

UNIVERSITY OF PISA

DEPARTMENT OF ELECTRICAL SYSTEMS AND AUTOMATION



UNIVERSITÀ DI PISA

**OBSTACLE AVOIDANCE FOR A GAME THEORETICALLY CONTROLLED
FORMATION OF UNMANNED VEHICLES**

Doctoral Thesis by

Mehmet Eren ERDOĞAN

Cycle XXII

Supervisor: Prof. Mario INNOCENTI

APRIL 2011

UNIVERSITY OF PISA

DEPARTMENT OF ELECTRICAL SYSTEMS AND AUTOMATION



UNIVERSITÀ DI PISA

**OBSTACLE AVOIDANCE FOR A GAME THEORETICALY CONTROLLED
FORMATION OF UNMANNED VEHICLES**

Doctoral Thesis by

Mehmet Eren ERDOĞAN

Cycle XXII

Supervisor: Prof. Mario INNOCENTI

Student: Mehmet Eren ERDOĞAN

APRIL 2011

FOREWORD

I would like to extend my sincere gratitude and appreciation to Prof. Mario Innocenti and Prof. Lorenzo Pollini for their guidance, supports and helps.

All my past and present friends and colleagues deserve special thanks, and I must say that it has been four years full of joy and experience thanks to them. It was a pleasure for me to share the same ambient with them during my doctoral studies.

Acknowledgement also goes to all the department members which assisted and helped me a lot to adapt especially during my first years in the faculty.

Also, I would like to thank a lot to my parents which always supported, and motivated me and have been very close to me even if they were physically far away. The completion of this thesis would not have been possible without their encouragement and support.

Finally, my very special, endless and immeasurable thanks go to my wife and little son. My little son has always been a motivation factor for me. I am forever grateful for my wife steadfast encouragement, love, and willingness to endure many sacrifices, to enable me to pursue this thesis, which without her support I would not have had the confidence to finish.

This thesis is dedicated to my beloved wife and son.

April 2011

Mehmet Eren Erdoğan

TABLE OF CONTENTS

	<u>Page</u>
TABLE OF CONTENTS	vi
ABBREVIATIONS	vii
LIST OF FIGURES	ix
1. INTRODUCTION	1
1.1. A Historical Survey of Robotics	4
2. PROBLEMS CONSIDERED AND MOTIVATING APPLICATIONS	7
2.1. Navigation and Obstacle Avoidance	7
2.2. Formations and Multi-Agents Robotics	7
3. LITERATURE SURVEY	10
3.1. Navigation and Obstacle Avoidance	10
3.2. Formations and Multi-Agents Robotics	11
4. BACKGROUND ON GAME THEORETICAL APPROACH	16
5. MODEL AND CONTROL	18
5.1. Robot Dynamics	18
5.2. Formation of Robots	19
5.3. Formation Cost Functions	21
5.4. Nash Equilibrium and Differential Games	23
5.5. NSB Behavioral Control	26
6. STABILITY ANALYSIS	31
6.1. Stability Analysis for Receding Horizon Nash Control	31
6.2. Stability Analysis for NSBBC Obstacle Avoidance Algorithm	32
7. SIMULATIONS	33
8. CONCLUSION	52
REFERENCES	54

ABBREVIATIONS

SLAM	: Simultaneous Localizations And Mapping
GPS	: Global Positioning System
NSBBC	: Null Space Based Behavioral Control
VLSR	: Very Large Scale Robotic

LIST OF FIGURES

	<u>Page</u>
Figure 1.1 : Water Raising Device of Al-Jazari.....	4
Figure 1.2 : Elektro, the Westinghouse Motoman.	5
Figure 1.3 : Honda’s Asimo Serving Coffee.....	6
Figure 5.1 : Triangular Formation of Robots.....	19
Figure 5.2 : A Formation of 4 Robots and Related Incidence Matrix.	19
Figure 5.3 : A case of two simultaneous behaviors: v_1 is the output of an obstacle avoidance task while v_2 is the output of a trajectory tracking task.	27
Figure 5.4 : Sketch of the null-space-based behavioral control in a 2-task example	27
Figure 7.1 : Triangle formation shape.....	33
Figure 7.2 : Phase plane of unmanned vehicles where the radius d is 3 times the real radius of the obstacle.....	34
Figure 7.3 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.2.	34
Figure 7.4 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.2.	35
Figure 7.5 : Phase plane of unmanned vehicles where the radius d is equal to the real radius of the obstacle.....	35
Figure 7.6 : Phase plane of unmanned vehicles where all the vehicles execute obstacle avoidance algorithm.....	35
Figure 7.7 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.6.	36
Figure 7.8 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.6.	36
Figure 7.9 : Phase plane of unmanned vehicles with a moving small obstacle which moves toward the vehicles where all the vehicles execute the obstacle avoidance algorithm.....	37
Figure 7.10 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.9.	37
Figure 7.11 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.9.	38
Figure 7.12 :Control signals that affect the leader vehicle for the simulations shown in Fig.7.9	38
Figure 7.13 : Phase plane of unmanned vehicles with a moving small obstacle which crosses the trajectory of the vehicles where all the vehicles execute the obstacle avoidance algorithm.....	39
Figure 7.14 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.13.	39
Figure 7.15 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.13.	39

Figure 7.16 :Control signals that affect the leader vehicle for the simulations shown in Fig.7.13 .	40
Figure 7.17 : Phase plane of unmanned vehicles with a moving small obstacle which moves on the same direction with the vehicles where all the vehicles execute the obstacle avoidance algorithm.	40
Figure 7.18 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.17.	41
Figure 7.19 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.17.	41
Figure 7.20 :Control signals that affect the leader vehicle for the simulations shown in Fig.7.17 .	41
Figure 7.21 : Phase plane of unmanned vehicles with a moving small obstacle which moves toward the vehicles where all the vehicles execute the obstacle avoidance algorithm and one of the vehicles slightly collides with the obstacle.	42
Figure 7.22 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.21.	42
Figure 7.23 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.21.	43
Figure 7.24 :Control signals that affect the leader vehicle for the simulations shown in Fig.7.21 .	43
Figure 7.25 : Phase plane of unmanned vehicles with a moving bigger obstacle which moves toward the vehicles where all the vehicles execute the obstacle avoidance algorithm.	44
Figure 7.26 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.25.	44
Figure 7.27 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.25.	44
Figure 7.28 :Control signals that affect the leader vehicle for the simulations shown in Fig.7.25 .	45
Figure 7.29 : Phase plane of unmanned vehicles with a suddenly appearing small obstacle where all the vehicles execute the obstacle avoidance algorithm	45
Figure 7.30 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.29.	46
Figure 7.31 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.29.	46
Figure 7.32 :Control signals that affect the leader vehicle for the simulations shown in Fig.7.29 .	46
Figure 7.33 : Phase plane of unmanned vehicles with a suddenly appearing bigger obstacle where all the vehicles execute the obstacle avoidance algorithm	47
Figure 7.34 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.33.	47
Figure 7.35 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.33.	47
Figure 7.36 :Control signals that affect the leader vehicle for the simulations shown in Fig.7.33 .	48

Figure 7.37: Phase plane of unmanned vehicles with two immobile obstacles where all the vehicles execute the obstacle avoidance algorithm.....	48
Figure 7.38 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.37.	49
Figure 7.39 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.37.	49
Figure 7.40 : Control signals that affect the leader vehicle for the simulations shown in Fig.7.37	49
Figure 7.41: Phase plane of unmanned vehicles with two successively immobile obstacles where all the vehicles execute the obstacle avoidance algorithm	50
Figure 7.42 : X position errors of the unmanned vehicles for the simulation shown in Fig.7.41.	51
Figure 7.43 : Y position errors of the unmanned vehicles for the simulation shown in Fig.7.41.	51
Figure 7.44 : Control signals that affect the leader vehicle for the simulations shown in Fig.7.41	51

1. INTRODUCTION

The word "robot" (from *robota*, Czech for "work") made its public debut in 1920, when it premiered on stage in Karel Čapek's play *R.U.R.* (Rossum's Universal Robots) [1]. The play told of a world in which humans relaxed and enjoyed life while robots - imitation humans - happily did whatever labor needed to be done. Reality has not yet reached that far, but today there are machines that have a high degree of autonomy and that can independently perform very complex tasks in which they interact with the environment and make on-line decisions. Much of the progress has taken place in the past few decades, during which the continuous development of hardware has made way for increasingly more advanced software.

Traditionally robots were thought of as humanoids, but today according to most people working in the robotics field, what defines a robot is its ability to function independently rather than its physical appearance. In 1998, Professor Ronald C. Arkin, co-worker and later director of the Mobile Robot Laboratory at Georgia Institute of Technology, made below written definition [2];

"An intelligent robot is a machine able to extract information from its environment and use knowledge about its world to move safely in a meaningful purposive manner".

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

Robots can adapt to complex environments and perform tasks more intelligently by working in groups. Robot groups may be composed of many different kinds of robots like ground vehicles, aerial vehicles, underwater vehicles or spacecrafts. A robot group may be homogenous; each member in the group may be identical, or it can be heterogeneous; the group may include different kinds of robots. Using a team of simple robots is advantageous than using a single but more complicated robot in many ways. Robot's

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

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

The shape formation in mobile robots is a challenging topic and there are many researches on that subject. For robot groups coordinating with shape formation, the flexibility of the shape formation is very important. With the increasing demand for autonomous robots in different fields, many different kinds of formation shapes are required. In non-idealized environments, forming many of the simple shapes may not be feasible. Besides, many different task definitions may require very complicated formation shapes. Another important issue of shape formation is the fault-tolerance. The shape formation algorithm should guarantee the completion of the task even if some of the group members are hurt. Since different tasks require different types of robot groups, a formation shape algorithm should also be flexible in the number and the heterogeneity of the team members.

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

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

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

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

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

Given the a priori knowledge of the environment and the goal position or trajectory to track, mobile robot navigation refers to the robot's ability to safely move towards the goal using its knowledge and the sensorial information of the surrounding environment. Even though there are many different ways to approach navigation, most of them share a set of common components or blocks, among which path planning and obstacle avoidance play a key role. Given a map and a goal location, path planning involves finding a geometric path from the robot actual location to the goal. This is a global procedure whose execution performance is strongly dependent on a set of assumptions that are seldom observed in nowadays robots. In fact, in mobile robots operating in unstructured environments, or in service and companion robots, the a priori knowledge of the environment is usually absent or partial, the environment is not static, i.e., during the robot motion it can be faced with other robots, humans or pets, and execution is often associated with uncertainty. Therefore, for a collision free motion to the goal, the global path planning has to be associated with a local obstacle handling that involves obstacle detection and obstacle avoidance. Obstacle avoidance refers to the methodologies of shaping the robot's path to overcome unexpected obstacles. The resulting motion depends on the robot actual location and on the sensor readings. There are a rich variety of algorithms for obstacle avoidance from basic re-planning to reactive changes in the control strategy. Proposed techniques differ on the use of sensorial data and on the motion control strategies to overcome obstacles.

The thesis provides a game theoretical approach to the control of a formation of unmanned vehicles. The objectives of the formation are to follow a prescribed trajectory, avoiding obstacle(s) while maintaining the geometry of the formation. Formation control is implemented using game theory while obstacles are avoided using Null Space Based Behavioral Control algorithm. Different obstacle avoidance scenarios are analyzed and compared. Numerical simulation results are presented, to validate the proposed approach.

In Chapter 2 it is given the considered problems and motivating applications. A literature review on formation control and obstacle avoidance can be found in Chapter 3. In Chapter 4 it is provided background on game theoretical approach. Chapter 5 is on the modeling and control of mobile robots. In Chapter 6 stability analysis both for receding horizon Nash control and NSBBC algorithm are made. Numerical simulations are shown in Chapter 7. Finally Chapter 8 concludes the thesis and indicates possible future directions.

1.1 A Historical Survey of Robotics

Constructing machines that can interact with the environment and even help or replace humans in performing dangerous or tedious tasks is not a new idea. Early work in automation and robotics was made by, for instance, the Arab engineer Al-Jazari (1136-1206) who, among other things, invented the earliest known automatic gates, which were driven by hydropower [3], [4]. He also invented automatic doors as part of one of his elaborate water clocks, and designed and constructed a number of other automata, including automatic machines and home appliances powered by water. According to Encyclopedia Britannica, the Italian Renaissance inventor Leonardo da Vinci may have been influenced by the classic automata of Al-Jazari. Later, in the 15th century, Leonardo da Vinci made drawings for the construction of a mechanical knight [5]. In the following centuries, many similar ideas saw the light of day, but not until the last century has technology reached the point where the realization of truly interacting machines is possible.



Figure 1.1 : Water Raising Device of Al-Jazari.

The American company Westinghouse Electric Corporation produced a series of human-resembling machines in the 1920's and 30's, some of them which could perform simple tasks such as vacuum cleaning. None of these machines were in a strict sense interacting with the environment but one of them, the humanoid "Elektro" (1939), had among his other skills (including blowing balloons and smoking cigarettes) the ability to distinguish between red and green light [6].



Figure 1.2 : Eletkro, the Westinghouse Motoman.

The first machines that could actually respond to stimuli are claimed to be Elmer and Elsie [7,8], two turtle-like machines on wheels that were developed by neurophysiologist William Grey Walter at Burden Neurological Institute, England, in 1948 - 49. Elmer and Elsie (names originating from ELectroMEchanical Robot, Light-Sensitive) had a light sensor, touch sensor, propulsion motor, steering motor, and a two vacuum tube analog computer. Even with this simple design, Grey demonstrated that his robots exhibited complex behaviors. By attaching light emitting sources on each of the two robots, they could even be made to interact with each other, something that was considered quite revolutionary at the time. Soon thereafter the first commercial industrial robots entered the market. The first models were only used to perform easy and repetitive tasks in static environments, such as for instance pick and place operations, painting, welding, etc., but over the years industrial robots have become increasingly more advanced and are now used in settings where a high degree of autonomy is required.

The rapid process in the electronics field has been the essential for the development within the field of robotics. The first robots had very simple control circuits based on electron tubes. In 1947, researchers at Bell Laboratories invented the transistor, which had the benefits of being much smaller and requiring significantly less power than the electron tube. This new device soon replaced the electron tube in most applications, but the real breakthrough for robotics came after the launch of the programmable microprocessor in the early seventies. Not only were these processors small enough to be incorporated in a freely movable body, they also made the cost for computer power drop dramatically.

There was a boost in the market for industrial and military robots after the arrival of the microprocessor. In the seventies and early eighties many companies started activities in the field of industrial robotics; among them were companies such as General Motors, General Electric, ASEA and KUKA. Several Japanese companies also joined this new trend and soon industrial robots became a common sight in manufacturing industries. However, nothing like that was seen in the market for domestic robots or entertainment robots intended for personal or small scale use. Even though public interest has always been substantial and experienced yet another top in the early eighties with the release of the Star Wars films and TV series like Star Trek, progress has been very slow in this area. Most probably the main reason for this has been the hardware cost. Prices on sensors and high precision mechanics have not decreased at the same rate as the price on computer power and are still comparatively high. Hardware prices have made it nearly impossible for companies that produce robots for private use to be commercially profitable. Besides this, another damping factor is that it has turned out to be more difficult than many anticipated to mimic the amazing ability seen in animals and humans to efficiently weed out relevant information from a potentially very large data set and to combine information to draw the "right" conclusions. In the attempts to solve these problems, a wide range of more or less independent research areas have evolved, covering disciplines such as computer vision, filtering, speech recognition and data fusion.

In recent years, the attempts to manufacture "Artificial Intelligence" have started to pay off. Today's robots are more reliable and can handle much more complex situations than their predecessors. Also, in the last years, prices on hardware have started to go down [9]. Although many technical problems remain to be solved, robots for civil use, like Honda's Asimo, have started to appear in a variety of different areas. In the future we will most likely see robots in many new applications.



Figure 1.3 : Honda's Asimo serving coffee.

2. PROBLEMS CONSIDERED AND MOTIVATING APPLICATIONS

The research problems considered in this thesis stem from two areas of robotics. Both are in the subfield called mobile robotics. The first research area is navigation and obstacle avoidance which basically deals with the question of getting from A to B in a safe and efficient manner. The second research area is multi-robot coordination. This is a somewhat broader area where the common theme is that of trying to achieve a collective goal using a group of robots, e.g. stay in a prescribed formation.

2.1 Navigation and Obstacle Avoidance

The problem of programming a mobile robot to move from one place to another is of course as old as the first mobile robot. This is, however, not as easy as one might think. Questions like “What path should be chosen to get to the goal location?” and “How fast and how close to the obstacles can the robot go without compromising safety?” need to be considered.

Obstacle avoidance refers to the methodologies of shaping the robot’s path to overcome the obstacles. The resulting motion depends on the robot actual location and on the sensor readings. There are a rich variety of algorithms for obstacle avoidance from basic re-planning to reactive changes in the control strategy. Proposed techniques differ on the use of sensorial data and on the motion control strategies to overcome obstacles.

In the literature there are various applications where it is assumed that it is given a high-quality map of the immediate surroundings of the robot. It must be noted however, that map building and localization contains a whole research field in itself. In [10], the area of simultaneous localization and mapping (SLAM) is investigated. Since GPS is not an option for indoor applications, this problem is quite hard, as can be seen by comparing a 15th century explorers map with the satellite images available today.

2.2 Formations and Multi-Agents Robotics

The thought of cooperating robots has received an increasing amount of attention in recent years. Besides the philosophical interest in cooperating machines the main reason is to try to take advantage of “strengths in numbers”, i.e., that there are properties like

- Efficiency
- Flexibility
- Redundancy/Robustness
- Price reduction
- Feasibility

to be gained. Having several robots doing something often means that you have the flexibility of dividing the robots into groups working at different locations. Having many also implies robustness, since losing one robot leaves the others intact to finish the mission. The mobile robots of today are typically produced in small numbers; however, if there is a big increase in multi-agent applications, there might be a price reduction due to mass production benefits. Efficiency can be gained in terms of e.g. fuel consumption in formation flight. Finally, some missions are impossible to carry out with only one robot; these include deep space interferometry, a satellite imaging application, and the surveillance of large areas or buildings.

Following [2], the coordination problems can be divided into the following fields;

- Foraging/Consuming, where randomly placed objects in the environment are to be found and either carried somewhere or operated on in place. This includes collecting rocks on e.g. Mars.
- Grazing, where an area should be swept by sensors or actuators. This includes lawn mowing (e.g. the Husqvarna Solar Mower) and vacuuming (e.g. the Electrolux Trilobite). Special cases of area sweeping include so-called search-and-rescue and pursuit-evasion scenarios. In these situations the looked for item, e.g. a missing person or an enemy vehicle, is moving.
- Formation keeping, where the robots are to form some geometric pattern and maintain it while moving about in the world. Applications include formation flight for fuel efficiency and coordinated motion for collaborative lifting of large objects. In this area biological influences are very common and efforts are being made both to understand animal flocking/schooling and copy their effective strategies. A satellite application called deep space interferometry requires a large and exact sensor spacing which would be impossible to achieve with a single satellite.
- Traffic control, where a number of vehicles share a common resource, highways or airspace, while trying to achieve their individual goals. In automated highway projects, the problem scales have ranged from keeping inter-vehicle spacing (formation keeping) via lane changes to choosing routes that minimize the overall effect of traffic jams. Air traffic control investigations, are motivated by the increasing congestion around major airports. The hope is to improve efficiency without compromising the vital safety.

The properties of flexibility and robustness is of course very attractive with the armed forces as is shown in the following quote *“The U.S. military is considering the use of multiple vehicles operating in a coordinated fashion for surveillance, logistical support, and combat, to offload the burden of dirty, dangerous, and dull missions from humans.”* Problems facing a multi-agent team operating in a limited space include blockage and collisions. More generally, a highly distributed system might generate competition rather than cooperation. Attempts to exploit such inter-robot competition in a market economic framework has been investigated in e.g. [11]. Finally there is always a cost of communication, in terms of additional hardware, increased computational load, and energy consumption. The old saying “Too many cooks spoil the broth” may explain the possible drawbacks of a multi-agent approach.

3. LITERATURE SURVEY

In this section we will take a look at examples of current research in the subfields of obstacle avoidance and formation control.

3.1 Navigation and Obstacle Avoidance

The problem of navigation and obstacle avoidance deals with making a robot move from one position to another as efficiently as possible, while not bumping into things on the way. There are a rich variety of algorithms for obstacle avoidance from basic re-planning to reactive changes in the control strategy. Proposed techniques differ on the use of sensorial data and on the motion control strategies to overcome obstacles.

The Bug's algorithms [12], [13], follow the easiest common sense approach of moving directly towards the goal, unless an obstacle is found, in which case the obstacle is contoured until motion to goal is again possible. In these algorithms only the most recent values of sensorial data are used.

Path planning using artificial potential fields, [14], is based on a simple and powerful principle that has an embedded obstacle avoidance capability. The robot is considered as a particle that moves immersed in a potential field generated by the goal and by the obstacles present in the environment. The goal generates an attractive potential while each obstacle generates a repulsive potential. Obstacles are either a priori known, (and therefore the repulsive potential may be computed off-line) or on-line detected by the on-board sensors and therefore the repulsive potential is on-line evaluated. Besides the obstacle avoidance functionality, the potential field planning approach incorporates a motion control strategy that defines the velocity vector of the robot to drive it to the goal while avoiding obstacles.

The Vector Field Histogram, [15], generates a polar histogram of the space occupancy in the close vicinity of the robot. This polar histogram, which is constructed around the robot's momentary location, is then checked to select the most suitable sector from among all polar histograms sectors with a low polar obstacle density and the steering of the robot is aligned with that direction.

Elastic bands [16] as a framework that combines the global path planning with a real-time sensor based robot control aiming at a collision free motion to the goal. An elastic

band is a deformable collision-free path. According to [16], the initial shape of the elastic band is the free path generated by a planner. Whenever an obstacle is detected, the band is deformed according to an artificial force, aiming at keeping a smooth path and simultaneously maintaining the clearance from the obstacles. The elastic deforms as changes in the environment are detected by the sensors, enabling the robot to accommodate uncertainties and to avoid unexpected and moving obstacles.

A dynamic approach to behavior-based robotics proposed in [17], [18], models the behavior of a mobile robot as a non-linear dynamic system. The direction to the goal is set as a stable equilibrium point of this system while the obstacles impose an unstable equilibria point of this non-linear dynamics. The combination of both steers the robot to the goal while avoiding obstacles.

The Null Space Based Behavioral Control (NSB Behavioral Control) strategy is based on degrading the general obstacle avoidance process into smaller problems which are less complex [19], [20]. Each task velocity is computed as if it were acting alone; then, before adding its contribution to the overall vehicle velocity, a lower-priority task is projected onto the null space of the immediately higher-priority task so as to remove those velocity components that would conflict with it.

In this thesis we make use of Null Space Based Behavioral Control as the obstacle avoidance algorithm.

3.2 Formations and Multi-Agents Robotics

In the recent years the coordination of multi-robot systems has been subjected to considerable research efforts. The main motivation is that in many tasks a group of robot can perform more efficiency than a single one and can accomplish tasks not executable by a single robot. Multi-robot systems have advantages like increasing tolerance to possible vehicle fault, providing flexibility to the task execution or taking advantage of distributed sensing and actuation [21]. Each animal in a herd, for instance, benefits by minimizing its encounters with predators [22]. Balch and Arkin [23] argued that two or more robots can be better than one for several reasons:

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

Among the tasks that are done with a robot group, operating in a special formation increases the capability of the robot team in many ways. Shape formation during the

operation of a task enhances the system performance by increasing instrument resolution and cost reduction. In [24], it is stated that global security and efficiency of the team can be enhanced by a proper configuration for the formation. Formations allow individual team members to concentrate their sensors across a portion of the environment while their partners cover the rest. In [25], it is stated that air force fighter pilots for instance direct their visual and radar search responsibilities depending on their position in a formation. Formation in a proper configuration is one of the ways to get the maximum efficiency from a robot team. There are many tasks that the shape formation of autonomous robots can be used. Examples in the literature include box pushing [26], load transportation [27], dispersing a swarm [28], [29], moving in formation [25], covering areas while maintaining constraints [30], perform shepherding behaviors [30] and enclosing an invader [31].

Shape formation of multiple mobile robots is a challenging subject. This subject includes many sub-problems like decision of the feasible formation shape, getting into formation, maintenance of the formation shape and switching between the formations. Shape formation and maintenance of the formation is one of the important problems in the shape formation on which much research has been done. There are many different approaches to modeling and solving these problems, ranging from paradigms based on combining reactive behaviors [23], to those based on leader-follower graphs [32] and potential field methods [33]. One of the common methods is to determine the desired position of each member within the group to control each robot to these specified positions. This methods works fine when the number of the group is small. When the number of robots increases, it becomes difficult and inefficient to manually determine the position of each and every agent within the formation. There are some approaches for formation control which are inspired by biological systems. Biologists who study animal aggregations such as swarms, flocks, schools, and herds have observed the individual-level behaviors which produce the group-level behaviors [34], [35]. In some studies this observation are applied on robot groups and the animal behaviors are mimicked by the robots. One of the well-known applications in this field is by Reynolds [36]. He developed simple egocentric behavior model for the individuals of the simulated group of birds or so-called "Boids". In this model, the basic flocking model consists of three simple steering behaviors which describe how an individual Boid maneuvers based on the positions and velocities its nearby flockmates. First behavior is separation which is steering to avoid crowding local flockmates. The other behavior is alignment which is steering towards the average heading of local flockmates and the last is cohesion; steering to move toward the average position of local flockmates. Reynolds showed that Boids behave just like real birds. Vicsek et al. reported the group behavior of real bacteria by simple model [37]. The simple "nearest neighbors" method is proposed in

order to investigate the emergence of autonomous motions in systems of particles with biologically motivated interaction. In this method, particles are driven with a constant absolute velocity and they choose the average direction of motion of the particles in their neighborhood with some random perturbation added. The developed model showed a good approximation to the motion of bacteria that exhibit coordination motion in order to survive under unfavorable conditions. This idea has then been widely used in the literature to attack the problem of modeling the coordinated motion of a group of autonomous mobile robots [38], [39], [40], [41].

Leader follower method is one of the most common approaches for formation control. In the leader following method one or more robots are assigned as leaders and responsible for guiding the formation. The other robots are required to follow the leader according to predefined behaviors. Examples include papers by Wang [42], presented some simple strategies for a fleet of autonomous robots to navigate in formation and studied the interaction dynamics of these robots with the presented navigation strategies. In this study, several strategies which are based on leader following and neighbor following are presented. The presented strategies include “Nearest-Neighbor Tracking” in which each robot is assigned to maintain its desired position according to its nearest neighbor. Another method presented is “Multi-Neighbor Tracking” in which several robots are assigned as leaders or the guardians of the fleet. [43] and [44] are some more recent examples of the formation control using the leader-follower strategy.

Behavior based approach is used in many studies for shape formation. In this approach, shape formation of the whole group is achieved through of the individual agents by using the weighted sum of some basic and intuitive behaviors. We can see successful applications of this idea in the subsumption architecture [45], [46], [47].

Balch and Arkin presented a behavior-based approach to robot formation keeping [48]. In this study, new reactive behaviors for implementing formations in robot groups are presented and evaluated. In this study, several motor schemas, move-to-goal, avoid-static-obstacle, avoid-robot and maintain formation are introduced. Each schema represents a vector representing the desired behavioral response to the current situation of the robot and the group. A gain value is indicated representing the importance of individual behaviors. The high-level combined behavior is generated by multiplying the outputs of each primitive behavior by its gain, summing and normalizing the result. This method makes the robot group to be able to move to a goal location while keeping in formation, avoiding obstacles and collision with other robots. In [49], this approach is extended by an additional motor schema which is based on a potential field method.

In [21], a novel behavior based approach is introduced for a platoon of mobile robots to shape formation while avoiding collision with themselves and external obstacles. It uses

a hierarchy-based approach so called Null-Space based Behavioral (NSB) control. This control uses the null-space projection to obtain the final motion command from outputs of multiple conflicting tasks.

Potential function approaches to robot navigation provide an elegant paradigm for expressing multiple constraints and goals in mobile robot navigation problems [50]. One of the first work applying artificial potentials to agent coordination is [51]. In this approach a distributed control for very large scale robotic (VLSR) systems is presented. Simple artificial force laws between pairs of robots or robot groups are introduced. This force laws are inverse-power force laws which incorporates both attraction and repulsion. These forces are used to reflect “social relations” among robots to a degree and therefore this method is called “Social Potential Fields”. In this method, each robot senses the resultant potential from components like other robots, obstacles, objectives etc. and acts under the resultant force. In this approach the parameters can be chosen arbitrarily to reflect the relationship between the robots whether they should stay close together or far apart to form the desired formation shape.

Yamaguchi and Arai [52] define a potential field on the space according to the relative distances between neighbors. In this study, the shape-generation problem is approached using systems of linear equations. Each robot, starting at some initial location, changes its position according to a linear function of its neighbors’ positions and some fixed constant. Simulations of the method show that a group of initially collinear robots will converge into the shape of an arc.

Song and Kumar [33] introduced a framework for control a group of robots for cooperative manipulation task. In this framework, the trajectory generation problem for cooperative manipulation task is addressed. This framework allows the robots to approach the target object, organize themselves into a formation that will trap the object and then transport the object to the desired destination. The robots in the group can also avoid static obstacles. The controllers are derived from simple potential fields and the hierarchical composition of the potential fields.

In [25], an approach which is inspired by the way molecules “snap” into place as they form crystals; the robots are drawn to particular “attachment sites” positioned with respect to other robots. Using this approach, a new class of potential functions is developed for shape formation control of multiple robots homogeneous large scale robot teams while navigating to a goal location through an obstacle field.

In [53] a shape formation method is presented for a heterogeneous robot group. In this method, the robots are controlled to reach the goals while controlling the group geometry, individual member spacing and obstacle avoidance is managed. Bivariate normal probability density functions are presented to construct the surface which swarm

members move on to generate potential fields. Limiting functions are also introduced to provide tighter swarm control by modifying and adjusting a set of control variables forcing the swarm according to set constraints. In this method, the swarm member orientation and the swarm movement as a whole is controlled by the combination of limiting functions and bivariate normal functions.

In [54], the potential field approach is combined with virtual leaders proposed in [55]. A virtual leader is a moving reference point that affects the robots in the group by means of artificial potentials. Virtual leaders are used to maintain group geometry and direct the motion of the group. In this approach, the potential produced from functions of relative distance between a pair of neighbors. The control force for an individual is derived as the minus gradient of the sum of all potentials affecting that individual. This leads the individuals are driven to the minimum of the total potential. The desired group is achieved by designing local potentials with some pre-described inter-vehicle spacing associated with virtual leaders which are moving reference points.

In this thesis a game theoretical approach is used to control of a formation of unmanned vehicles. In game theory, the Nash equilibrium is a solution concept of a game involving two or more players, in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his or her own strategy unilaterally. Distributed control is synthesized by defining cost functions that include neighboring vehicles only, and a leader-follower approach is used with the leader's cost function incorporating trajectory tracking, while formation control implemented in the followers' performance index.

4. BACKGROUND ON GAME THEORETICAL APPROACH

Game theory is a branch of applied mathematics. The word “game” is inspired by parlor games such as chess, or field games such as football. Rules of parlor games and players’ behaviors are modified in the game theory. For instance, the act of bluffing in poker is quite similar posturing of nations about their military strength [56], [57]. We make decisions every day about whether a situation is important or not. Game theory deals with the choices of people in the real world [58]. Players would like to gain the best profit for themselves in the game theory. Therefore the theory is based on decision theory and utility theory [59].

One of the major successes in the field of economics and social sciences in the past decades has been the application of Game Theory to the modeling of social interactions of rational entities for the prediction of outcomes of conflicts among them [60]. It turns out that the same approach may be used in the modeling of robot swarms, since their formation may be thought of as a social interaction of individuals [61].

Game Theory can be defined as ‘the study of mathematical models of conflict and cooperation between intelligent rational decision-makers’ [60]. Therefore, it seems natural to explore this technique in order to represent the behavior of robots, since robots may be regarded as ‘intelligent rational decision-makers’. Evolutionary models have been developed using Game Theory where, obviously, the agents involved cannot be regarded as ‘intelligent rational’ entities [62] and the situations they usually are involved in concern mainly conflict and cooperation.

One very important thing to notice is what is meant by conflict. Conflict does not mean fight or engagement and does not presuppose an enemy. Even teammates have conflicts and even one single individual has conflicts. It is not our intention to analyze conflicts from a philosophical point of view, but we do not restrain ourselves on the usual definition of conflict as the fight between contraries. For our purposes, a conflict is established when one trait of personality leads to a different action than another trait of personality or when one robot has individual interests that are against another robot’s interests, but we suppose they have the same task objective. In this context, we are interested in modeling relationships between robots that are on the same side and we do not intend to model fights between groups of robots.

In this thesis we model the formation control as a non-cooperative game where the self-enforcing Nash equilibrium can be used as the formation control strategy. The self-enforcing concept implies that no player has incentive to deviate from its Nash equilibrium because no player can gain by unilaterally deviating from it. Robots can adopt this mechanism to establish their strategies to interact with other team members during the process of formation keeping.

Mobile robots with double integrator dynamics can be modeled as a controllable linear system. Formation control cost functions can be casted as a linear quadratic form by using graph theory. Therefore, the formation control of mobile robots with double integrator dynamics can be modeled as a linear-quadratic Nash differential game. Under the framework of this game, the formation control problem is converted to the coupled (asymmetric) Riccati differential equation problem.

The type of coupling between coupled Riccati differential equations depends on the information structure in a game. In the practical control, the state-feedback control is particular demanding. The best way to design a state feedback controller is to use the state feedback information structure in a game. However, the state-feedback differential game is analytically and computationally intricate due to its complex information structure. The open-loop information structure is based on the assumption that the only information players have is their present states and the model structure. It can be interpreted as such that the players simultaneously determine their strategies at the beginning of the game and use this open-loop solution for the whole period of the game. Due to its analytic tractability, the open-loop Nash equilibrium solution is, in particular, very popular for the problems where the underlying model can be described by a set of linear differential equations and individual objectives can be approximated by functions which quadratically penalize deviations from some equilibrium targets [63], [64]. The finite horizon open-loop Nash equilibrium can be combined with a receding horizon approach to produce a resultant receding horizon Nash control. The use of receding horizon control in differential zero-sum games has been reported in [65], [66], [67]. It works in such a way; at each step, a state is read and the first control signal in the control profile generated from the open-loop Nash equilibrium is used to control robots. At the next step, this procedure repeats again.

5. MODEL AND CONTROL

5.1 Robot Dynamics

In this thesis it is considered a m dimensional space where a group of unmanned vehicles are moving. A formation group consists of N vehicles, and each vehicle has double integrator dynamics. The position vector is $p_i = [x_i, \dots, x_i^m]^T \in \mathbb{R}^m$ for vehicle $i \in N$. The state vector for agent $i \in N$ is $z_i(t) = [p_i(t)^T, \dot{p}_i(t)^T]^T \in \mathbb{R}^{2m}$ and the desired state vector is, $z_i^d(t) = [p_i^d(t)^T, \dot{p}_i^d(t)^T]^T \in \mathbb{R}^{2m}$. The control vector is $u_i(t) \in \mathbb{R}^m$ ($i=1, \dots, N$). The control and state vectors are defined as follows:

$$u_i(t) \in \mathbb{U}, z_i(t) \in \mathbb{Z} \quad (1)$$

Vehicles' dynamics and the dynamics of the reference are assumed to be linear:

$$\dot{z}_i = az_i + bu_i \quad (2)$$

$$\dot{z}_i^d = az_i^d + bu_i^d \quad (3)$$

where $a = \begin{bmatrix} 0 & I_{(m)} \\ 0 & 0 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ I_{(m)} \end{bmatrix}$. $I_{(m)}$ is the m dimensional identity matrix. The

formation state and control vectors are: Position vector $p = [p_1^T, \dots, p_N^T]^T$, velocity vector $\dot{p} = [\dot{p}_1^T, \dots, \dot{p}_N^T]^T$, state vector $z = [z_1^T, \dots, z_N^T]^T$ and control vector $u = [u_1^T, \dots, u_N^T]^T$, where $p, \dot{p}, u \in \mathbb{R}^{mN}$ and $z \in \mathbb{R}^{2mN}$. The team dynamics and its desired target are then:

$$\dot{z}(t) = Az(t) + \sum_{i=1}^N B_i u_i(t), t \geq 0 \quad (4)$$

$$\dot{z}_i^d(t) = Az^d(t) + \sum_{i=1}^N B_i u_i^d(t), t \geq 0 \quad (5)$$

where $A = I_{(N)} \otimes a$ and $B_i = [0, \dots, 1, \dots, 0]^T \otimes b$, where the operator \otimes stands for Kronecker product. To optimize control performance, the convexity assumption is necessary for optimization algorithms.

Assumption 5.1 (Convexity Assumption): \mathbb{U} is a compact and convex subset of \mathbb{R}^m containing the origin in its interior, and \mathbb{Z} is a convex, connected subset of \mathbb{R}^{2m} containing z_i^d in its interior, for every i .

5.2 Formation of Robots

To maintain the connection between the unmanned vehicles graph theoretical approach can be used. A graph $G=(V,E)$ is specified by an edge set $E=\{(v_i,\dots,v_j)\in V\times V\}$ and vertex set $V=\{v_1,\dots,v_N\}$ which they identify the incidence relation between different pairs of vertices. If $\{i,j\}\in E$, the vertices i and j are called adjacent (or neighbors). We assume that the graph has no loops, that is $(v_i,v_j)\in V$ where $v_i\neq v_j$. Each formation member is a vertex of the graph while each edge represents the connection between the neighbors.

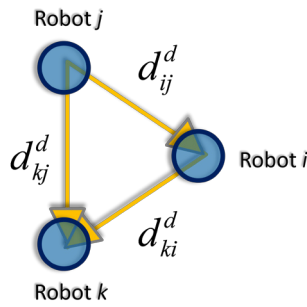


Figure 5.1: Triangular formation of robots

An edge-weighted graph is a graph that has a weight associated with each edge, i.e the weight of the edge (v_i,v_j) is associated with weight $w_{ij}\geq 0$. If there is a path of edges in E from any vertex $(v_i)\in V$ to any other vertex $(v_j)\in V$ in the graph, a graph is called connected. Connectivity is necessary to keep the formation of the team. To control a team to keep a formation, the graph connectivity is necessary.

Assumption 5.2 (Connectivity Assumption): Graph G is connected.

The incidence matrix shows the connection between two classes of objects. The incidence matrix B of a directed graph G is a $N\times q$ matrix with elements b_{ij} where N and q are the number of vertices V and edges E respectively. If i is the tail of edge $b_{ij}=-1$, if i is the head of edge $b_{ij}=+1$. Otherwise it is equal to zero. Below, an example how to write the incidence matrix is given.



Figure 5.2: A formation of 4 robots and related incidence matrix

In this thesis, the agents are form a directed connected graph, with edges to the neighboring agents only. The desired distance vector between the neighbors v_i and v_j is $d_{ij}^d = z_i^d - z_j^d$. The formation error vector can be written as $z_i - z_j - d_{ij}^d$ for the neighbors v_i and v_j . The team formation error can be expressed in a matrix form as follows;

$$\sum_{(i,j) \in E} w_{ij} \|z_i - z_j - d_{ij}^d\| = (z - z^d)^T \tilde{B} \tilde{W} \tilde{B}^T (z - z^d) \quad (6)$$

where $\tilde{B} = B \otimes I_{(2m)}$, $\tilde{W} = W \otimes I_{(2m)}$. W is defined as $W = \text{diag}[w_{ij}]$ which is the weight matrix in diagonal form with dimension q (number of edges E). By tuning the values of weighting matrices, the relative importance of the deviation of each of the states from their desired values can be weighted. By decreasing w_{ii} (which demonstrates the i th element of W), for instance, less importance is given to deviation of the related state from its desired value.

The Laplacian is an important matrix associated with a graph. The Laplacian can be used in a number of ways to provide interesting geometric representations of a graph.

Let σ be an arbitrary orientation of a graph G , and let B be the incidence matrix of G^σ . Then the Laplacian of G is the matrix $Q(G) = DD^T$. The Laplacian does not depend on the orientation σ , and hence is well-defined.

Theorem 5.1: Let G be a graph with N vertices and c connected components. If σ is an orientation of G and B is the incidence matrix of G^σ then $rkB = N - c$

Lemma 5.1: Let G be a graph with N vertices and c connected components. If Q is the Laplacian of G , then $rkQ = N - c$.

Proof 5.1: Let B be the incidence matrix of an arbitrary orientation of G . We shall show that $rkB = rkB^T = rkBB^T$, and the result then follows from Theorem 5.1. If $s \in \mathbb{R}^N$ is a vector such that $BB^T s = 0$, then $s^T BB^T s = 0$. But this is the squared length of the vector $B^T s$, and hence we must have $B^T s = 0$. Thus any vector in the null space of BB^T is in the null space of B^T , which implies that $rkBB^T = rkB$.

Following the above given data, the Laplacian matrix of graph G is;

$$Q = BWB^T \quad (7)$$

$$\tilde{Q} = \tilde{B} \tilde{W} \tilde{B}^T \quad (8)$$

where $\tilde{Q} = Q \otimes I_{(2m)}$. The Laplacian matrix Q is independent of the graph orientation, it is symmetric, its eigenvalues are real and it is positive semi-definite. This is also valid for \tilde{Q} .

For real value matrices X, Y, U, V with appropriate dimensions, the Kronecker product has the following properties:

$$\begin{aligned}(X \otimes Y)^T &= (X^T \otimes Y^T) \\ (X \otimes Y)(U \otimes V) &= (XU) \otimes (YV)\end{aligned}$$

Based on these properties, we have

$$\begin{aligned}\tilde{Q} &= \tilde{B}\tilde{W}\tilde{B}^T \\ &= (B \otimes I_{(2m)})(W \otimes I_{(2m)})(B \otimes I_{(2m)}) \\ &= Q \otimes I_{(2m)}\end{aligned}\tag{9}$$

The team formation error becomes;

$$\begin{aligned}\sum_{(i,j) \in E} w_{ij} \|z_i - z_j - d_{ij}^d\| \\ = (z - z^d)^T \tilde{Q} (z - z^d) \\ = \|z - z^d\|_{\tilde{Q}}^2\end{aligned}\tag{10}$$

5.3 Formation Cost Functions

A general cost function can be written as follows;

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt\tag{11}$$

The finite horizon cost function for vehicle i can be written as follows;

$$J_i = h_i(T, z(T)) + \int_0^T g_i(\tau, z(\tau), u(\tau)) d\tau\tag{12}$$

$$h_i(T, z(T)) = \sum_{(i,j) \in E} w_{ij} \|z_i(T) - z_j(T) - d_{ij}^d\|^2\tag{13}$$

$$g_i(\tau, z(\tau), u(\tau)) = \sum_{(i,j) \in E} \lambda_{ij} \|z_i(\tau) - z_j(\tau) - d_{ij}^d\|^2 + \sum_{R_{ij}} \|u_j(\tau)\|^2\tag{14}$$

where $\lambda_{ij} \geq 0$ and $R_{ij} > 0$ (where $i = 1, \dots, N$) are weights and T is the finite time horizon.

By transforming (13) and (14) to a standard linear quadratic form we obtain (15) and (16) respectively;

$$\|z(T) - z^d(T)\|_{H_f}^2\tag{15}$$

$$\|z(\tau) - z^d(\tau)\|_{G_i}^2 + \sum_{(i,j) \in E} \|u_j(\tau)\|_{R_j}^2 \quad (16)$$

From the previous definitions; $H_{if} = \tilde{Q}_{if} = \tilde{B}\tilde{W}_{if}\tilde{B}^T$, $\tilde{W}_{if} = W_{if} \otimes I_{(2n)}$, $W_{if} = \text{diag}[w_{ij}]$ and $G_i = \tilde{Q}_i = \tilde{B}\tilde{W}_i\tilde{B}^T$, $\tilde{W}_i = W_i \otimes I_{(2n)}$, $W_i = \text{diag}[\lambda_i]$, where H_{if} and G_i are symmetric and positive semi-definite.

The formation cost functions are used to keep the desired distances d_{ij}^d between neighbors, they are suitable models for the followers. The leader must track a desired trajectory z_l^d , while keeping the desired distances d_{ij}^d between its neighbors. In order to accomplish this goal, its cost functions must include a linear-quadratic standard tracking term;

$$\begin{aligned} h_{leader}(T, z(T)) &= \|z(T) - z^d(T)\|_{H_{if}}^2 + \|z_l(T) - z_l^d(T)\|_{h_{if}}^2 \\ &= \|z(T) - z^d(T)\|_{H_{if}^*}^2 \end{aligned} \quad (17)$$

$$\begin{aligned} g_{leader}(\tau, z(\tau), u(\tau)) &= \|z(\tau) - z^d(\tau)\|_{G_i}^2 + \|z_l(\tau) - z_l^d(\tau)\|_{g_l}^2 \\ &\quad + \|u_l(\tau)\|_{R_l}^2 \\ &= \|z(\tau) - z^d(\tau)\|_{G_i^*}^2 + \|u_l(\tau)\|_{R_l}^2 \end{aligned} \quad (18)$$

where $h_{if} = \text{diag}[w_{ij}]$, $H_{if}^* = H_{if} + \text{diag}[0, \dots, h_{if}, \dots, 0]$ and $g_l = \text{diag}[\lambda_l]$, $G_l^* = G_l + \text{diag}[0, \dots, g_l, \dots, 0]$, where H_{if}^* and G_l^* are symmetric and positive semi-definite.

As mentioned before, the leader keeps the desired distance d_{ij}^d between its neighbors in order to maintain the formation. If there is only a desired trajectory requirement, this can be accounted for by assigning 0 (zero) to both H_{if} and G_l .

From the state equations (4) and (5) and the cost functions (13), (14) and (17), (18) it can be seen that the formation control is a linear-quadratic tracking problem. By using error state and control vectors, the formation control is viewed as a linear-quadratic regulating problem with as the state vector and as the control vector in the following presentation.

5.4 Nash Equilibrium and Differential Games

As it is mentioned before, in our case all the vehicles may be regarded as intelligent rational decision-makers. The cost function J_i defined in (12) is known to each robot. To find its strategy, the player i tries to choose a control signal to minimize the cost function. In other words it can be said that the robots in the team need to minimize their cost functions in order to find their controllers. If the cost functions of the players are different, which means that, in our case, not all the players are the leader, one has to find the Nash equilibrium. If for each player i its choice u_i^* is a best response to the other players' choices u_{-i}^* , where $-i$ indicates the set $I \setminus \{i\}$ then a strategy profile $(u_1^*, u_2^*, \dots, u_N^*)$ is a Nash equilibrium. In a Nash equilibrium, no individual can do strictly better by changing its strategy while the others keep their strategies fixed.

A collection of strategies constitutes a Nash equilibrium if and only if the following inequalities are satisfied for all :

$$J_i(u_1^*, \dots, u_{i-1}^*, u_i^*, u_{i+1}^*, \dots, u_N^*) \leq J_i(u_1^*, \dots, u_{i-1}^*, u_i, u_{i+1}^*, \dots, u_N^*), (i = 1, \dots, N). \quad (19)$$

Two types of information structures are interested in differential games: open-loop and state-feedback information structures. In the open-loop information structure, all players make their decisions based on the initial state $z(0)$. Each player computes its equilibrium strategy at the beginning of the game and no state feedback is available during the whole control period. In the state-feedback information structure, all players make their decisions based on the current state $z(t)$. The state-feedback information structure provides more information than the open-loop information structure. Accordingly, the players make more reasonable decisions based on the state-feedback information structure than the open-loop information structure. In this thesis, open-loop differential games are used for computational simplicity and closed loop control is taken into account by casting the problem in a receding horizon structure.

The receding horizon Nash control works as follows;

- 1- At time t and for the current state $z(t)$, solve an optimal control problem over a fixed future interval, say $[t, t+Y-1]$, taking into account the current and future constraints.
- 2- Apply only the first step in the resulting optimal control sequence.
- 3- Measure the state reached at time $t+1$.
- 4- Repeat the fixed horizon optimization at time $t+1$ over the future interval $[t, t+Y-1]$, starting from the current state $z(t+1)$.

The state-dynamic model for each agent can be written as follows to describe the linear quadratic game of N players;

$$\begin{aligned}\dot{z}(t) &= Az(t) + \sum_{i=1}^N B_i u_i \\ &= Az(t) + Bu(t) \\ z(0) &= z_0, t \geq 0\end{aligned}\tag{20}$$

where $B = [B_1, B_2, \dots, B_N]$ and $u = [u_1^t, u_2^t, \dots, u_N^t]^T$.

Under the open-loop information structure of a Nash game, the derivation of open-loop Nash equilibria is closely related to the problem of jointly solving N optimal control problem. According to the minimum principle, the conditions for an open-loop Nash equilibrium for two player games are given in [64]. This result can be generalized straightforward to player games [64]. We use this result here for our formation control problem.

Theorem 5.2: For an N -robot formation control defined as a finite horizon open-loop Nash differential game by (19) and (20), let there exist a solution set $(H_i, i=1, \dots, N)$ to the coupled Riccati differential equations

$$-\dot{H}_i = H_i A + A^T H_i + G_i - H_i \sum_{j=1}^N B_j R_{j,j}^{-1} B_j^T H_j\tag{21}$$

by multiplying both sides of (21) by -1 we obtain;

$$\begin{aligned}\dot{H}_i &= -H_i A - A^T H_i - G_i + H_i \sum_{j=1}^N S_j H_j \\ H_i(T) &= H_{if}\end{aligned}\tag{22}$$

where $S_i = B_i R_{ii}^{-1} B_i^T$. Then, the differential game admits a unique Nash equilibrium solution given by;

$$u_i^*(t) = -R_{i,i}^{-1} B_i^T H_i(t) e_i(t)\tag{23}$$

$$\dot{e}_i(t) = A_{cl}(t) e_i(t)\tag{24}$$

where $A_{cl}(t) = A - \sum_{i=1}^N S_i H_i(t)$. e_i is a vector where the j -th component is $z_i - z_j - d_{i,j}$, e.g.;

for $i=1$ and $d_{1,j} = z_1^d - z_j^d$, $e_1 = [z_1 - z_1 - d_{1,1}, \dots, z_1 - z_j - d_{1,j}, \dots, z_1 - z_N - d_{1,N}]^T$.

For the proof of the theorem, please refer to [64].

The control signal depends on the current error which is defined separately for each vehicle, unlike standard game theory approaches found in the literature. The error vector is different for each vehicle.

Remark 5.1: Due to the stated assumption 5.1 (convexity) and assumption 5.2 (connectivity), the cost functions J_i are strictly convex functions of u_i for all admissible control functions $u_j, i \neq j$ and for all z_0 . This implies that the conditions following from the minimum principle are both necessary and sufficient.

The cost function for the receding horizon approach which the vehicles try to minimize can be written as;

$$j_i(t, z(t), u(t)) = h_i(t+T) + \int_t^{t+T} g_i(\tau, z(\tau), u(\tau)) d\tau \quad (25)$$

where t is the current time and $z(t)$ is the current state vector. The receding horizon approach at each step the control uses $z(t)$ as the initial state vector. The control signal is;

$$u_i^*(t, z(t)) = -R_{i,i}^{-1} B_i^T H_i(t) e_i(t) \quad (26)$$

The existence conditions of the receding horizon Nash control is the same as those of the finite horizon open-loop Nash control, i.e., the receding horizon Nash control exists for every $z(0)$ initial state if and only if matrix $F(T)$ is invertible.

The receding horizon Nash control signal (26) needs the state vector $z(t)$, which includes all states from the formation team. However, the weight parameters w_{ij} and λ_{ij} in the Nash game can be selected as zero for robot if robot is not its neighbor.

This selection will lead to the following matrix form of G_i and H_{ij} :

$$G_i = \begin{bmatrix} g_i^{1,1} & \dots & g_i^{1,j-1} & 0 & g_i^{1,j+1} & \dots & g_i^{1,N} \\ & & & \dots & & & \\ g_i^{j-1,1} & \dots & g_i^{j-1,j-1} & 0 & g_i^{j-1,j+1} & \dots & g_i^{j-1,N} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ g_i^{j+1,1} & \dots & g_i^{j+1,j-1} & 0 & g_i^{j+1,j+1} & \dots & g_i^{j+1,N} \\ & & & \dots & & & \\ g_i^{N,1} & \dots & g_i^{N,j-1} & 0 & g_i^{N,j+1} & \dots & g_i^{N,N} \end{bmatrix}$$

$$H_{j_f} = \begin{bmatrix} h_{j_f}^{1,1} & \dots & h_{j_f}^{1,j-1} & 0 & h_{j_f}^{1,j+1} & \dots & h_{j_f}^{1,N} \\ & & & \dots & & & \\ h_{j_f}^{j-1,1} & \dots & h_{j_f}^{j-1,j-1} & 0 & h_{j_f}^{j-1,j+1} & \dots & h_{j_f}^{j-1,N} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ h_{j_f}^{j+1,1} & \dots & h_{j_f}^{j+1,j-1} & 0 & h_{j_f}^{j+1,j+1} & \dots & h_{j_f}^{j+1,N} \\ & & & \dots & & & \\ h_{j_f}^{N,1} & \dots & h_{j_f}^{N,j-1} & 0 & h_{j_f}^{N,j+1} & \dots & h_{j_f}^{N,N} \end{bmatrix}$$

where $g_i^{u,v}$ or $k_{j_f}^{u,v}$ is a block with size $2m \times 2m$. G_i and H_{j_f} has $N \times N$ blocks. The j th block row or column consists of m zero blocks. It should be noted that matrix A has a block diagonal structure. Based on these matrix structures, it can be found the j th block row of H_i solution consists of zero blocks from the coupled Riccati differential (22) and the following below;

$$H_i = \begin{bmatrix} h_i^{1,1} & \dots & h_i^{1,j-1} & h_i^{1,j} & h_i^{1,j+1} & \dots & h_i^{1,N} \\ & & & \dots & & & \\ h_i^{j-1,1} & \dots & h_i^{j-1,j-1} & h_i^{j-1,j} & h_i^{j-1,j+1} & \dots & h_i^{j-1,N} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ h_i^{j+1,1} & \dots & h_i^{j+1,j-1} & h_i^{j+1,j} & h_i^{j+1,j+1} & \dots & h_i^{j+1,N} \\ & & & \dots & & & \\ h_i^{N,1} & \dots & h_i^{N,j-1} & h_i^{N,j} & h_i^{N,j+1} & \dots & h_i^{N,N} \end{bmatrix}$$

Therefore, the receding horizon Nash control $u_i^*(t, z(t))$ (26) does not need the state $z_j(t)$ from nonneighbor robot j . If there is more than one robot in the team, which are not the neighbors of robot i , the same conclusion can be made. Thus, $u_i^*(t, z(t))$ is a distributed control law.

$H_i(t)$ is calculated as follows; Let δ denote the control time interval and $0 \leq \delta \leq T$. Each robot computes an open-loop Nash equilibrium solution for the period $t \leq \delta \leq t+T$. The algorithm uses this solution to control robots for the period $[t, t+\delta]$. At the next time instant $t+\delta$, this procedure repeats.

5.5 NSB Behavioral Control

An autonomous robot, needs to achieve several goals at the same time; each goal may require a robot movement different from the others, i.e., the goals may conflict one with the other and a strategy to handle this situation is needed. The most common approach is to assign a relative importance among them. Moreover, the importance of a goal is often context-dependent; for example, obstacle avoidance of a static, far, obstacle might be of smaller importance with respect to the goal of reaching a close target position.

In a static environment the task is achieved when its output is constant at a value that minimizes the task function. In presence of multiple behaviors, each task output may achieve its specific goal, but there is no guarantee that a single command to the robot can accomplish all the assigned behaviors at the same time. In particular, when a motion command cannot reduce simultaneously all the task functions there is a conflict among the tasks that must be solved by a suitable policy.

For example, if the task output is a velocity command to a mobile robot, to reach a given goal position a distance from goal task function can be considered; the velocity command will then be generated so as to reduce the distance between the vehicle and the goal and it will be null when the goal position is reached. If, in addition, an obstacle must be avoided, another velocity command will be generated so as to increase the distance between the vehicle and the obstacle; this command will be null when the obstacle is considered out of reach. In this scenario, when the obstacle is somewhere along the line of sight of the goal position from the robot the two behaviors come in conflict: in fact, the two individual-task velocity commands will counteract and the vehicle can either approach the goal position (and come closer to the obstacle too) or escape the obstacle (and drive away from the goal position too).

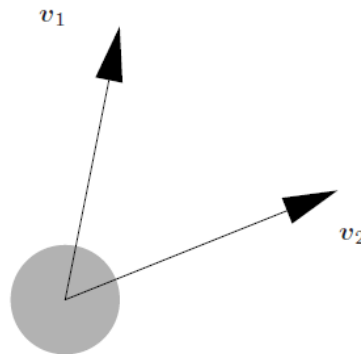


Figure 5.3: A case of two simultaneous behaviors: v_1 is the output of an obstacle avoidance task while v_2 is the output of a trajectory tracking task.

To handle conflicts among different task functions, NSBBC approach is used. The NSB Behavioral Control strategy is based on degrading the general obstacle avoidance process into smaller problems which are less complex (19), (20).

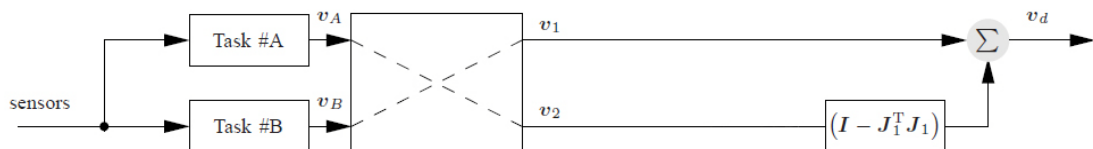


Figure 5.4: Sketch of the null-space-based behavioral control in a 2-task example.

Each task velocity is computed as if it were acting alone; then, before adding its contribution to the overall vehicle velocity, a lower-priority task is projected onto the

null space of the immediately higher-priority task so as to remove those velocity components that would conflict with it.

Before being added to the higher-priority Task 1 velocity, the lower-priority Task 2 velocity must be projected onto the null space of the primary-task Jacobian matrix. If the primary task is specified so as to ensure a minimum given distance from the obstacle (i.e., is expressed as a scalar function), its Jacobian has a monodimensional range space aligned to the vehicle-obstacle direction and a monodimensional null space along the orthogonal direction. Therefore, by projecting the secondary task velocity onto the null space of J_1 , the component of v_2 that would affect the vehicle obstacle distance is filtered out; on the other hand, the other component of v_2 results in a slide around the obstacle motion.

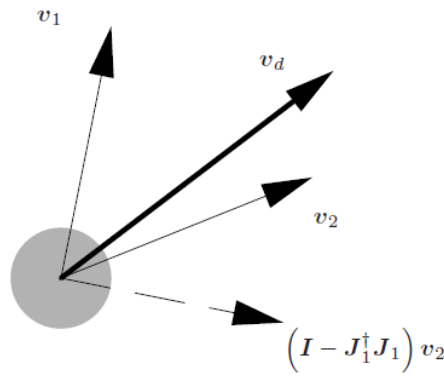


Figure 5.5: Task velocity composition of the null-space-based behavioral control for solving the problem sketched in Figure 5.3.

A brief review of the method is presented here for clarity's sake. Let us consider a generic manipulator; the end effector velocity can be written as a function of the joints variables q set as;

$$\dot{x}_e = J_e(q) \cdot \dot{q} \quad (27)$$

From (27), it can be computed the joints velocities \dot{q} required, in order for the end effector to move at a given speed as shown in (28),

$$\dot{q} = J_e^\dagger(q) \cdot \dot{x}_e \quad (28)$$

where $J_e^\dagger(q)$ is Jacobian pseudo inverse.

In general, the additional degrees of freedom are used to obtain a solution of (28) that minimizes some joint velocity norm. It is possible, however, to get a non minimum norm solution, and to use the redundancy for other objectives. A possible non minimum norm solution that uses the Jacobian pseudo inverse $J_e^\dagger(q)$ is given by:

$$\dot{q} = J_e^\dagger(q) \cdot \dot{x}_e + (I - J_e^\dagger(q) \cdot J_e(q)) \cdot \dot{q}_a \quad (29)$$

In (29), \dot{q}_a is projected on $\text{Ker}(j_e)$. The end effector's velocity vector \dot{q} has a component that allows the effector to follow a prescribed trajectory. The additional component \dot{q}_a does not have any effect on the motion. This procedure can be used to compute \dot{q}_a to yield a hierarchical structure subdivided into tasks.

The above mentioned property can be generalized from the case of redundant manipulators, to problems requiring dividing an objective into tasks having different priorities, such as the motion control of an autonomous vehicle within an environment with obstacle(s) (19).

In our case, the robot configuration q can be replaced by the vehicle position, joint velocities vector \dot{q} may be used to represent the vehicle planar velocity, the end effector position x_e becomes the variable to be regulated to a desired value, and the Jacobian matrix, as for robots, describes the vector space of allowed velocities. Thus (28) becomes;

$$\dot{p}_d = v_d = J^\dagger(p_d) \cdot \dot{\sigma} \quad (30)$$

where $p_d = p_d(t)$ is the time dependent desired position at some time t , v_d the vehicle's desired velocity, J the matrix of velocities allowed for the task, and σ the task variable that must be controlled. Equation (30) is modified with the addition of a control term for closed loop position error elimination; thus

$$\begin{aligned} \dot{p}_d &= v_d = J^\dagger(p_d) \cdot (\dot{\sigma} + \Lambda \cdot \tilde{\sigma}) \\ \tilde{\sigma} &= \bar{\sigma} - \sigma \end{aligned} \quad (31)$$

where $\bar{\sigma}$ is the desired value of the task variable, $\tilde{\sigma}$ is the tracking error and Λ is a positive definite gain matrix. So, σ and J can be used to define tasks appropriately, and then to compute the speed necessary to accomplish them.

In order to avoid a collision, the obstacle avoidance task affects the vehicle's velocity along the direction toward the obstacle. The magnitude of task velocity is a function of the distance to the obstacle and it becomes zero in the proximity of obstacle edge (note that we are assuming a 2D motion space in this paper). Should the vehicle go beyond the circle's limit, the speed sign is reversed.

$$\begin{aligned} J_0 &= \left(\frac{p - p_0}{\|p - p_0\|} \right)^T \\ \sigma_0 &= \|p - p_0\| \\ \bar{\sigma}_0 &= d \end{aligned} \quad (32)$$

where p is the vehicle position, p_0 is the position of the obstacle, J_0 represents the velocity direction on which the task can operate, σ_0 is the distance between leader vehicle and obstacle, and $\bar{\sigma}_0$ is the safety circle radius. Thus;

$$v_0 = J_0^\dagger \cdot \Lambda_0 \cdot (d - \|p - p_0\|) \quad (33)$$

Since any velocity direction must be able to be imposed by the task, J_g is a 2x2 identity matrix, which identifies all possible velocities on the plane. The planar position is the task variable σ_g , which must reach the desired value $\bar{\sigma}_g$. Therefore:

$$\begin{aligned} J_g &= I \\ \dot{\sigma}_g &= u_i^* \end{aligned} \quad (34)$$

and:

$$v_g = u_l^* \quad (35)$$

The matrix J_g is full rank, thus its null space is empty. In the case of a single obstacle, the main rule is that if the desired position of the vehicle is inside the safety circle, then the primary task becomes the obstacle avoidance task and following the trajectory becomes the secondary task. Otherwise, the velocity is calculated using (26). When the obstacle avoidance is the primary task, the velocity of the vehicle is calculated via projecting in the null space of the obstacle avoidance task by using the following equation;

$$u_l = v_0 + (I - J_0^T J_0) v_g \quad (36)$$

6. STABILITY ANALYSIS

6.1 Stability Analysis for Receding Horizon Nash Control

Based on the theorem 5.2, the solvability of the coupled Riccati differential equation (22) is vital to the finite horizon Nash equilibrium solution. In the following, a necessary and sufficient condition is established for the solvability of the coupled Riccati differential equations.

Let us define;

$$M = \begin{bmatrix} A & -S_1 & \dots & -S_N \\ -Q_1 & -A^T & 0 & 0 \\ & \dots & & \\ -Q_N & 0 & 0 & -A^T \end{bmatrix}$$

and

$$F(T) = \begin{bmatrix} I_{2Nm} & 0 & \dots & 0 \end{bmatrix} e^{-MT} \begin{bmatrix} I_{2Nm} \\ H_{1f} \\ \dots \\ H_{Nf} \end{bmatrix}$$

In reference [64] it is given an approach to judge if the solution exists for two-player games. This result can be generalized straightforward to N player games [64]. Based on this theorem with N players, the formation control problem has the following result.

Theorem 5.3: For a N -robot finite horizon formation control defined as a finite horizon open-loop Nash differential game by (19) and (20), the coupled Riccati differential (22) has a solution for every initial state z_0 on $[0, T]$ if and only if matrix $F(T)$ is invertible.

Proof 5.1: For the formation control of the multiple robot systems (20), it is known that $[A, B]$ is stabilizable. As the Laplacian is symmetrical and positive semi-definite $\tilde{Q}_i \geq 0$ and $\tilde{Q}_f \geq 0$, the symmetrical Riccati differential equations

$$\dot{P}_i = -A^T P_i - P_i A - G_i + P_i S_i P_i, P_i(T) = H_f \quad (37)$$

have a symmetrical solution P_i on $[0, T]$ for all $i=1, \dots, N$. This result combining with that $F(T)$ is invertible proves the coupled Riccati differential equations (22) has a solution for every initial state z_0 on $[0, T]$, as indicated in [64].

Remark 5.2: The matrix M consists of $(N+1) \times (N+1)$ blocks. Also e^{-MT} has the same block structure. Denoted by $W_{ij}(T)$ as the ij th block of e^{-MT} , we have $F(T) = W_{11}(T)$. The invertibility of $F(T)$ depends on M and T . It has been shown in [64] and [70] that different T leads to different invertibility of $F(T)$. In the finite receding horizon Nash control, T is the length of control horizon. The selection of T in the finite receding horizon control should guarantee that $F(T)$ is invertible.

The formation control defined as a finite horizon Nash differential game (20), (25) has a receding horizon Nash control for every initial state z_0 if and only if $F(T)$ matrix is invertible. As long as all the eigenvalues of $A_{ci}(0)$ have negative real parts, the receding horizon Nash control is asymptotic stable.

6.2 Stability Analysis for NSBBC Obstacle Avoidance Algorithm

The control law for NSBBC obstacle avoidance is;

$$v_0 = J_0^\dagger \cdot \Lambda_0 \cdot (d - \|p - p_0\|) = 0 \quad (38)$$

J_0 is 1 in every point of the plane except over the obstacle; Λ_0 is a value different from 0. In a polar representation centered on the obstacle, the control law becomes;

$$v_0(p, \varphi) = \hat{p} \cdot \Lambda_0 \cdot (r - p) = 0 \quad (39)$$

and it vanishes when $r = p \quad \forall \varphi$

Applying Lyapunov we obtain;

$$V = \frac{1}{2} \cdot (r - p)^2 \quad (40)$$

$$\dot{V} = (r - p)(-\dot{p}) = -\Lambda_0 \cdot (r - p)^2 < 0 \quad \forall \varphi, p \neq 0 \quad (41)$$

The safety circle is the place of the points of equilibrium of the obstacle avoidance task. This means that the vehicle is driven towards the safety circle and stops once it reaches. All points of the safety circle are therefore stable equilibrium points.

7. SIMULATIONS

In this chapter the results of various simulations are provided to justify the model and solutions.

For all the simulations, the formation consists of $N = 10$ unmanned vehicles moving on a plane. The underlying graph structure is shown in Fig. 1.

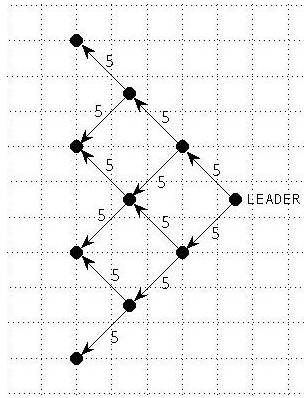


Figure 7.1: Triangle formation shape.

Agent 1 is assigned as the leader of the formation and the rest of the agents, are the followers. The geometry of the formation and the weights between the neighbors which are used to compute the cost functions (15) and (16) are shown in Fig. 7.1.

The tracking weight matrix of the leader agent is selected as;

$$h_{1f} = g_1 = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

The weight parameters are selected the same for all agents, that is $R_{ii} = I_{2 \times 2}$ for $i = 1, 2, 3, \dots, 10$. The value of Λ_0 is selected as 0.01. The finite horizon length $T = 10$ s and sample time is 0.1 s.

In the first simulation, it is considered the problem of formation keeping and obstacle avoidance for a triangle shaped formation, which must follow a rectilinear trajectory. The leader agent makes use of (17) and (18) in order to calculate its cost function which includes both tracking and formation costs. The followers just make use of (15) and (16) in order to find their cost functions, which only includes formation cost.

The phase plane history of the formation where just the leader vehicle executes the obstacle avoidance algorithm is shown in Fig. 7.2 and Fig.7.5. The leader vehicle must set the radius of the obstacle, which is d in (32), such that all the vehicles in the formation avoid the obstacle. In Fig.7.2, the radius d is 3 times the real radius of the obstacle. In Fig.7.5, the radius d is equal to the real radius of the obstacle. It is seen that in both cases the triangle shaped formation is maintained during the entire mission. But, in Fig.7.5. it is seen that not all the vehicles in formation can avoid the obstacle. The reason why some of the followers cannot avoid the obstacle is that, the leader vehicle does not take care about the followers if they can avoid the obstacle or not. Since the followers do not execute the obstacle avoidance algorithm, they cannot avoid the obstacle perfectly.

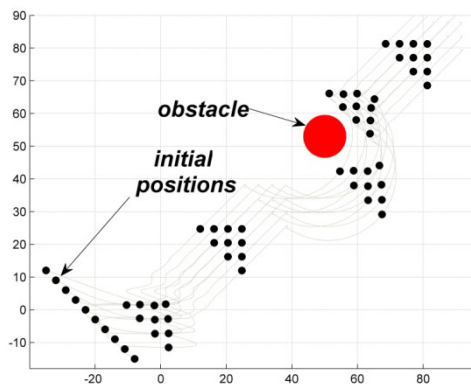


Figure 7.2: Phase plane of unmanned vehicles where the radius d is 3 times the real radius of the obstacle.

In Fig.7.3 and Fig.7.4, it is shown the error between the desired trajectory and the trajectory followed by the agents for the formation which just the leader vehicle executes the obstacle avoidance algorithm. From these figures, it can be seen that the errors are almost the same for all the vehicles. This means that the formation is almost maintained during entire mission.

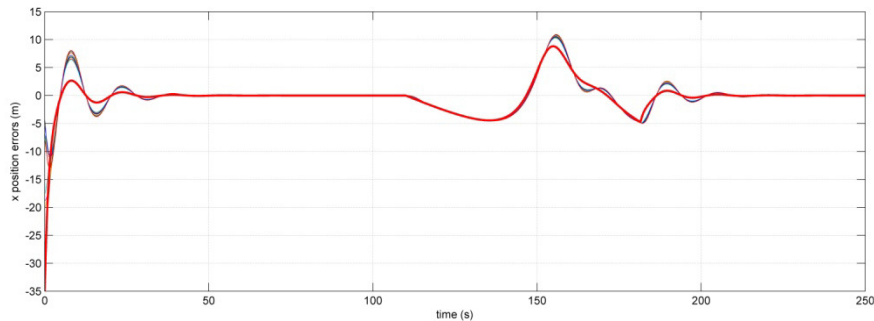


Figure 7.3: X position errors of the unmanned vehicles for the simulation shown in Fig.7.2

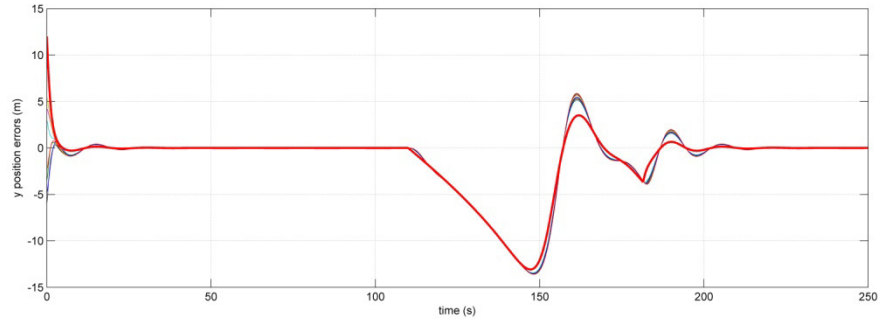


Figure 7.4: Y position errors of the unmanned vehicles for the simulation shown in Fig.7.2

In Fig.7.5 it is given the simulation results where the leader vehicle does not take care about the followers if they can avoid the obstacle or not. Since the followers do not execute the obstacle avoidance algorithm, some of the followers cannot avoid the obstacle.

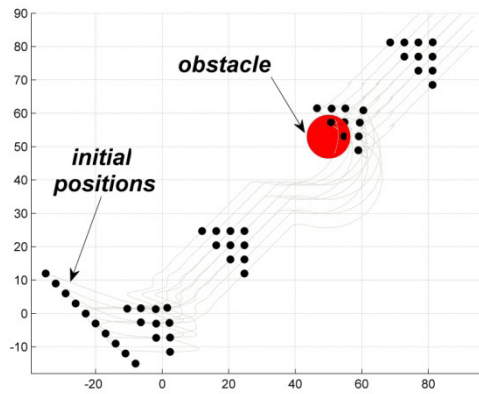


Figure 7.5: Phase plane of unmanned vehicles where the radius d is equal to the real radius of the obstacle.

In Fig. 7.6, it is seen the phase plane history of the formation where all the vehicles execute the obstacle avoidance algorithm.

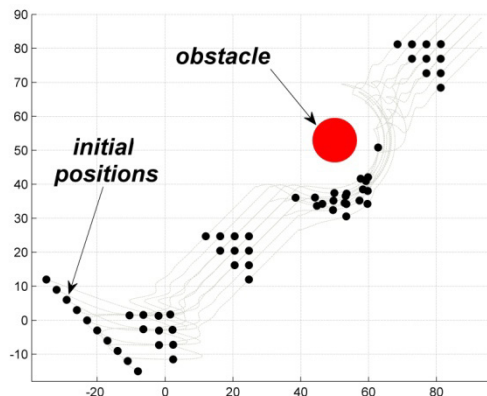


Figure 7.6: Phase plane of unmanned vehicles where all the vehicles execute obstacle avoidance algorithm.

In Fig.7.6 it is seen that, the formation is not maintained while the vehicles avoid the obstacle. The reason of this behaviour is that the velocity vector computed by (26) for all

the agents is changed because of the presence of the obstacle; moreover this change depends on the position of each vehicle. Since the position of each vehicle is different, the obstacle affects differently each agent, causing the formation break up.

In Fig.7.7 and Fig.7.8, it is shown the error between the desired trajectory and the trajectory followed by the agents of the formation which all the vehicles execute the obstacle avoidance algorithm. From these figures, it is seen that the errors are different for each vehicle. This means that the formation is broken up.

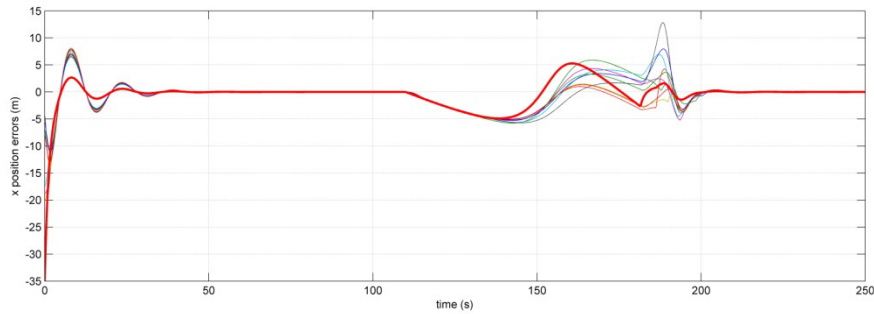


Figure 7.7: X position errors of the unmanned vehicles for the simulation shown in Fig.7.6

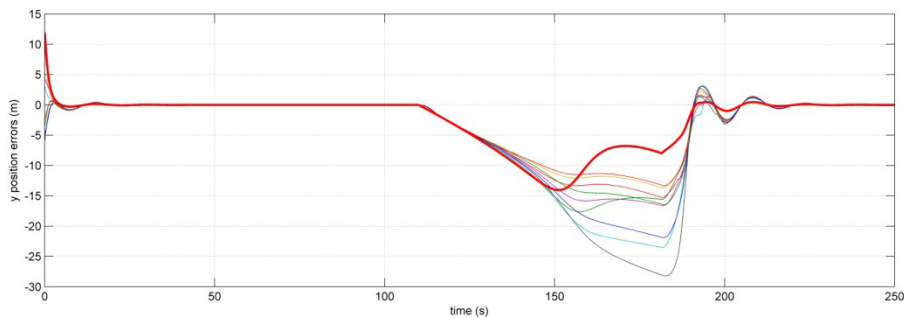


Figure 7.8: Y position errors of the unmanned vehicles for the simulation shown in Fig.7.6

In above provided simulations results, it is always considered an immobile obstacle. But also it must be considered the problems where the obstacle may be a mobile one or a suddenly appearing one i.e. a big stone falling from a hill, a big hole may be formed suddenly, or any unexpected and unknown prior to the start of the mission.

In the simulations results shown in Fig.7.9., it is considered a moving small abstacle towards the formation of the unmanned vehicles and the related figures are given to illustrate the necessary information. In fig. 7.9 different phases of the mission is provided to be able to illustarate the simulations results beter and understand better the scenario. Same thecnique is used for the similar scenarios.

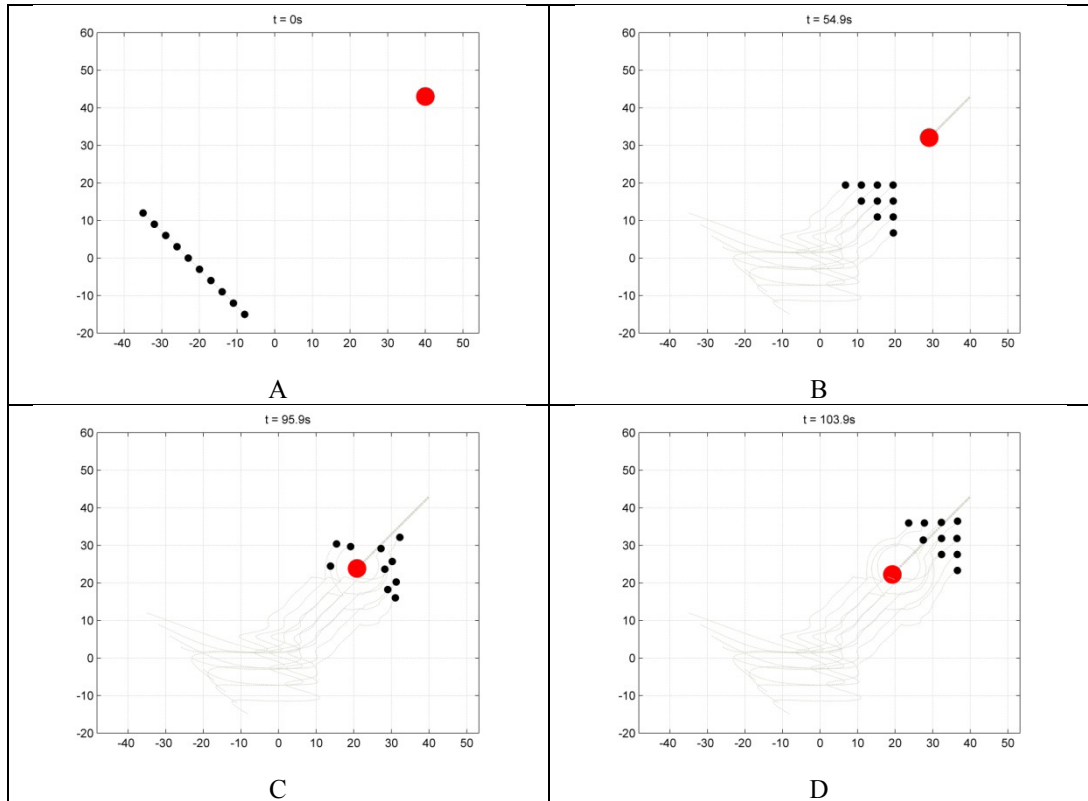


Figure 7.9: Phase plane of unmanned vehicles with a moving small obstacle which moves toward the vehicles where all the vehicles execute the obstacle avoidance algorithm.

In Fig. 7.10 and Fig. 7.11 the position errors of the unmanned vehicles on X and Y axis are provided respectively.

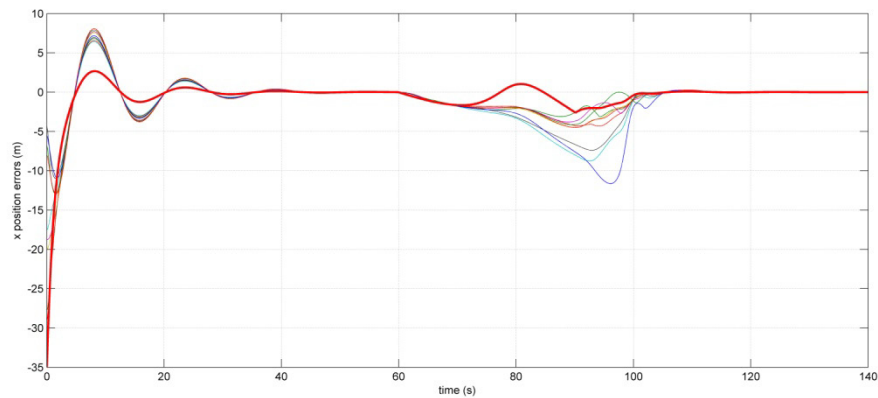


Figure 7.10: X position errors of the unmanned vehicles for the simulations shown in Fig.7.9

As it can be seen from Fig. 7.10 and 7.11, the errors for each vehicle are different and this means that all the agents execute the obstacle avoidance algorithm and the obstacle affects differently each agent. As a result of this, X and Y errors for each agent are different and the formation is not maintained during the obstacle avoidance algorithm.

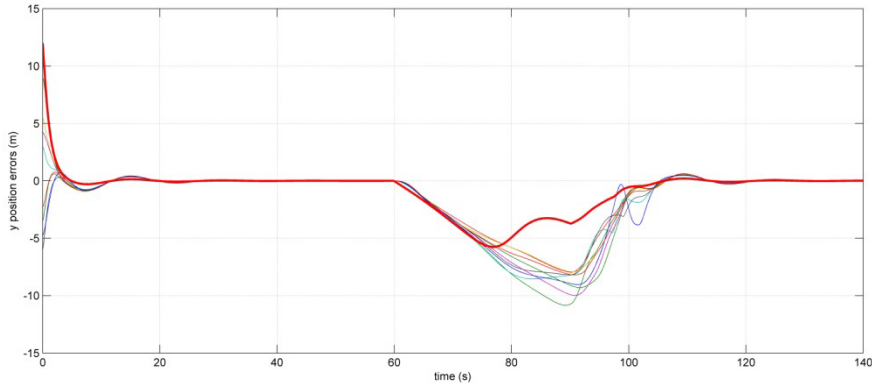


Figure 7.11: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.9

In Fig. 7.12 it is provided the control signals which affect the leader vehicle.

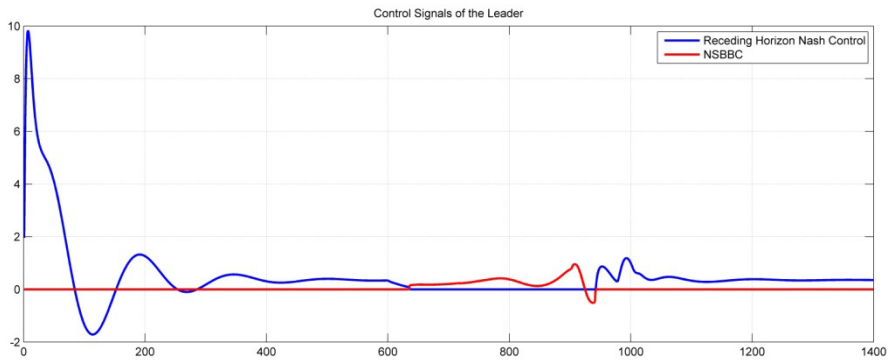


Figure 7.12: Control signals that affect the leader vehicle for the simulations shown in Fig.7.9

As it can be seen from Fig.7.12, while the Receding Horizon Nash Control is used as the control signal, the NSBBC control signal is not active. During the obstacle avoidance process, the NSBBC control signal is used and Receding Horizon Nash Control is not active as the control signal. As soon as the obstacle is avoided, again Receding Horizon Nash Control becomes active. This comment is valid for all the scenarios provided below.

In the following simulation, again it is considered a moving small abstacle but this time it moves across the trajectory of the formation of the unmanned vehicles. Again, in Fig.7.13, it is given different pahases of the simualtions results to be able to illustarate the scenario in an efficient manner. As it can be seen from the figures, the vehicles are also able to avoid this kind of moving obstacle. While avoiding the obstacle, the formation is broken up and as soon as the obstacle is avoided, or in other words as soon as the primary task becomes the trajectory tracking task, they again find the geometry of the formation and keep tracking the desired tejectory with zero error.

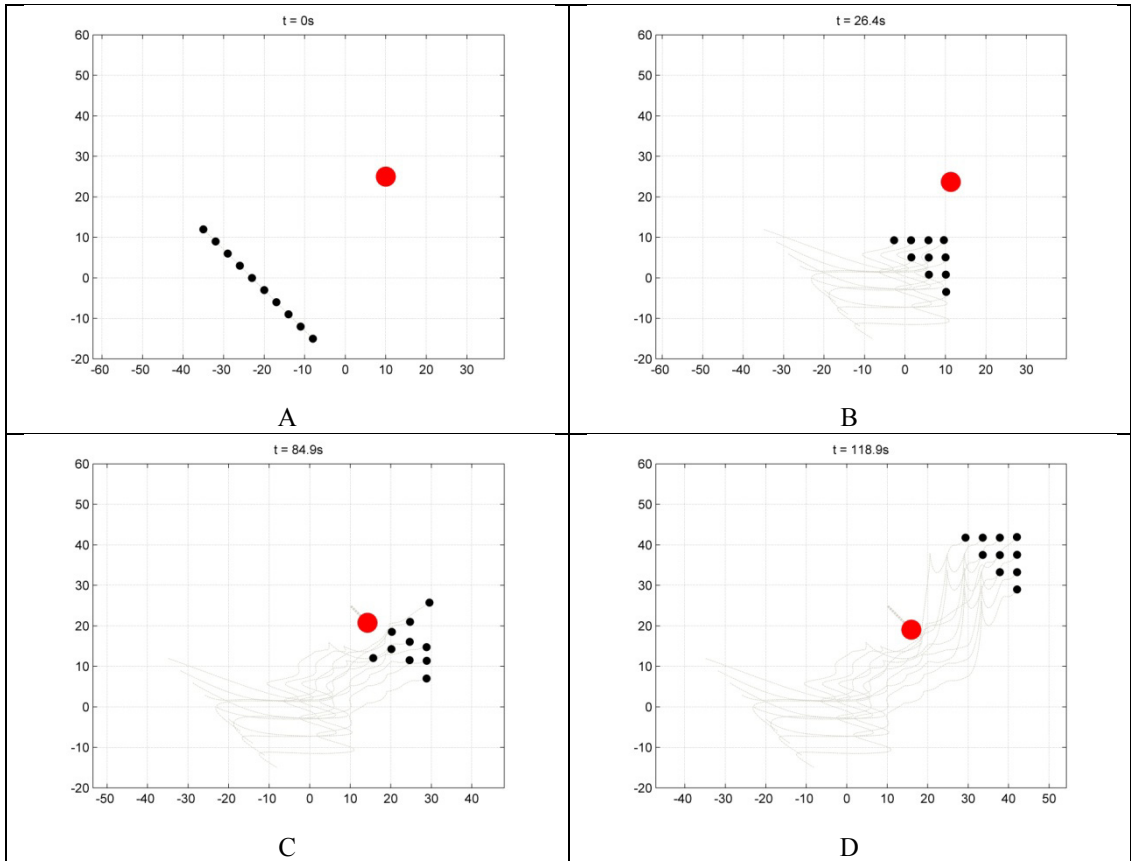


Figure 7.13: Phase plane of unmanned vehicles with a moving small obstacle which crosses the trajectory of the vehicles where all the vehicles execute the obstacle avoidance algorithm.

In Fig. 7.14 and Fig. 7.15 the position errors of the unmanned vehicles on X and Y axis are provided respectively.

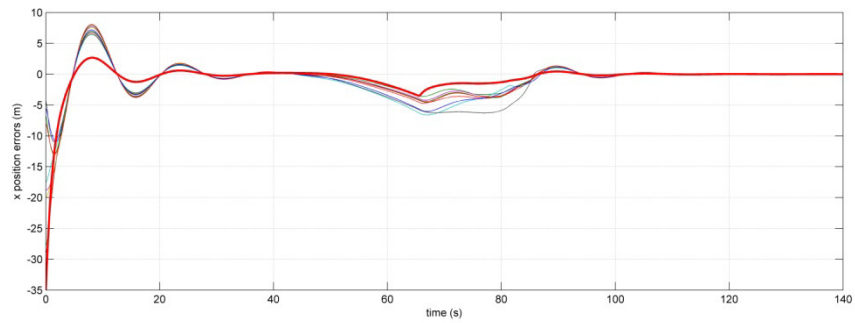


Figure 7.14: X position errors of the unmanned vehicles for the simulations shown in Fig.7.13

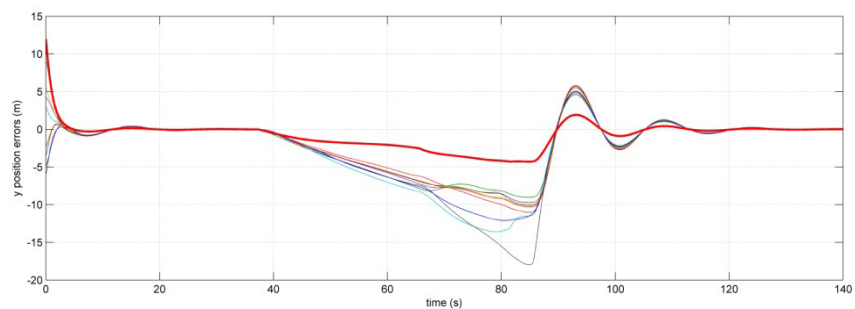


Figure 7.15: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.13

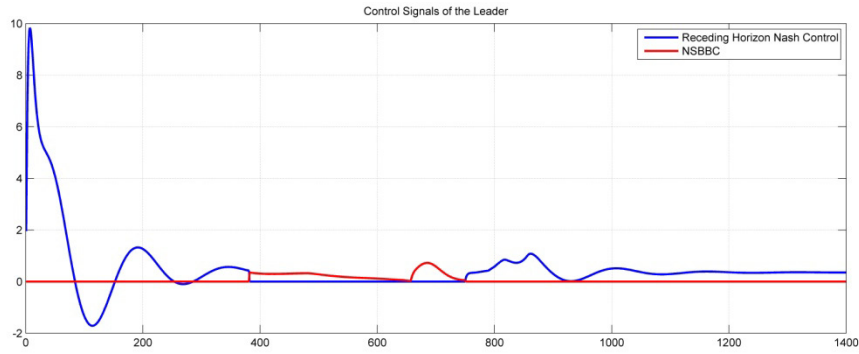


Figure 7.16: Control signals that affect the leader vehicle for the simulations shown in Fig.7.13

In Fig.7.16 it is provided the control signals which affect the leader vehicle.

In Fig.7.17, we provide the simulations results where the obstacle moves on the same direction with the formation of the unmanned vehicles.

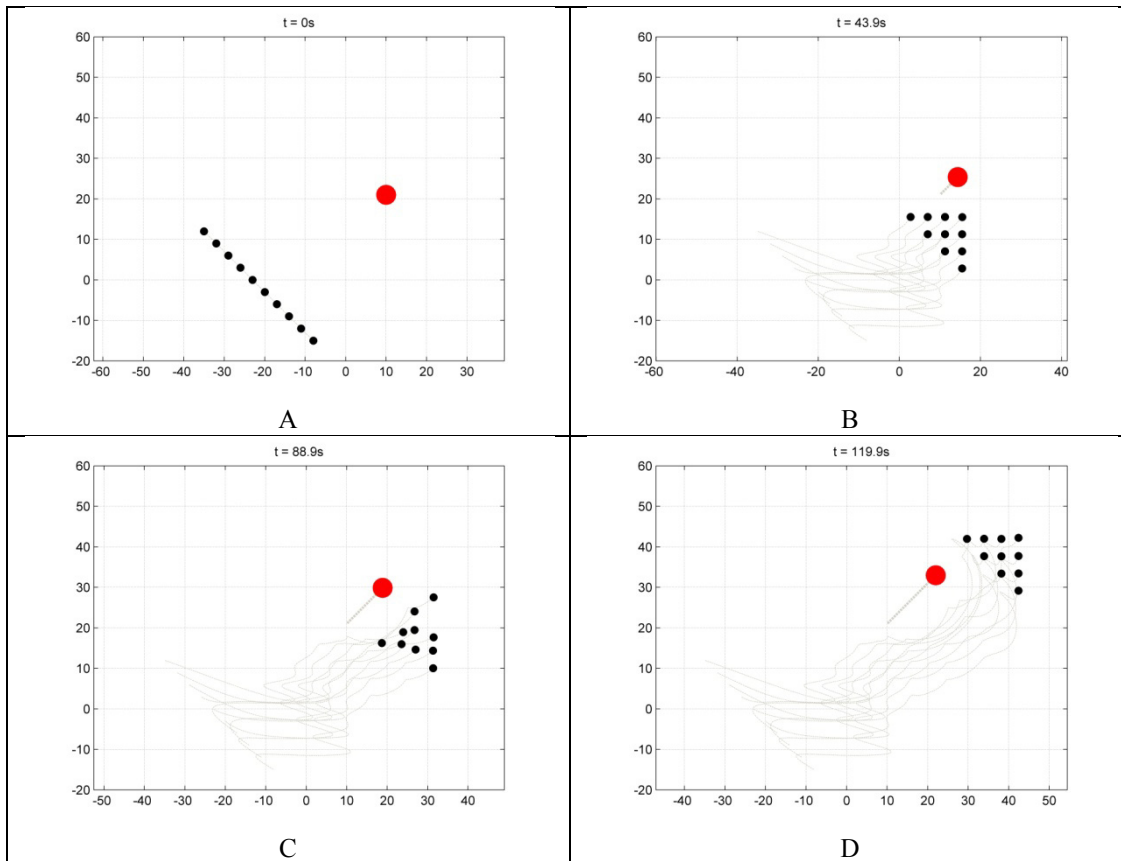


Figure 7.17: Phase plane of unmanned vehicles with a moving small obstacle which moves on the same direction with the vehicles where all the vehicles execute the obstacle avoidance algorithm.

In Fig. 7.18 and Fig. 7.19 the position errors of the unmanned vehicles on X and Y axis are provided respectively. Since the errors for each vehicle is different, it is understood that all the vehicles are executing the obstacle avoidance algorithm.

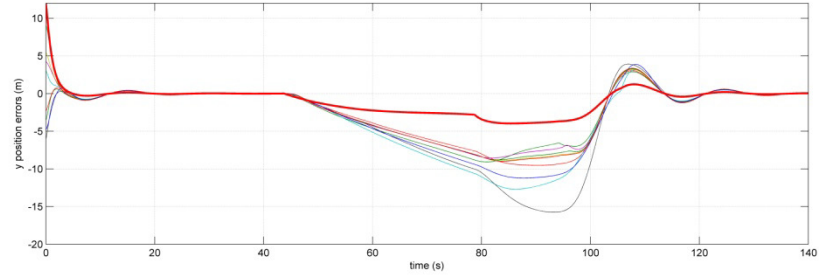


Figure 7.18: X position errors of the unmanned vehicles for the simulations shown in Fig.7.17

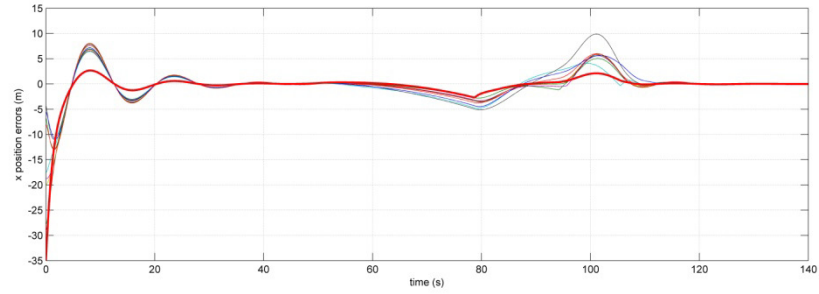


Figure 7.19: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.17

In Fig. 7.20 it is provided the control signals which affect the leader vehicle.

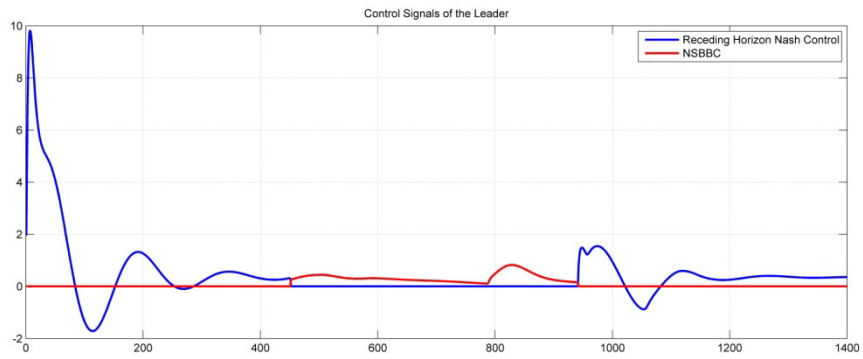


Figure 7.20: Control signals that affect the leader vehicle for the simulations shown in Fig.7.17

In Fig.7.21, the simulations results are seen where the small obstacle again moves towards the formation of the unmanned vehicles.

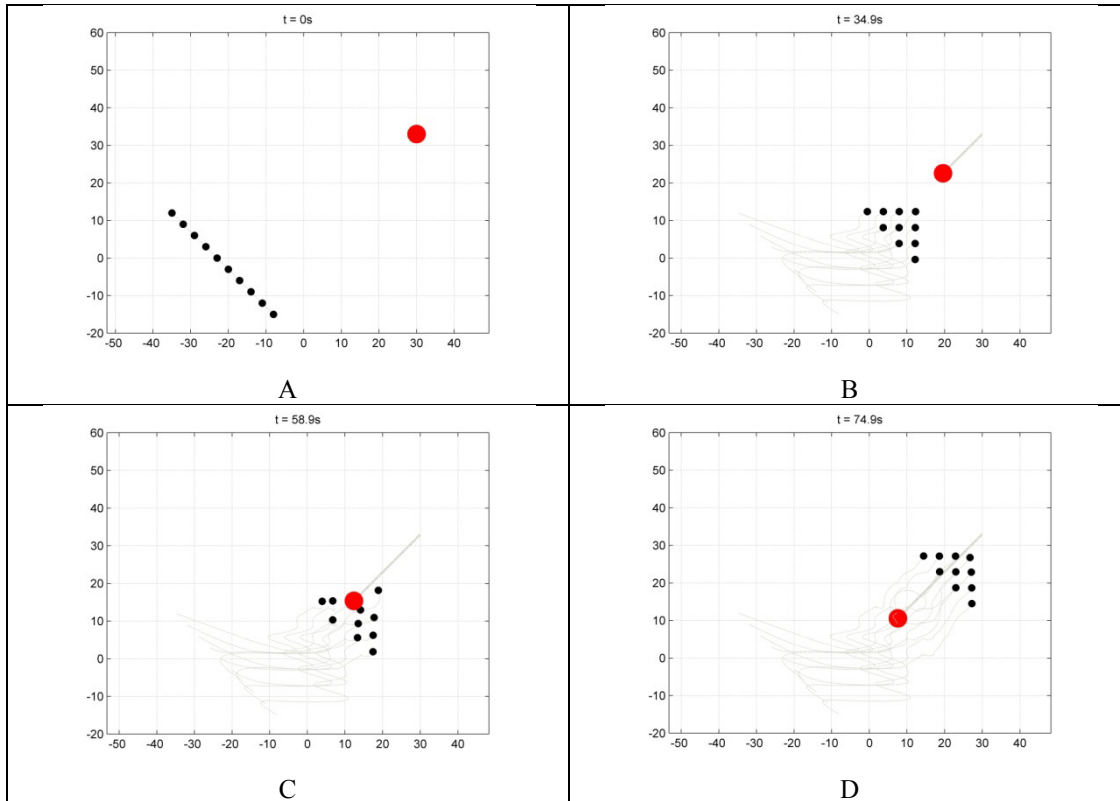


Figure 7.21: Phase plane of unmanned vehicles with a moving small obstacle which moves toward the vehicles where all the vehicles execute the obstacle avoidance algorithm and one of the vehicles slightly collides with the obstacle.

As it is seen from the Fig. 7.21 (C), one of the unmanned vehicles slightly collides with the obstacle and cannot avoid it perfectly. The reason of this imperfection is related with the diameter of the obstacle. The results for the same scenario but with a bigger obstacle is given in the Fig. 7.25.

The position errors on X axis and Y axis of the vehicles are provided in the Fig.7.22 and Fig.7.23.

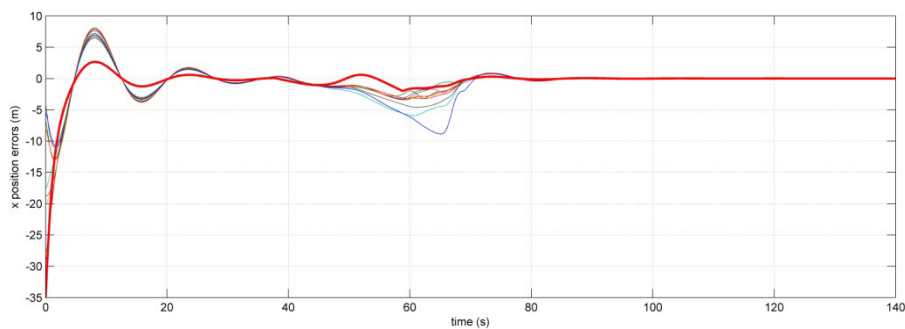


Figure 7.22: X position errors of the unmanned vehicles for the simulations shown in Fig.7.21

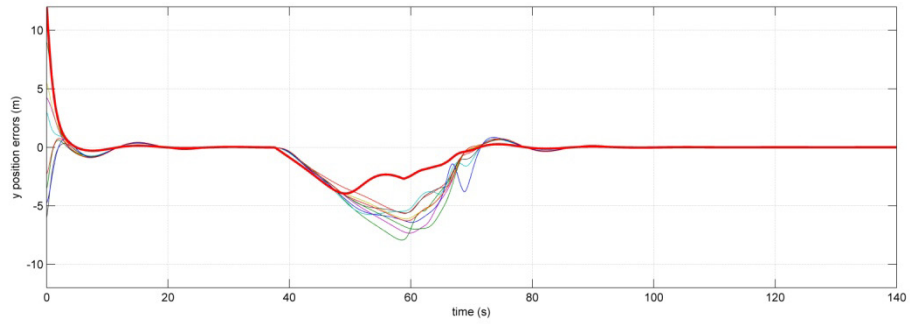


Figure 7.23: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.21

In Fig. 7.24 it is provided the control signals which affect the leader vehicle.

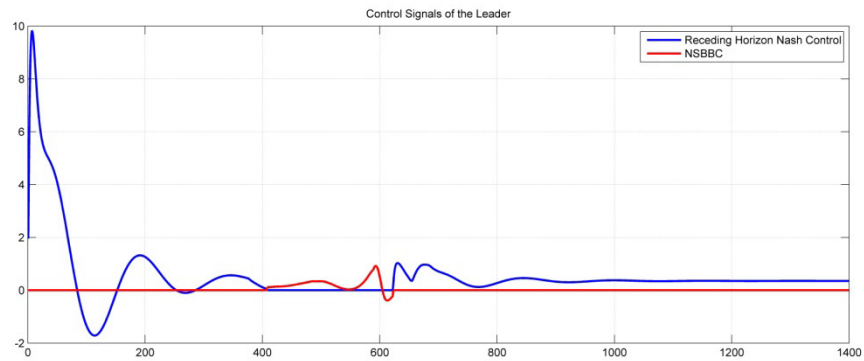


Figure 7.24: Control signals that affect the leader vehicle for the simulations shown in Fig.7.21

In the next simulation, Fig. 7.25, the same scenario which was shown in Fig. 7.21 is simulated but instead of a small obstacle, this time it is simulated with a bigger obstacle. As a result it is seen that when the obstacle is bigger, all the agents can avoid the obstacle perfectly. In the previous simulations it is seen that when the obstacle was small, one of the unmanned vehicles could not avoid it and slightly collides with the obstacle. This result verifies that the diameter of the obstacle is an important parameter for the unmanned vehicles to avoid the obstacle and the obstacle avoidance algorithm is directly related with the diameter of the obstacle.

In Fig. 7.26 and Fig. 7.27 the position errors of the unmanned vehicles on X axis and Y axis are provided respectively. In Fig. 7.28 it is seen that when the primary task is trajectory tracking, the control signal from receding horizon Nash control is active and the control signal from NSBBC obstacle avoidance is passive. When the vehicles realize the obstacle, then the primary task becomes obstacle avoidance algorithm and this time the control signal from NSBBC obstacle avoidance becomes active and the control signal from receding horizon Nash control becomes passive. During the whole obstacle avoidance period this remains the same but as soon as the obstacle is avoided, again the primary task becomes trajectory tracking and control signal from receding horizon Nash control becomes active.

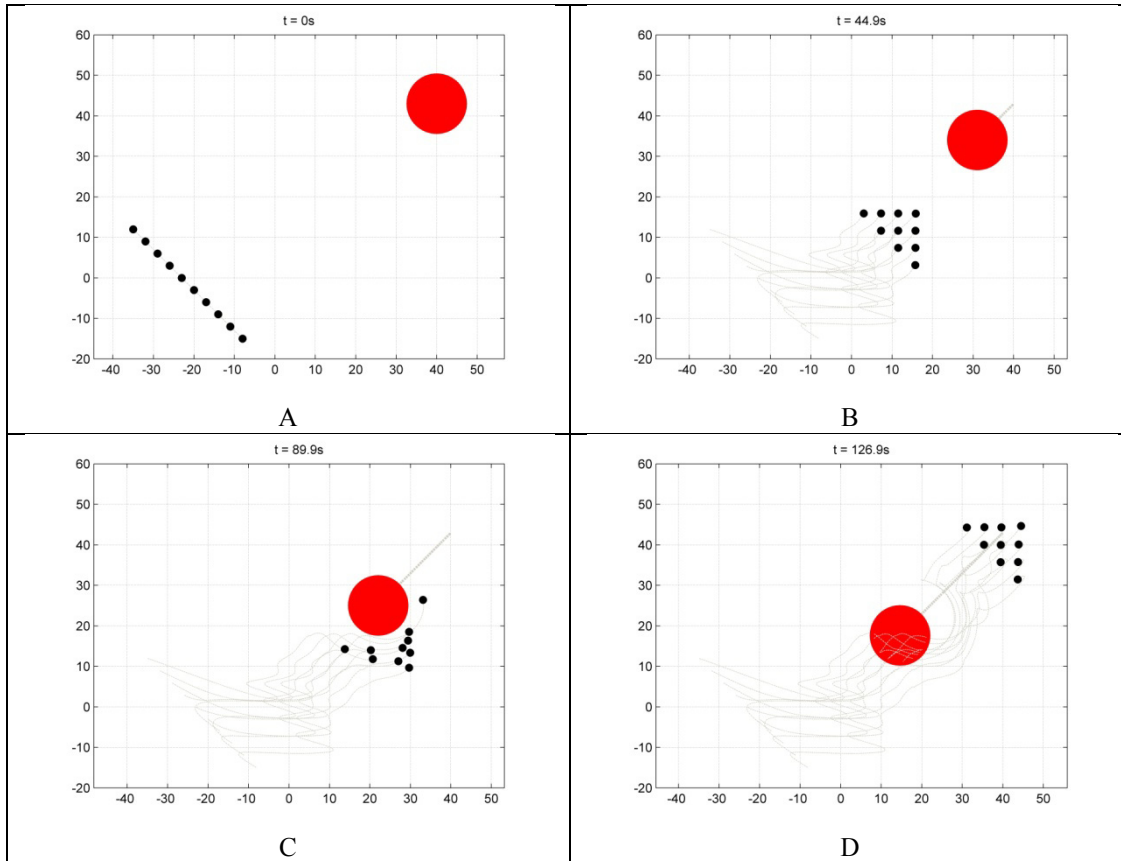


Figure 7.25: Phase plane of unmanned vehicles with a moving bigger obstacle which moves toward the vehicles where all the vehicles execute the obstacle avoidance algorithm.

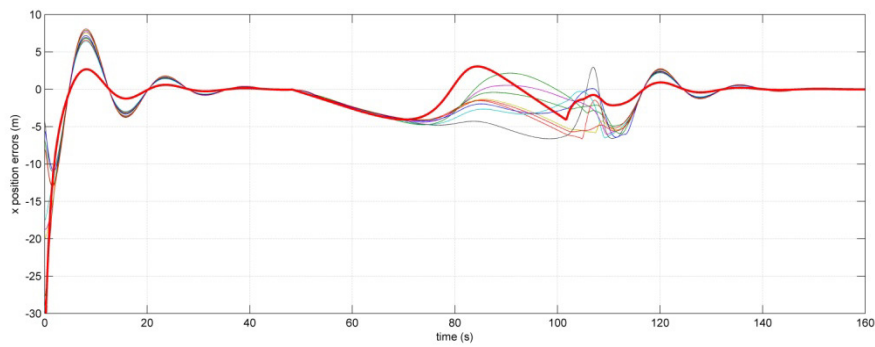


Figure 7.26: X position errors of the unmanned vehicles for the simulations shown in Fig. 7.25

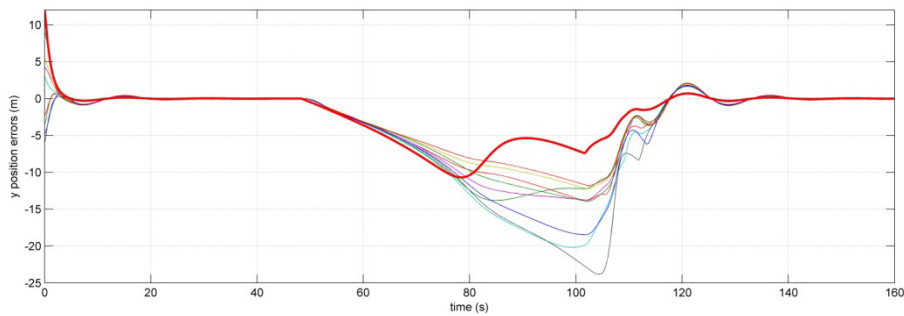


Figure 7.27: Y position errors of the unmanned vehicles for the simulations shown in Fig. 7.25

In Fig. 7.28 it is provided the control signals which affect the leader vehicle.

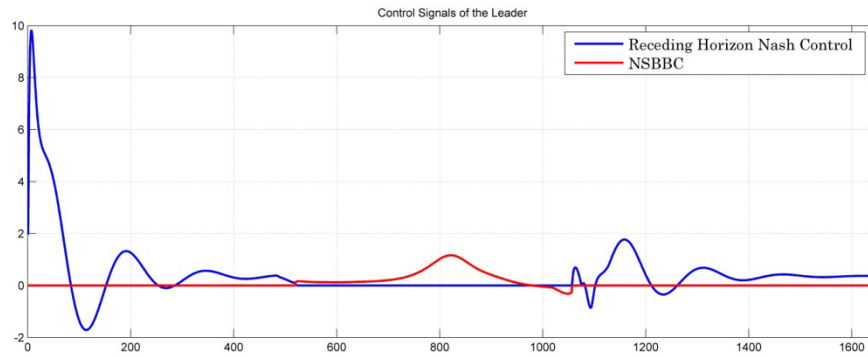


Figure 7.28: Control signals that affect the leader vehicle for the simulations shown in Fig.7.25

Till now, the simulations results are provided for the obstacle which was always on the plane, mobile or immobile, where the formation of unmanned vehicles was tracking the desired trajectories. But, what happens if something unexpected occurs on their trajectory? In other words, what happens if an obstacle suddenly appears? In the following simulations results, the answers to these questions are given.

In the following simulation it is considered a suddenly appearing small obstacle and all the agents runs the obstacle avoidance algorithm.

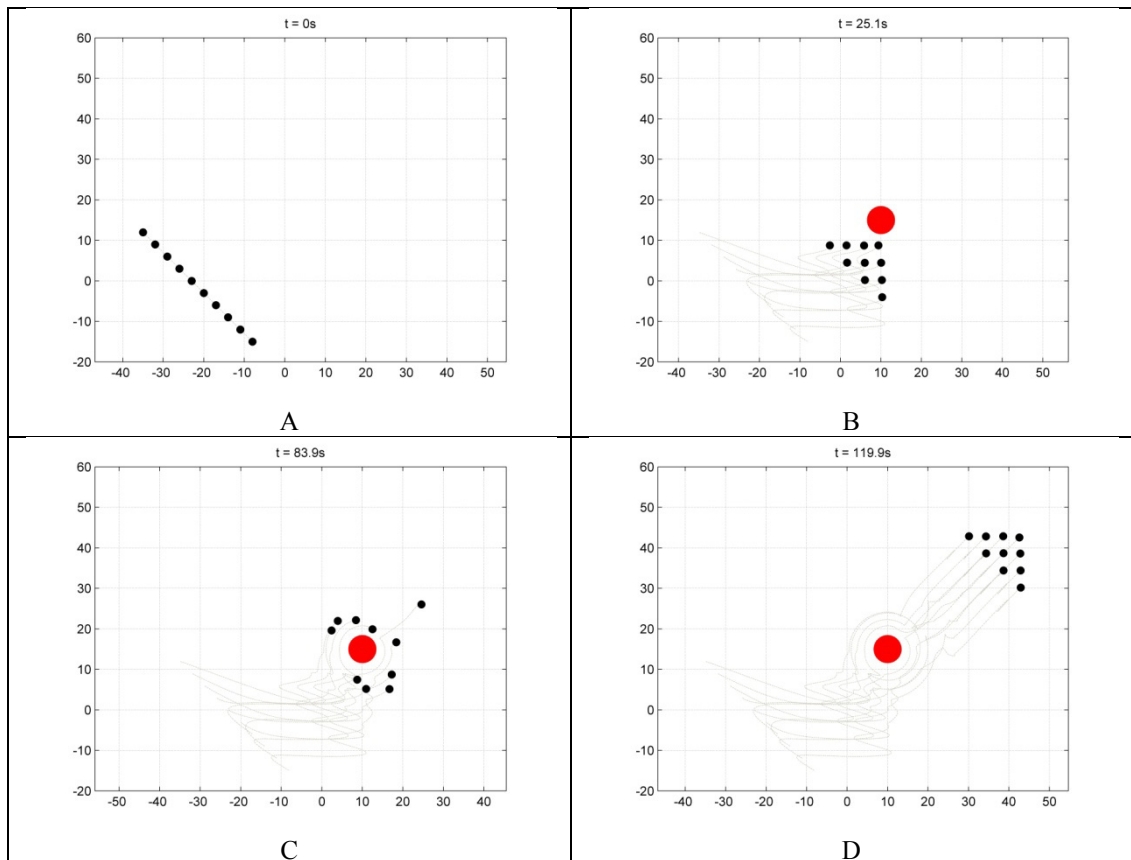


Figure 7.29: Phase plane of unmanned vehicles with a suddenly appearing small obstacle where all the vehicles execute the obstacle avoidance algorithm.

As it can be seen from Fig. 7.29, at the beginning of the simulation, there is no obstacle on the plane where the agents are moving. Then suddenly an obstacle appears in front of the formation of the unmanned vehicles and they can also avoid this suddenly appearing small obstacle.

In Fig. 7.30 and Fig. 7.31 the position errors of the unmanned vehicles on X axis and Y axis are provided respectively.

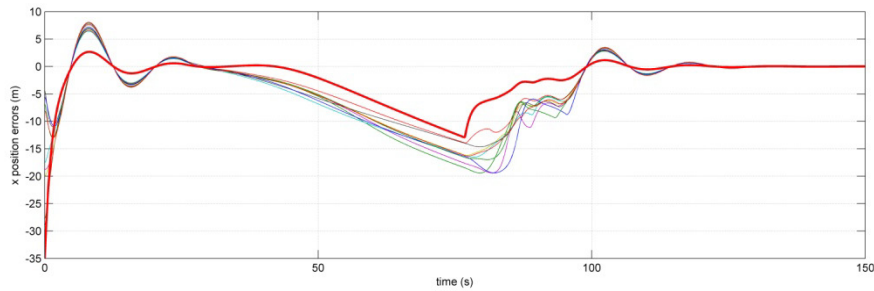


Figure 7.30: X position errors of the unmanned vehicles for the simulations shown in Fig.7.29

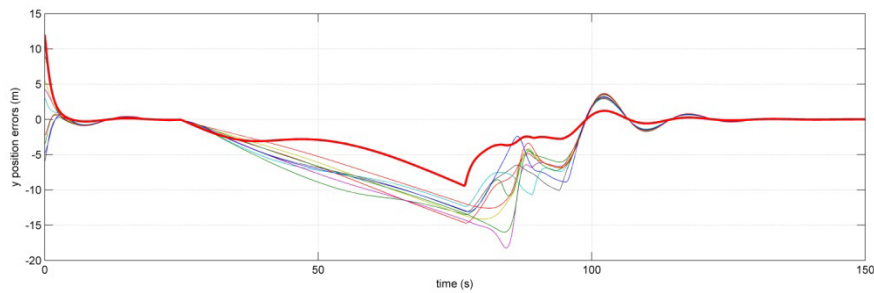


Figure 7.31: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.29

In Fig. 7.32 it is provided the control signals which affect the leader vehicle.



Figure 7.32: Control signals that affect the leader vehicle for the simulations shown in Fig.7.29

In Fig. 7.33, the simulation results for a suddenly appearing obstacle are provided, but this time the obstacle is bigger than the previous one.

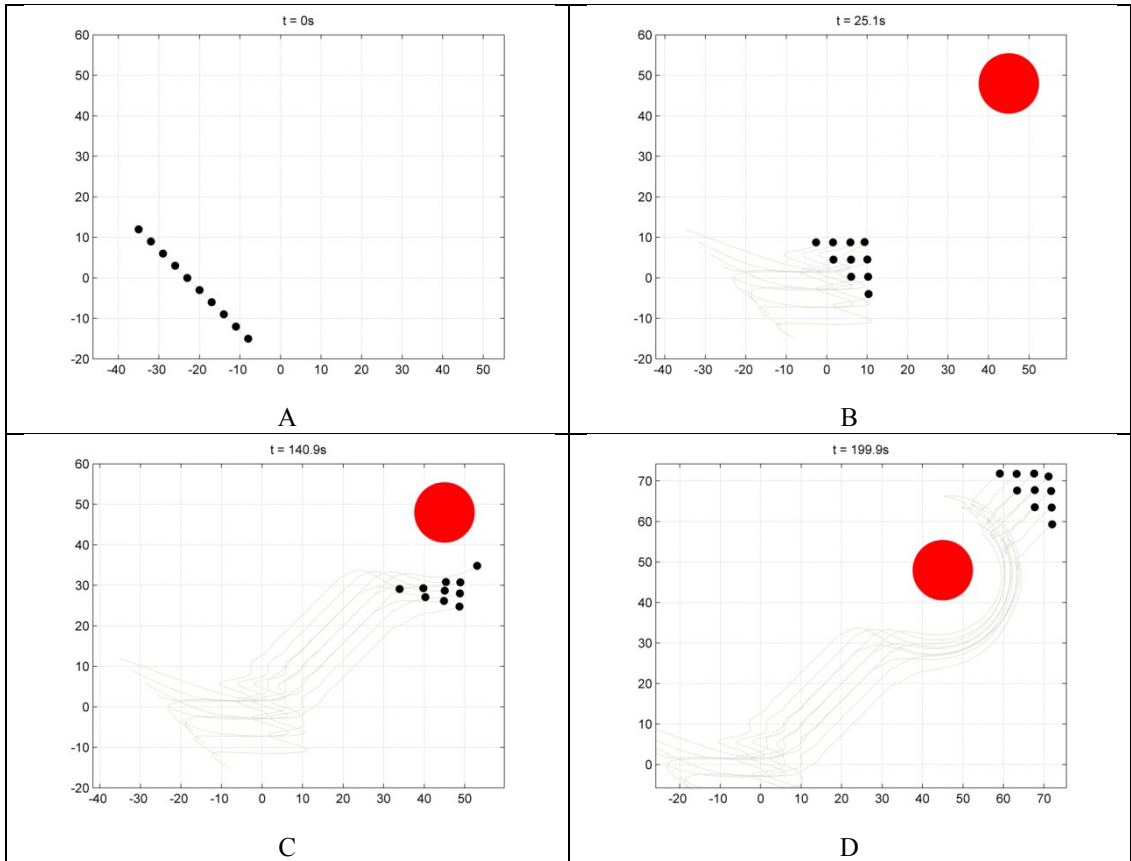


Figure 7.33: Phase plane of unmanned vehicles with a suddenly appearing bigger obstacle where all the vehicles execute the obstacle avoidance algorithm.

In Fig. 7.34 and Fig. 7.35 the position errors of the unmanned vehicles on X axis and Y axis are provided respectively.

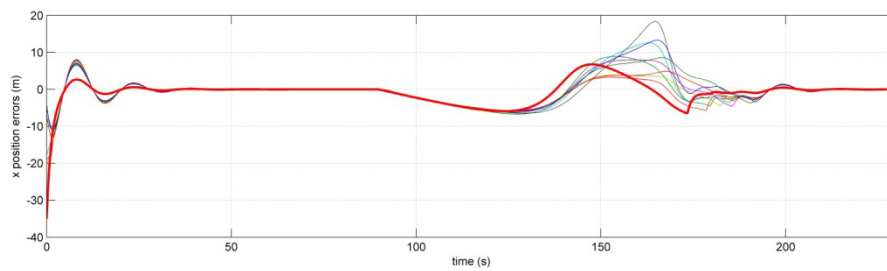


Figure 7.34: X position errors of the unmanned vehicles for the simulations shown in Fig.7.33

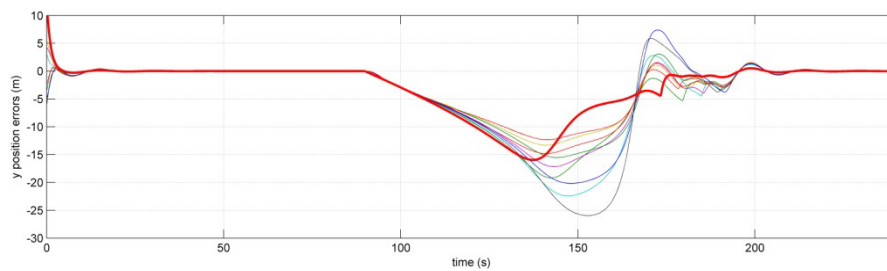


Figure 7.35: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.33

In Fig. 7.36 it is provided the control signals which affect the leader vehicle.

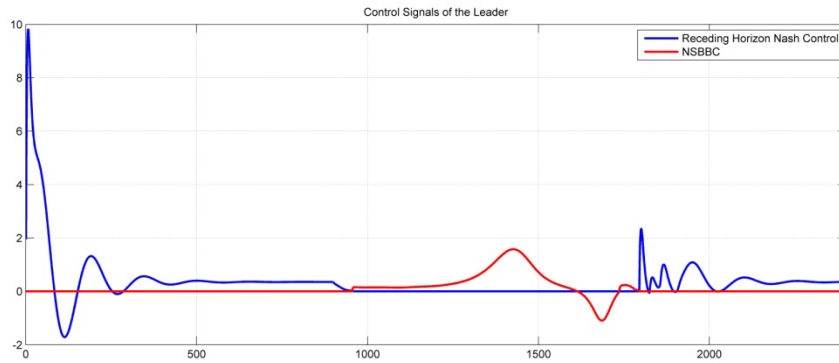


Figure 7.36: Control signals that affect the leader vehicle for the simulations shown in Fig.7.33

The below provided simulations were for the cases where there was only one obstacle. But also it is necessary to think the situations where the formation faces with more complex situations such as several obstacles. In the next simulations it will be provided the cases where there is more than one obstacle.

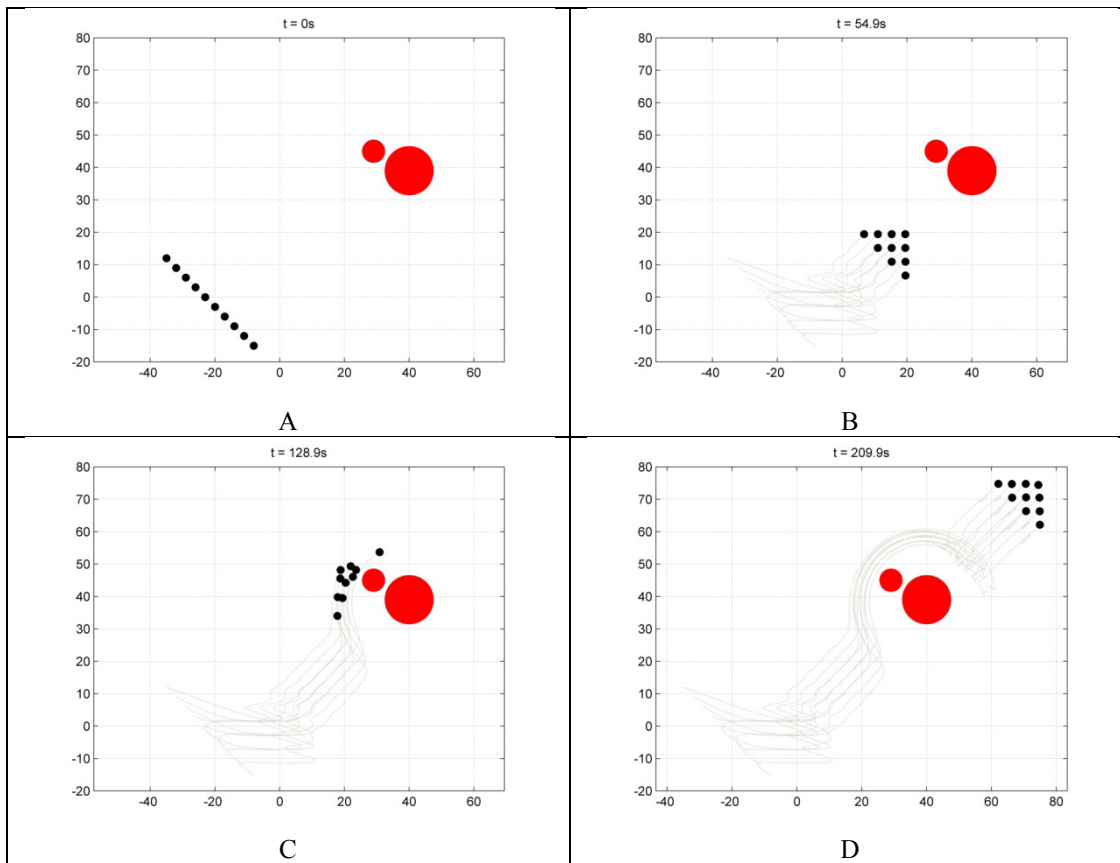


Figure 7.37: Phase plane of unmanned vehicles with two immobile obstacles where all the vehicles execute the obstacle avoidance algorithm.

In Fig. 7.38 and Fig. 7.39 the position errors of the unmanned vehicles on X axis and Y axis are provided respectively.

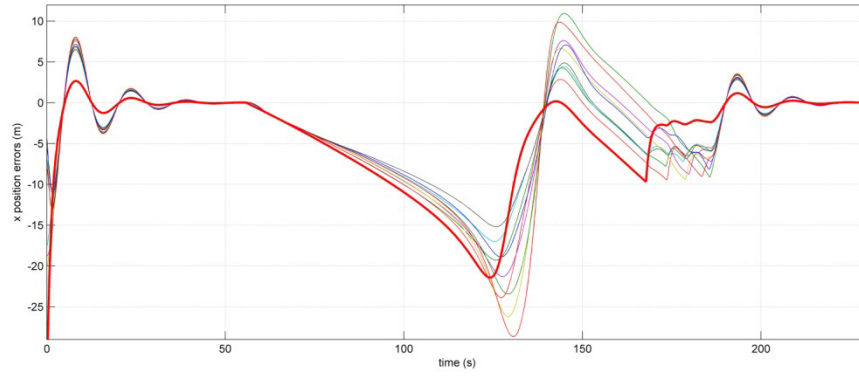


Figure 7.38: X position errors of the unmanned vehicles for the simulations shown in Fig.7.37

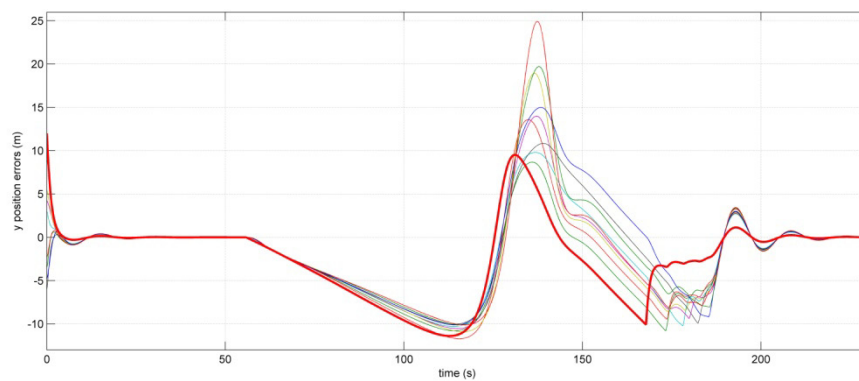


Figure 7.39: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.37

In Fig. 7.40, it is seen the control signals that affect the leader vehicle.



Figure 7.40: Control signals that affect the leader vehicle for the simulations shown in Fig.7.31

In the next figures, it will be provided the simulations results again more than one obstacle case with a different scenario.

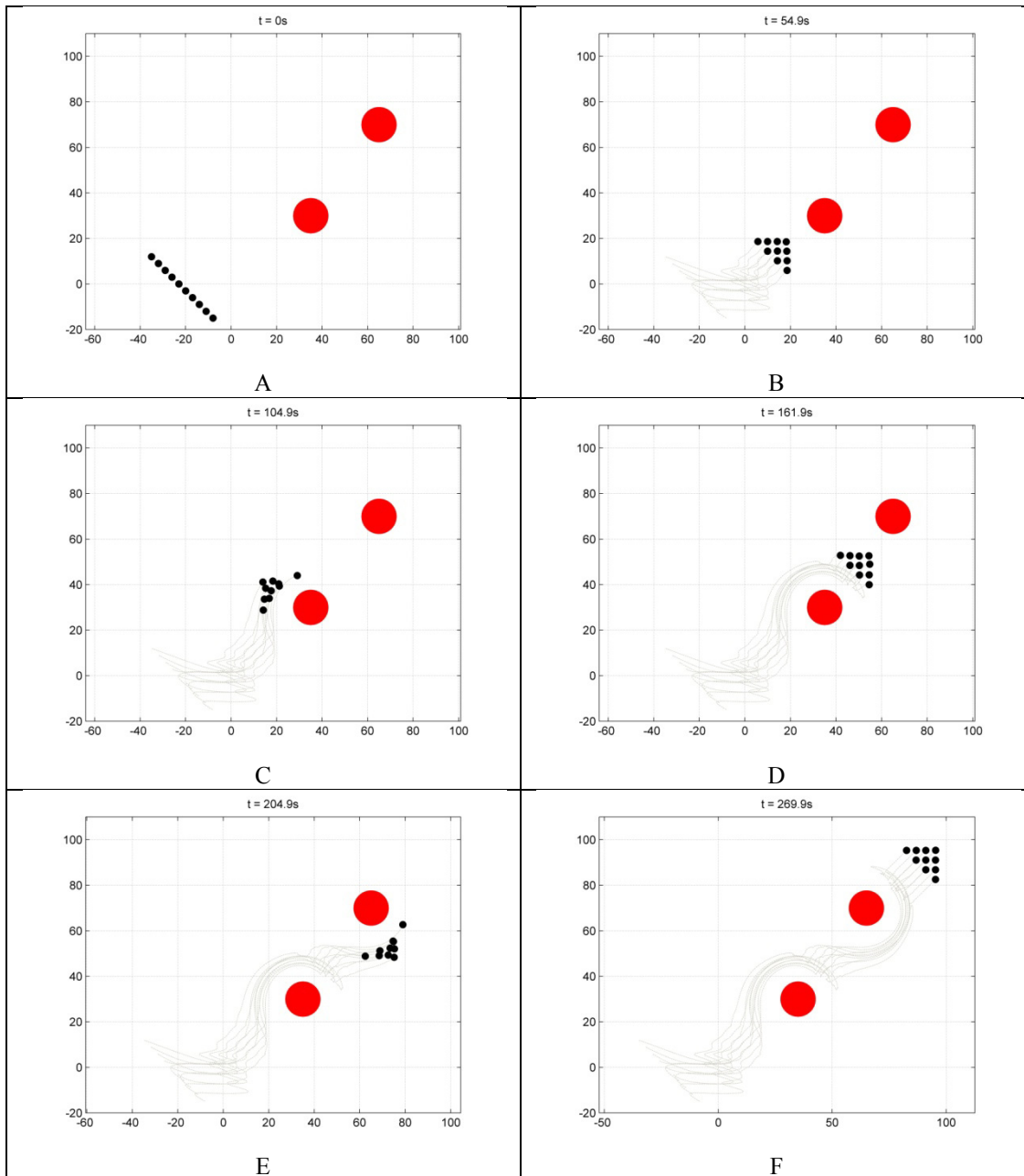


Figure 7.41: Phase plane of unmanned vehicles with two successively immobile obstacles where all the vehicles execute the obstacle avoidance algorithm.

In Fig. 7.42 and Fig. 7.43 the position errors of the unmanned vehicles on X axis and Y axis are provided respectively.

As it can be seen from Fig. 7.37 and Fig. 7.41, the vehicles can avoid also several obstacles even if they are side by side or successively.

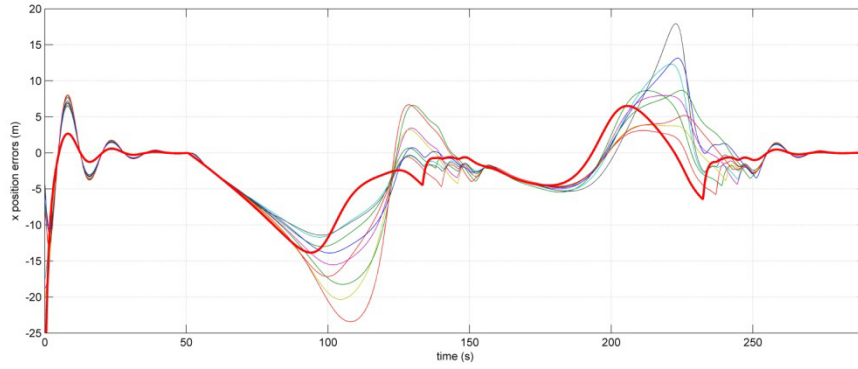


Figure 7.42: X position errors of the unmanned vehicles for the simulations shown in Fig.7.41

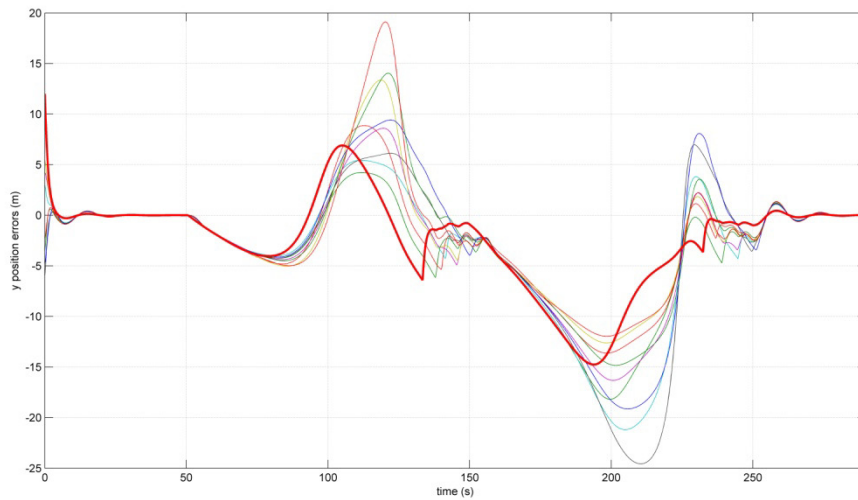


Figure 7.43: Y position errors of the unmanned vehicles for the simulations shown in Fig.7.41

In Fig. 7.44, it is seen the control signals that affect the leader vehicle.

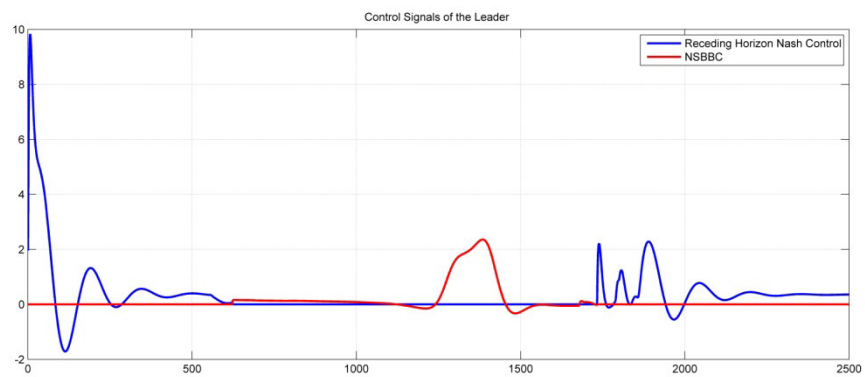


Figure 7.44: Control signals that affect the leader vehicle for the simulations shown in Fig.7.41.

8. CONCLUSION

The major purpose of this thesis was to provide a game theoretical approach to the control of a formation of unmanned vehicles. The objectives of the formation were to follow a prescribed trajectory and avoiding obstacles while maintaining the geometry of the formation. Formation control is implemented using game theoretical approach while obstacles were avoided using NSBBC algorithm. Different obstacle avoidance scenarios are analyzed and compared. Numerical simulation results are presented, to validate the proposed approach.

From the simulations provided in Chapter 7, it is seen that when the leader vehicle takes care about the followers the whole formation can avoid the obstacle while maintaining the triangular formation. But if the leader does not care about the followers, it can again avoid the obstacle perfectly but it is seen that some of the followers cannot avoid it since they try to maintain the shape of the formation. To overcome this problem, we executed NSBBC obstacle avoidance algorithm on all the unmanned vehicles and it is seen that when all of them executes the algorithm, they can avoid the obstacle(s) even if the leader does not take about them. Naturally, to be able to avoid the obstacle(s), the unmanned vehicles do not try to maintain the shape of the formation and the primary task becomes obstacle avoidance. As soon as they avoid the obstacle, the primary task becomes trajectory tracking and they track their prescribed trajectories while also maintaining the shape of the formation.

Different simulations results are provided to show that the solution proposed in this thesis is valid also for the cases where the vehicles must avoid different types of obstacles like moving, suddenly appearing and more than one obstacle.

As the future work, constrained control can be applied to game theoretical formation control and based on this idea it is foreseen that there might be no need to any additional algorithm to avoid the obstacle(s). Constrained control can deal with the saturation of the control signals. Thus it can also handle the obstacle and collision avoiding problems by converting the obstacle avoidance problem into state constraints.

References

- [1] K. Capek, *R.U.R (Rossum's Universal Robots)*, 1920.
- [2] R. C. Arkin, *Behavior-Based Robotics*, Cambridge, Mass., MIT Press, 1998.
- [3] D. R. Hill, "Mechanical Engineering in the Medieval Near East", *Scientific American*, May, 1991.
- [4] C. B. Fowler, "The Museum of Music: A History of Mechanical Instruments", *Music Educators Journal*, vol. 54, no. 2, October, 1967.
- [5] M. E. Moran, "The da Vinci Robot", *Journal of Endourology*, vol. 20, no. 12, December, 2006.
- [6] J. Renner, "Robot Dreams: The Strange Tale Of A Man's Quest To Rebuild His Mechanical Childhood Friend", *Free Times*, vol. 13, no. 35, December, 2005.
- [7] W. W. Grey, "An Imitation of Life", *Scientific American*, May, 1950.
- [8] W. W. Grey, "A Machine that Learns", *Scientific American*, August, 1951.
- [9] B. Gates, "A Robot in Every Home", *Scientific American*, January, 2007.
- [10] P. Jensfelt, "Approaches to Mobile Robot Localization in Indoor Environments", *PhD thesis, KTH, Sweden*, ISBN 91-7283-135-9,
- [11] M.B. Dias and A. Stentz. Opportunistic optimization for market based multirobot control. *IEEE International Conference on Intelligent Robots and Systems*, September 2002.
- [12] V. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Transactions on Systems Man and Cybernetics*, vol. 20, pp. 1058 – 1068, 1990.
- [13] V. Lumelsky and Stepanov, "Path-planning strategies for a point mobile automaton amidst unknown obstacles of arbitrary shape," in *Autonomous Robots Vehicles, I.J. Cox, G.T. Wilfong (Eds), New York, Springer*, pp. 1058 – 1068, 1990.
- [14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1995.
- [15] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Transaction on Robotics and Automation*, vol. 7, no. 3, pp. 278 – 288, 1991.
- [16] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 802 – 807, 1993.
- [17] E. Bicho and G. Schoner, "The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform," *Robotics and Autonomous Systems*, vol. 21, pp. 23 – 35, 1997.
- [18] E. Bicho, *Dynamic Approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation*. Shaker Verlag, 2000.
- [19] G. Antonelli, F. Arrichiello, S. Chiaverini (2005). The Null-Space Based Behavioral Control for Mobile Robots, *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland.

- [20] M. Cellini, R. Mati, L. Pollini, M. Innocenti (2007). Obstacle Avoidance for Unmanned Ground Vehicle in Unstructured Environment. *AIAA Guid., Nav. and Con. Conf. and Exh.*, Hilton Head Island.
- [21] S. Chiaverini F. Arrichiello and T. I. Fossen. Formation control of underactuated surface vessels using the null-space-based behavioral control. Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5942–5947, 2006.
- [22] S. L. Veherencamp. A handbook of behavioral neurobiology. P. Marler and J.G. Vandenbergh, pages 354–382, 1987.
- [23] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams ", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, December, 1998.
- [24] S. Lacroix G. Hattenberger and R. Alami. Formation flight: Evaluation of autonomous configuration control algorithms. Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2628–2633, 2007.
- [25] T. Balch and M. Hybinette. Social potentials for scalable multi-robot formations. *Proc. of the 2000 IEEE Int. Conference on Robotics and Automation*, pages 73–80, 2000.
- [26] M.A. Lewis and K.H. Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robot*, pages 387–403, 1997.
- [27] P. Johnson and J. Bay. Distributed control of simulated autonomous mobile robot collectives in payload transportation. *Autonomous Robot*, pages 43–64, 1995.
- [28] M. A. Bender S. P. Fekete T.-R. Hsiang, E. M. Arkin and J. S. B. Mitchell. Algorithms for rapidly dispersing robot swarms in unknown environments. *Algorithmic Foundations of Robotics*, pages 77–94, 2003.
- [29] M. Mataric A. Howard and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems*, 2002.
- [30] S. Poduri and G. S. Sukhatme. Constrained coverage for mobile sensor networks. *IEEE International Conference on Robotics and Automation*, pages 165–172, 2004.
- [31] H. Yamaguchi. A cooperative hunting behavior by mobile robot troops. *Intl. J. Robotics Research*, pages 931–940, 1999.
- [32] J. P. Ostrowski J. Desai and V. Kumar. Controlling formations of multiple mobile robots. *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2864–2869, 1998.
- [33] P. Song and V. Kumar. A potential field based approach to multi-robot manipulation. *Proc. of the 2002 IEEE Int. Conference on Robotics and Automation*, pages 870–876, 2002.
- [34] A. Okubo. Dynamical aspects of animal grouping: swarms, schools, flocks and herds. *Advances in Biophysics*, pages 1–94, 1985.
- [35] J. K. Parrish and W. H. Hammer. Animal Groups in Three Dimensions. *Cambridge University Press*, 1997.
- [36] C.W. Reynolds. Flocks: herds and schools: A distributed behavioral model. *Proceedings of the 14th annual conference on Computer graphics*, pages 25–34, 1987.
- [37] E. Ben-Jacob I. Cohen T. Vicsek, A. Czirok and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, pages 1226–1229, 1995.

- [38] H. Yamaguchi. A cooperative hunting behavior by multiple nonholonomic mobile robots. *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3347–3352, 1998.
- [39] A.S. Morse A. Jadbabaie, J. Lin. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 2953–2958, 2002.
- [40] K.T. Simsarin M.J. Mataric, M. Nillson. Cooperative multi-robot boxpushing. *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 556–561, 1995.
- [41] R.L. Frost T.B. Gold, J.K. Archibald. A utility approach to multi-agent coordination. *International Conference on Robotics and Automation, ICRA '00*, pages 2052–2057, 2000.
- [42] P. K. C. Wang. Navigation strategies for multiple autonomous mobile robots moving in formation. *Journal of Robotic Systems*, pages 177–195, 1991.
- [43] J. Lawton R. W. Beard and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Trans. Control Syst. Technol.*, pages 777–790, 2001.
- [44] G.J. Pappas H.G. Tanner and V. Kumar. Input-to-state stability on formation graphs. *Proc. 41st IEEE Conf. Decision and Control*, pages 2439–2444, 2002.
- [45] M. Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, page 321331,1995.
- [46] G. Bekey L. E. Parker and Eds J. Barhen. Current state of the art in distributed autonomous mobile robotics. *Distributed Autonomous Robotic Systems*, page 312, 2000.
- [47] R. Brooks. A robust layered control system for a mobile robot. *IEEE J.Robotics and Automation*, page 1423, 1986.
- [48] T. Balch and R. C. Arkin. Behavior-based formation control for multiagent robot teams. *IEEE Transactions on Robotics and Automation*,1999.
- [49] T. Balch and M. Hybinette. Social potentials for scalable multi-robot formations. *Proc. of the 2000 IEEE Int. Conference on Robotics and Automation*, pages 73–80, 2000.
- [50] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *In Proc. IEEE Int. Conf. Robotics and Automation*, 1985.
- [51] J. H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*,1999.
- [52] H. Yamaguchi and T. Arai. Distributed and autonomous control method for generating shape of multiple mobile robot group. *International Conference on Intelligent Robots and Systems*, pages 800–807, 1994.
- [53] M. Fields K. Valavanis L. Barnes, W. Alvis and W. Moreno. Heterogeneous swarm formation control using bivariate normal functions to generate potential fields. *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 2006.
- [54] G. Bekey L. E. Parker and Eds J. Barhen. Current state of the art in distributed autonomous mobile robotics. *Distributed Autonomous Robotic Systems*, page 312, 2000.
- [55] J. Jongusuk and T. Mita. Tracking control of multiple mobile robots: A case study of inter-robot collision-free problem. *Proc. IEEE International Conference on Robotics and Automation*, pages 2885–2890, 2001.

- [56] Fudenberg D. and Tirole, J. 1996. Game Theory, *the MIT Press*, Fifth Printing (First Printing In 1991).
- [57] Gardner, R. 1995. Games for Business and Economics, *John Wiley&Sons, Inc.*
- [58] Ray, I. 2000. Game Theory and the Environment: Old Models, New Solution Concepts, Games & Environment, *Journal of Economic Literature*.
- [59] Ritzberger, K. 2002. Foundations of Non-Cooperative Game Theory, *Oxford University Press*.
- [60] Myerson R. 1991. Game Theory: Analysis of Conflict. London, England: *Harvard University Press*.
- [61] Bonabeau E, Dorigo M, Theraulaz G. 1999. Swarm Intelligence: From Natural to Artificial Systems. *New York, NY: Oxford University Press*.
- [62] Smith MJ. 1982. Evolution and the Theory of Games. Oxford, Great Britain: *Cambridge University Press*.
- [63] H. A. Kandil, G. Freiling, V. Ionescu, and G. Jank, Matrix Riccati Equations. *Birkhauser, Germany: Verlag*, 2003.
- [64] J. C. Engwerda, LQ Dynamic Optimization and Differential Games. *New York: Wiley*, 2005.
- [65] J. Shinar and V. Glizer, "Application of receding horizon control strategy to pursuit-evasion problems," *Opt. Control Appl. Methods*, vol. 16, no. 2, pp. 127–142, 1995.
- [66] K. B. Kim, M.-J. Kim, and W. H. Kwon, "A differential game missile guidance law via receding horizon control without time-to-go," in *Proc. Int. Conf. Elect. Eng. (ICEE)*, 1998, pp. 47–50.
- [67] J. B. J. Cruz, M. A. Simaan, A. Gacic, H. Jiang, B. Letellier, M. Li, and Y. Liu, "Game-theoretic modeling and control of military air operations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 4, pp. 1393–1405, 2001.
- [68] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 53–64, Jan. 2003.
- [69] Patric Jensfelt. *Approaches to Mobile Robot Localizatoin in Indoor Environments*. PhD thesis, KTH, Sweden, ISBN 91-7283-135-9, 2001.
- [70] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory*. Warrendale, PA: SIAM, 1995.