field. This vector field satisfies equation (7.8).

Example: Consider the scalar function $V_i^j(p_i)$ as proposed in (4.6). Taking the negative gradient of this scalar function at p_i we obtain

$$
\begin{aligned}
-\nabla V_i^j(p_i) &= -\nabla\left(\frac{c_i^j}{2}\left(k_{ip}^{1j}\left(\frac{1}{d_i^j}-\frac{1}{r_0^\alpha}\right)^2 + k_{ip}^{2j}\left(d_i^j - r_0^\alpha\right)^2\right)\right) \\
&= -c_i^j k_{ip}^{1j}\left(\frac{1}{d_i^j}-\frac{1}{r_0^\alpha}\right)\nabla\left(\frac{1}{d_i^j}-\frac{1}{r_0^\alpha}\right) - c_i^j k_{ip}^{2j}\left(d_i^j - r_0^\alpha\right)\nabla\left(d_i^j - r_0^\alpha\right) \\
&= c_i^j k_{ip}^{1j}\left(\frac{1}{d_i^j}-\frac{1}{r_0^\alpha}\right)\frac{1}{(d_i^j)^2}\nabla d_i^j - c_i^j k_{ip}^{2j}\left(d_i^j - r_0^\alpha\right)\nabla d_i^j \\
&= \left(c_i^j k_{ip}^{1j}\left(\frac{1}{d_i^j}-\frac{1}{r_0^\alpha}\right)\frac{1}{(d_i^j)^2} - c_i^j k_{ip}^{2j}\left(d_i^j - r_0^\alpha\right)\right)\nabla d_i^j
\end{aligned}
\tag{7.9}
$$

Where the distance between robots i and j is calculated as $d_i^j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Hence, we have

$$
\begin{aligned}
\nabla d_i^j &= \nabla\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\
&= \left(\frac{\partial}{\partial x_i}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},\ \frac{\partial}{\partial y_i}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right)^{\mathrm{T}} \\
&= \left(\frac{(x_i - x_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}},\ \frac{(y_i - y_j)}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}\right)^{\mathrm{T}} \\
&= \frac{(p_i - p_j)}{\left\| p_i - p_j \right\|}.
\end{aligned}
\tag{7.10}
$$

Substitute (7.10) into (7.9), we obtain

$$
\begin{aligned}
f_i^j(p_i) &= -\nabla V_i^j(p_i) \\
&= c_i^j\left(\left(\frac{1}{d_i^j}-\frac{1}{r_0^\alpha}\right)\frac{k_{ip}^{1j}}{(d_i^j)^2} - k_{ip}^{2j}\left(d_i^j - r_0^\alpha\right)\right)\frac{(p_i - p_j)}{\left\| p_i - p_j \right\|}.
\end{aligned}
\tag{7.11}
$$

Equation (7.11) shows that the scalar function $V_i^j(p_i)$ as proposed in (4.6) is the potential function of the force field $f_i^j(p_i)$ as given in (4.7).

7.4 Proof of gravitational force field

In this subsection, we first prove that the gravitational force field (2.1) is a gradient vector field. In order to simplify the analysis, we assume that: the point mass m_2 located at the origin attracts the point mass m_1 located at $p_1 = (x,\ y,\ z)^{\mathrm{T}} \in \mathbb{R}^3/(0,\ 0,\ 0)^{\mathrm{T}}$ with a force of magnitude $F_{12} = Gm_1m_2r/\|r\|^2$. Therefore, equation (2.1) can be rewrite as follows

$$
\begin{aligned}
f_{12} &= \frac{-Gm_1m_2}{\left(\sqrt{x^2+y^2+z^2}\right)^3}(x,\ y,\ z)^{\mathrm{T}} \\
&= \left(\frac{-Gm_1m_2x}{\left(\sqrt{x^2+y^2+z^2}\right)^3},\ \frac{-Gm_1m_2y}{\left(\sqrt{x^2+y^2+z^2}\right)^3},\ \frac{-Gm_1m_2z}{\left(\sqrt{x^2+y^2+z^2}\right)^3}\right)^{\mathrm{T}} \\
&= \left(P(x,\ y,\ z),\ Q(x,\ y,\ z),\ R(x,\ y,\ z)\right)^{\mathrm{T}}.
\end{aligned}
\tag{7.12}
$$

According to definition 7.2, the curl of this force field at $p_1 = (x,\ y,\ z)^{\mathrm{T}}$ is calculated as

$$
\begin{aligned}
\operatorname{curl} f_{12} &= \left(\frac{\partial \mathrm{R}}{\partial y} - \frac{\partial \mathrm{Q}}{\partial z},\ \frac{\partial \mathrm{P}}{\partial z} - \frac{\partial \mathrm{R}}{\partial x},\ \frac{\partial \mathrm{Q}}{\partial x} - \frac{\partial \mathrm{P}}{\partial y}\right)^{\mathrm{T}} \\
&= (0,\ 0,\ 0\ )^{\mathrm{T}}.
\end{aligned}
\tag{7.13}
$$

Equation (7.13) shows that the force field (7.12) is a gradient vector field.

Now, in order to prove that the scalar function $V_{12} = -Gm_1m_2/\|r\|$ is the potential function of the force field f_{12}, we take the negative gradient of V_{12} as follows

$$
\begin{aligned}
-\nabla V_{12} &= \left(\frac{\partial}{\partial x}\frac{Gm_1m_2}{\sqrt{x^2+y^2+z^2}},\ \frac{\partial}{\partial y}\frac{Gm_1m_2}{\sqrt{x^2+y^2+z^2}},\ \frac{\partial}{\partial z}\frac{Gm_1m_2}{\sqrt{x^2+y^2+z^2}}\right)^{\mathrm{T}} \\
&= \left(\frac{-Gm_1m_2x}{\left(\sqrt{x^2+y^2+z^2}\right)^3},\ \frac{-Gm_1m_2y}{\left(\sqrt{x^2+y^2+z^2}\right)^3},\ \frac{-Gm_1m_2z}{\left(\sqrt{x^2+y^2+z^2}\right)^3}\right)^{\mathrm{T}} \\
&= \frac{-Gm_1m_2}{\left(\sqrt{x^2+y^2+z^2}\right)^3}(x,\ y,\ z)^{\mathrm{T}} = f_{12}.
\end{aligned}
\tag{7.14}
$$

Equation (7.14) shows that the scalar function $V_{12} = -Gm_1m_2/\|r\|$ is the potential function of the force field $f_{12} = -Gm_1m_2r/\|r\|^3$.

8 Publications

[1] Anh Duc Dang, Hung Manh La and Joachim Horn, “Distributed formation control for autonomous robots following desired shape in noisy environment”, *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016), Badeb-Baden, Germany, pages 285-290, September 19-21, 2016.*

[2] Anh Duc Dang, Hung Manh La and Joachim Horn, “Influence of formation angle on stability of V-formation while tracking a moving target”, *Journal of Automation and Control Engineering, 2016, accepted.*

[3] Anh Duc Dang and Joachim Horn, “Stability of V-formation while tracking a moving target”, *Proceedings of the Automatics- 2015 XXII International Conference on Automatic Control, Odessa, Ukraine, pages 154-156, September 10-11, 2015.*

[4] Anh Duc Dang and Joachim Horn, “Formation control of autonomous robots following desired formation during tracking a moving target”, *Proceedings of the 2nd IEEE International Conference on Cybernetics, Gdynia, Poland, pages 160-165, June 24-26, 2015.*

[5] Anh Duc Dang and Joachim Horn, “A mobile sensor network tracking moving targets in a dynamic environment”, *Proceedings of the 3rd IFAC Workshop on Multi-vehicle systems (MVS 2015), Genova, Italy, pages 1-6, May 18, 2015.*

[6] Anh Duc Dang and Joachim Horn, “Collinear formation control of autonomous robots to move towards a target using artificial force fields”, *Proceedings of the 2015 IEEE International Conference on Technologies for Practical Robot Applications, Boston, USA, pages 1-6, May 11-12, 2015.*

[7] Anh Duc Dang and Joachim Horn, “Formation adaptation control of autonomous robots in a dynamic environment”, *Proceedings of the 2015 IEEE International Conference on Industrial Technology, Seville, Spain, pages 3190-3195, March 17-19, 2015.*

[8] Anh Duc Dang and Joachim Horn, “Formation control of leader-following UAVs to track a moving target in dynamic environment”, *Journal of Automation and Control Engineering, Vol.3, No.1, pages 1-8, February 2015.*

[9] Anh Duc Dang and Joachim Horn, “Formation control of autonomous robots to track a Moving target in an unknown Environment”, *Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics, Bali, indonesia, pages 896-901, December 5-10, 2014.*

[10] Anh Duc Dang and Joachim Horn, “Intelligent Swarm-Finding in Formation Control of Multi-Robots to Track a Moving Target”, *World Academy of Science, Engineering and Technology, International Journal of Computer, Information Science and Engineering, Vol.8, No.4, pages 12-18, 2014.*

[11] Anh Duc Dang and Joachim Horn, “Path planning for a formation of autonomous robots in an unknown environment using artificial force fields”, *Proceedings of the 18th IEEE International Conference on System Theory, Control and Computing, Sinaia, Romania, pages 773-778, October 17-19, 2014.*

9 References

[1] C. Reynolds, “Flocks, birds, and schools: A distributed behavioral model,” *Computer Graphics,ACM SIGGRAPH ’87 Conference Proceedings, Anaheim, California*, vol. 21, no. 4, pp. 25–34, 1987.

[2] S. G. Reebs, “Can a minority of informed leaders determine the foraging movements of a fish shoal?,” *Animal Behavior*, vol. 59, pp. 403–409, 2000.

[3] A. Avitabile, R. A. Morse, and R. Boch, “Swarming honey bees guded by pheromones”, Annuals of the Entomological Society of America, 68(6):1079-1082, 1975.

[4] Lisa Lofland Gould and Frank Heppner, “The vee formation of canada geese”, *the Auk 91, pages 494–506, July 1974.*

[5] Timothy C. Williams, Thomas J. Klonoski and Philip Berkeley, “Angle of canada goose v flight formation measured by radar”, *the Auk 93, pages 554–559, July 1976.*

[6] Pete Seiler, aniruddha Pant, and Karl Hedrick, “Analysis of bird formations”, *Proceedings of the 41st IEEE Conference on Decision and Control, pages 118–123, December 2002.*

[7] John P. Badgerow, “An analysis of function in the formation flight of Canada geese”, *the Auk 105, pages 749–755, October 1988.*

[8] A. Mogilner, L. Edelstein-Keshet, L. Bent, and A. Spiros, “Mutual interactions, potentials, and individual distance in a social aggregation,” *J. Math. Biol*, vol. 47, pp. 353–389, 2003.

[9] S. Pang and J. A. Farrell, “Chemical plume source localization,” *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 36, no. 5, pp. 1068–1080, 2006.

[10] J. A. Farrell, S. Pang, and W. Li, “Chemical plume tracing via an autonomous underwater vehicle,” *IEEE J. Oceanic Eng.*, vol. 30, no. 2, pp. 428–442, Apr. 2005.

[11] Simon Garnier, Jacques Gautrais, Guy Theraulaz,“The biological principles of swarm intelligence’’, *Springer, Swarm Intell (2007) 1, pp: 3–31.*

[12] Aleksandar Jevtic, Diego Andina, ‘Swarm Intelligence and Its Applications in Swarm Robotics’’, *6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, Tenerife, Spain, pp. 41-46, December 2007.*

[13] E. Bonabeau, M. Dorigo, and G, “Theraulaz. Swarm Intelligence: From Natural to Artifcial Systems” , Oxford Univ. Press, NY, 1999.

[14] Shi and Yiwen Zhao, “An efficient path planning algorithm for mobile robot using improved potential field”, *Proc. of the 2009 IEEE Intl. Conf. on Robotics and Biomimetics, pp. 1704-1708, December 2009.*

[15] Simon János and István Matijevics, “Implementation of potential field method for mobile robot navigation in greenhouse environment with WSN support”, *IEEE 8th Intl. Symp. on Intelligent Systems and Informatics, pp. 319-323, September 2010.*

[16] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots ”, *International Journal of Robotics Research, Vol.o, No.1, pp. 90-98, 1986.*

[17] Kim, J. O and Khosla, P. K, “Real-time obstacle avoidance for manipulators and mobile robotusing harmonic potentiel functions”, *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 338-349, 1992.

[18] L. Huang, “Velocity planning for a mobile robot to track amoving target – a potential field approach,” *Robotics and Autonomous Systems*, pp.1–9, 2008.

[19] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for a mobile robot navigation," *Proceedings of the 46th IEEE Conference on Robotics and Automation*, pp. 1398–1404, 1991.

[20] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning", *IEEE Trans. on Robotics and Automation, vol. 16, no.5, pp. 615-620, October 2000.*

[21] Ding Fu-guang, Jiao Peng, Bian Xin-qian, Wang Hong-jian "AUV local path planning based on virtual potential field", *Proc. of the IEEE Intl. Conf. on Mechatronics and Automation Niagara Falls, pp. 1711-1716, July 2005.*

[22] Qing Li, Lijun Wang, Bo Chen, and Zhou Zhou, "An Improved Artificial Potential Field Method for Solving Local Minimum Problem". *IEEE the 2th Intl. Conf. on Intelligent control and Information Processing, pp. 420-424, 2011.*

[23] Frank E. Schneider and Dennis Wildermuth, "A potential field based approach to multi robot formation navigation", *Proc. of the 2003 IEEE Intl. Conf. on Robotics, Intelligent System and Signal Processing, pp. 680-685, October 2003.*

[24] Anh Duc Dang and Joachim Horn, "Intelligent swarm-finding in formation control of multi-robots to track a moving target", *Intl. Journal of Computer, Information Science and Engineering, Vol.8, No. 4, pp.12-18, 2014.*

[25] Min Gyu Park and Min Cheol Lee "Artificial potential field based path planning for mobile robots using a virtual obstacle concept", *Proc. of the 2003 IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics, pp. 735-740, 2003.*

[26] Feilong Li, Yan Tan, Yujun Wang and Gengyu Ge "Mobile robots path planning based on evolutionary artificial potential fields approach", *Proc. of the the 2th Intl. Conf. on Computer Science and Electronics Engineering, pp. 1314-1317, 2013.*

[27] Anh Duc Dang and Joachim Horn, "Formation control of autonomous robots to track a Moving target in an unknown Environment", *Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics, Bali, indonesia, pp. 896-901, December 5-10, 2014.*

[28] Anh Duc Dang and Joachim Horn, "Path planning for a formation of autonomous robots in an unknown environment using artificial force fields", *Proceedings of the 18th IEEE International Conference on System Theory, Control and Computing, Sinaia, Romania, pages 773-778, October 17-19, 2014.*

[29] A. Abdessameud and A. Tayebi, "Formation control of VTOL-UAVs", *Joint 48th IEEE Conf. on Decision and Control and 28th Chinese Control Conference , pp. 3454-3459,December 2009.*

[30] Luis García- Delgado, Alejandro Dzul, Víctor Santibánez and Miguel Llama, "Quadrotor formation control as an interconnected system", *Preprints of the 18th IFAC World Congress, pp. 13582-13587, August 2011.*

[31] M. Alighanbari and J. P. How, "Decentralized task assignment for unmanned aerial vehicles," *Proceedings of the 44nd IEEE Conference on Decision and Control and European Control Conference*, pp. 5668–5673, 2005.

[32] Kishorekumar H Kowdiki, Ranjit Kumar Barai and Samar Bhattacharya, "Leader-follower formation control using artificial potential functions: A kinematic approach", *IEEE Intl. Conf. on Advances in Engineering, Science and Management, pp. 500-505, March 2012.*

[33] Naomi Ehrich Leonard and Edward Fiorelli, "Virtual leader, artificial potential and coordinated control of groups", *Proc. of the 40th IEEE Conf. on Decision and Control, pp. 2968-2973, December 2001.*

[34] H. Su, X. Wang, and Z. Lin, "Flocking of multi–agents with a virtual leader," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 293–307, 2009.

[35] H. Su, X. Wang, and Z. Lin, "Flocking of multi–agents with a virtual leader, part I: with a minority of informed agents," *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 2937–2942, 2007.

[36] H. Su, X. Wang, and Z. Lin, "Flocking of multi–agents with a virtual leader, part II: with a virtual leader of varying velocity," *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 1429–1434, 2007.

[37] H. Shi, L. Wang, T. Chu, F. Fu, and M. Xu, "Flocking of multi–agents with a virtual leader," *Proceedings of the 2007 IEEE Symposium on Artificial Life*, pp. 287–29, 2007.

[38] J. Jennings, G.Whelan, andW. F. Evans, "Cooperative search and rescue with a team of mobile robots," *Proceedings of the 8th International Conference on Advanced robotics, Monterrey, CA, USA*, 1997.

[39] H. M. La and W. Sheng, "Flocking control of a mobile sensor network to track and observe a moving target," *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (*ICRA*), Kobe, Japan*, 2009.

[40] A. Regmi, R. Sandoval, R. Byrne, H. Taner, and C. T. Abdallah, "Experimental implementation of flocking algorithms in wheeled mobile robots," *American Control Conference*, pp. 4917–4922, 2005.

[41] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory", *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[42] H. G. Tanner, A. Jadbabai, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, pp. 863–868, May 2007.

[43] Michael M. Zavlanos and George J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 812–816, August 2007.

[44] K. M. Lynch, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transaction on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[45] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[46] M. Tubaishat and S. Madria, "Sensor networks: An overview," *IEEE Potentials*, pp. 20–23, May 2003.

[47] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A surve of sensor networks," *IEEE Communications Magazine*, pp. 102–114, 2002.

[48] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 24–33, 2004.

[49] Dong Eui Chang, Shawn C. Shadden, Jerrold E. Marsden and Reza Olfati-Saber, " Collision avoidance for multiple agent systems", *Proc. of the 42nd IEEE Conf. on Decision and Control, pp. 539-543, December 2003.*

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

[51] H. M. La and W. Sheng, "Distributed sensor fusion for scalar field mapping using mobile sensor networks", *IEEE Transactions on Cybernetics*, pp. 766-778, Vol. 43, No. 2, April, 2013.

[52] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, "Distributed control applications within sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1235–1246, 2003.

[53] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," *ISA transactions*, vol. 46, pp. 3–13, 2007.

[54] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, 2009.

[55] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.

[56] T. H. Chung, J. W. Burdick, and R. M. Murray, "Decentralized motion control of mobile sensing agents in a network," *Proceedings of the 44th IEEE Conference on Decision andControl*, 2005.

[57] J. Cortes, S.Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[58] P. Yang, R. A. Freeman, and K. M. Lynch, "Distributed cooperative active sensing using consensus filters," *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pp. 405–410, 2007.

[59] K. M. Lynch, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transaction on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[60] H. M. La, R. Lim and W. Sheng, "Multi-robot cooperative learning for predator avoidance", *IEEE Transactions on Control Systems Technology*, *April 2014.*

[61] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[62] Z. Lin, M. E. Broucke, and B. A. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

[63] B. Jung and G. S. Sukhatme, "Cooperative multi–robot target tracking," *The 8^{th} International Symposium on Distributed Autonomous Robotic Systems*, pp. 81–90, 2006.

[64] S. Kamath, E. Meisner, and V. Isler. "Triangulation based multi target tracking with mobile sensor networks". *IEEE Intl. Conf. on Robotics and Automation, pages 3283–3288, 2007.*

[65] B. Jung and G. S. Sukhatme, "Tracking targets using multiple robots: The effect of environment occlusion," *Autonomous Robots Journal*, vol. 13, no. 3, pp. 191–205, 2002.

[66] A. Kolling and S. Carpin, "Cooperative observation of multiple moving targets: an algorithm and its formalization," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 935–953, 2007.

[67] J. Spletzer and C. Taylor, "Dynamic sensor planning and control for desiredly tracking targets," *International Journal of Robotics Research*, 2003.

[68] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," *Proc of IEEE International Conference on Robotics and Automation,* ICRA2001, vol. 2, pp. 1665–1670, 2001.

[69] Anh Duc Dang and Joachim Horn, "A mobile sensor network tracking moving targets in a dynamic environment", *Proceedings of the 3rd IFAC Workshop on Multivehicle systems MVS 2015, Genova, Italy, pages 1-6, May 18, 2015.*

[70] H. M. La and W. Sheng, "Dynamic targets tracking and observing in a mobile sensor network", *the Elsevier journal on Robotics and Autonomous Systems*, pp. 996–1009, Vol. 60, Issue 7, July 2012.

[71] G. Lee and N.Y. Chong , "Adaptive flocking of robot swarms: Algorithms and properties", *IEICE Trans. on Communications, vol. E91-B, no.9, pp.* 2848–2855*, September 2008.*

[72] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2006.

[73] Y. Yang, N. Xiong, N. Y. Chong and X. Défago, "A decentralized and adaptive flocking algorithm for autonomous mobile robusts", *IEEE 3th Intl. Conf. on Grid and Pervasive Computing-Workshops, pp. 262-268, 2008.*

[74] Anh Duc Dang and Joachim Horn, "Formation adaptation control of autonomous robots in a dynamic environment", *Proceedings of the 2015 IEEE International Conference on Industrial Technology, Seville, Spain, pages 3190-3195, March 17-19, 2015.*

[75] Y. Dai, K. S. Choi and S. G. Lee "Adaptive formation control and collision avoidance using a priority for nonholonomic mobile robots", *Intl. Jour. of Advanced Robotics System, vol. 10, , pp.* 1–14*, 2013.*

[76] Wenwu Yu and Guanrong Chen, "Robust adaptive flocking control of nonlinear multi-agent systems", *IEEE Intl. Sym. on Computer-Aided Control system Design, pp. 363-367, September 2010.*

[77] H. G. Tanner, A. Jadbabai, and G. J. Pappas, "Stable flocking of mobile agents, part i: fixed topology", *Proceedings of the 42nd IEEE Conference on Decision and Control, pages 2010–2015, 2003.*

[78] Y. Liu and K. M. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 1, pp. 30–44, 2004.

[79] Saptarshi Bandyopadhyay, Chakravarthini M. Saaj and Bijnan Bandyopadhyay, "Stability analysis of small Satellite formation flying and reconfiguration missions in deep space ", *12th IEEE Workshop on Variable Structure System, VSS'12, pp. 285-290, January 2012.*

[80] Veysel Gazi and Kevin M. Passino,"Stability analysis of swarms", *IEEE Transactions on Automatic Control*, vol. 48, No. 4, pp. 692–697, April 2003.

[81] L. Moreau, "Stability of multiagent systems with time–dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[82] K.-H. Tan and M. A. Lewis, "Virtual structure for high-precision cooperative mobile robot control", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 132-139*, 1999.

[83] Chang Boon Low, "A dynamic virtual structure formation control for fixed-wing UAVs", *IEEE Intl. Conf. on control and Automation, pp. 627-632, December, 2011.*

[84] Huizhen Yang and Fumin Zhang, "Geometric formation control for autonomous underwater vehicles", *IEEE Intl. Conf. on Robotics and Automation, pp. 4288-4293, May, 2010.*

[85] Aveek Das, Rafael Fierro, Vijay Kumar, Jim Ostrowski, John Spletzer and Camillo Taylor, "A framework for vision based formation control", *IEEE Transactions on Robotics and Automation,Vol. XX, No. Y, Month, pp. 1-13, 2001.*

[86] X. Sun, T.Mao, J. D. Kralik, and L. E. Ray, "Cooperative multi-robot reinforcement learning: A framework in hybrid state space," *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (*IROS 2009*), St. Louis, MO, USA*, 2009.

[87] T. Eren, P. N. Belhumeur, B. D. O. Anderson, and A. S. Morse, "A framework for maintaining formations based on rigidity," *Proceedings of the* IFAC *World Congress*, pp. 2752–2757, 2002.

[88] Tolga Eren, "Formation shape control based on bearing rigidity", *Intl Journal of control, Vol. 85, No. 9, pp. 1361–1379, September, 2012.*

[89] S. P. Hou, C. C. Cheahand J. J. E. Slotine, "Dynamic Region Following Formation Control for a Swarm of Robots", *IEEE Intl. Conf. on Robotics and Automation, pp. 1929-1934, May, 2009.*

[90] John R. Spletzer and Rafael Fierro. "Desired positioning strategies for shape changes in robot teams*", IEEE Intl. Conf. on Robotics and Automation, pp. 742-747, April, 2005.*

[91] Anh Duc Dang and Joachim Horn, "Collinear formation control of autonomous robots to move towards a target using artificial force fields", *Proceedings of the 2015 IEEE International Conference on Technologies for Practical Robot Applications, Boston, USA, pages 1-6, May 11-12, 2015.*

[92] Anh Duc Dang and Joachim Horn, "Formation control of autonomous robots following desired formation during tracking a moving target", *Proceedings of the 2nd IEEE International Conference on Cybernetics, Gdynia, Poland, pages 160-165, June 24-26, 2015.*

[93] Geunho Lee, Seokhoon Yoon, Nak Young Chong and Henrik Christensen, "A mobile sensor network forming concentric circles through local interaction and consensus building", *Journal of Robotics and Mechatronics, vol.21, no. 4, pp.469-477,2009.*

[94] Chen Wang, Guangming Xie and Ming Cao, "Forming circle formation of anonymous mobile agents with order preservation", *IEEE Transactions on Automatic Control*, *vol. 58, no.12, pp. 3248–3254, December 2012.*

[95] Randy A. Freeman and Petar V. Kokotovíc, "Robust Nonlinear Control Design: State-Space and Lyapunov Techniques, Birkhäuser", 1996.

[96] Lial and Hungerford, "Mathematics with applications", *8th edition, Addison Wesley, 2003.*

[97] Horst Stöcker, "Taschenbuch mathematischer Formeln und moderner Verfahren", 4*., korrigierte Auflage, Harri Deutsch, Frankfurt am Main, 1999.*

[98] David Halliday and Robert Resnick, "Physik 1", *Walter de Gruyter, Berlin, New York, 1993.*

[99] Steffen Bohrmann, Rudolf Pitka, Horst Stöcker, Georg Terlecki "Physik für Ingenieure ", *Harri Deutsch, Frankfurt am Main, Thun, 1993.*

Curriculum Vitae

Personal Data

Name: Anh Duc Dang

Date of birth: 14.12.1977

Place of birth: Thai nguyen, Vietnam

Nationality: Vietnamese

E-Mail: adang@hsu-hh.de

Education

12/ 2005	Received the M.S. degree from Thai Nguyen University of Technology, Thai Nguyen, Vietnam, in Major Electrical Engineering.
05/ 2001	Received the B.S. degree from Thai Nguyen University of Technology, Thai Nguyen, Vietnam, in Major Electrical Engineering.

Experience

05/2010 – 02/2017	Research Assistant, Helmut Schmidt University / University of the Federal Armed Forces Hamburg, Germany.
05/2009 – 05/2010	Lecturer and researcher of the Electronics - Electrical Engineering Department, Hanoi University of Economic and Technical Industries.
02/2006 – 07/2006	Internship in automation technology, HTW university and vocational training center in Sulzbach, Saarland, Germany.
04/2005 – 05/2009	Lecturer and researcher of the Electronics Engineering Department, Thai Nguyen University of Technology.
09/2001 – 04/2005	Lecturer and researcher of the Electrical Engineering Department, Thai Nguyen University of Technology.