

University of Denver

Digital Commons @ DU

Electronic Theses and Dissertations

Graduate Studies

1-1-2013

Robot Team Formation Control Using Communication "Throughput Approach"

FatmaZahra Ahmed BenHalim
University of Denver

Follow this and additional works at: <https://digitalcommons.du.edu/etd>



Part of the [Robotics Commons](#)

Recommended Citation

BenHalim, FatmaZahra Ahmed, "Robot Team Formation Control Using Communication "Throughput Approach"" (2013). *Electronic Theses and Dissertations*. 762.
<https://digitalcommons.du.edu/etd/762>

This Thesis is brought to you for free and open access by the Graduate Studies at Digital Commons @ DU. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ DU. For more information, please contact jennifer.cox@du.edu, dig-commons@du.edu.

Robot Team Formation Control using Communication
"Throughput Approach"

A Thesis

Presented to

the Faculty of the Daniel Felix Ritchie School of Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Fatma Zahra A. Ben Halim

August 2013

Advisor: Dr. Kimon Valavanis

©Copyright by Fatma Zahra A. Ben Halim 2013

All Rights Reserved

Author: Fatma Zahra A. Ben Halim

Title: Robot Team Formation Control using Communication "Throughput Approach"

Advisor: Dr. Kimon Valavanis

Degree Date: August 2013

ABSTRACT

In this thesis, we consider a team of robots forming a mobile robot network cooperating to accomplish a mission in an unknown but structured environment. The team has no a-priori knowledge of the environment. Robots have limited memory storage capabilities, not enough to map the environment. Each robot also has limited sensor capability and computational power. Due to the need to avoid obstacles and other environment effects, some robots get delayed from the rest. Using tracking controller, the robot team should follow the leader in a flexible formation shape without losing network connectivity, and that was achieved by monitoring the end-to-end throughput level.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Kimon Valavanis for his support and endless aiding material, my friends for their help, and of course, I would like to thank my dear husband and daughter for their encouragement and patience.

TABLE OF CONTENTS

Abstract.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
Chapter 1.....	1
INTRODUCTION.....	1
1.1 Overview.....	1
1.2 Motivation.....	2
1.3 Problem Statement.....	3
1.4 Proposed Solution.....	4
1.5 Contributions.....	6
Chapter 2.....	8
RELATED LITERATURE REVIEW.....	8
2.1 Overview.....	8
2.2 Related Work.....	9
2.2.1 Formation Control.....	9
2.2.2 Path Planning and Trajectory tracking.....	13
Chapter 3.....	16
MATHEMATICAL MODELING OF FORMATION CONTROL STRATEGY.....	16
3.1 Research Assumptions.....	18
3.2 Sensors.....	19
3.3 Shape Formation.....	20
3.3.1 Formation Shapes.....	21
3.3.2 Methodology.....	22
3.3.3 Formation Control Algorithm.....	24
3.3.4 Algorithm 1, Formation shape control and leader election.....	24
3.3.5 Failing or Lost Robot scenarios.....	25
3.4 On-Line Path planning and Obstacle Avoidance.....	28
3.4.1 Methodology.....	28
3.4.2 Algorithm 2, Modified PRM for Unknown Environments.....	31
3.5 Tracking controller.....	34
3.6 Ad-Hoc network.....	38
3.7 Throughput.....	38
3.7.1 Signal Strength and Bit Error Rate.....	40
3.7.2 Routing Protocols:.....	41
3.8 Discussion.....	43
3.8.1 Shape Formation.....	43

3.8.2	On-Line Path Planning.....	44
3.8.3	Ad Hoc Network and Throughput Calculations	44
Chapter 4	45
SIMULATION STUDIES	45
4.1	On-line Path Planning (PRM).....	45
4.2	Throughput Calculation	48
4.2.1	Map1	50
4.2.2	Map2	54
4.2.3	Map3	58
4.3	Failing Robot Scenarios.....	61
Chapter 5	64
CONCLUSIONS AND FUTURE WORK	64
References	66

CHAPTER 1

INTRODUCTION

1.1 Overview

Multi-robot systems have many advantages over single robot ones. A team of robots may accomplish efficiently a wider range of tasks compared to single robots, providing robustness and fault tolerance to mission execution. To make the process more efficient, the team should be able to preserve resources, and follow the trajectory path with minimum power consumption.

The problem under consideration is defined as the derivation of the optimal placement of a small number of robots relative to a leader robot, such that missions are completed in unknown but structured and static environments, while at the same time a minimum level of “communication quality” in terms of link throughput is guaranteed. The robot team will be assumed to be in a fixed or loose formation, depending on whether the immediate surrounding area is obstacle free or not. This problem accounts for challenges that are typically considered separately. However, they cannot be defined independently of one another, as is the case of coordinated motion planning and path planning, to say the least.

1.2 Motivation

The research focus is on a small team of robots that is capable of moving in a formation from a starting point to the goal point avoiding obstacles with minimum level of communication, and maintaining network connectivity for transmitting critical data if and when needed. Using a tracking controller [25] alone may complete an assigned task/mission, where the system control inputs depend only on the separation distance, in such a way that the formation/separation errors (linear and angular) converge to zero to achieve the desired geometric formation. However, keeping the wireless network of the team connected with an acceptable level of communication quality is not guaranteed [56]. To be specific, any two mobile robots can exchange messages as long as the transmitted data power or signal to noise ratio, SNR, is above the receiver threshold [55], but that does not guarantee good data throughput or minimum delay, especially if considering robot mobility and how throughput is affected [56].

Such applications require an on-line measurement of communication quality, like network throughput or SNR, then, using such measurements as an input for the mobile robot local controller, which also considers the environmental constraints and obstacles in an unknown but structured and static environment. According to these measurements, aside from the separating distance, the local controller will increase (or decrease) the robot velocity to adjust the position of the robot, hence, keeping the level of communication quality within acceptable levels.

Thus, the goal of this work is to achieve follow-the-leader formation control for such a robot team, where the follower's position is not fixed with respect to the leader, but it is robust enough to handle disjoint formation without the need to know the other

robots' exact positions. The robot team formation may also need to avoid obstacles and sometimes it may be required to perform formation switching during complex tasks (e.g., passing a narrow passage) or as a result of a robot that failed. The considered system must also overcome the challenge of self-localization of the followers due to limited on-board sensors and computational power.

1.3 Problem Statement

A major challenge related to a typical robot formation strategy is the lack of inter-robot information feedback throughout the team. For example, feedback from the followers is not used by the leader in a leader-follower approach, so the formation can become disjoint and followers can be left behind if they are not able to track and follow the motion of the leader correctly [14], as shown in Fig. 1.1.

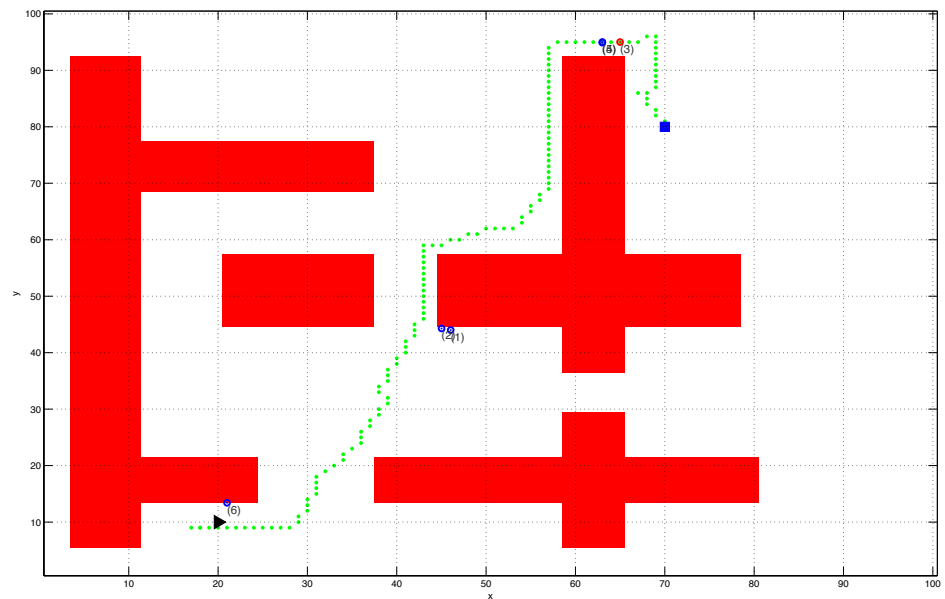


Figure 1.1 Followers left behind when their feedback is ignored [simulation results]

On the other hand, if the formation is based on using sensors only and without inter-communication, continuous knowledge of other robots' positions and headings are needed to avoid obstacles and pass narrow passages, which becomes a tricky and difficult problem [29].

The focus is on making the followers responsible for locally controlling the formation shape and maintaining the desired distance/bearing to the leader by using end-to-end throughput level as the metric for evaluating the performance of the mobile robot team. Using end-to-end throughput level assessment gives an indication of how long the time delay between the leader and the followers is, hence, signifying the distance between them.

Additionally, robots lack prior knowledge of the environment during the path planning phase, except that it is a structured and static environment. Since robots cannot predict local minima before detecting the obstacles forming local minima, robots may get trapped in local minima [26]. Furthermore, if the planner is not efficient enough, robots attracted by the goal point will take more time to find the right path, which leads to more power consumption.. Hence, an efficient path planner should produce low cost paths and avoid local minima.

1.4 Proposed Solution

Since robot teams usually operate in remote regions with little to no infrastructure and the robots are often equipped with low-power short-range wireless network interfaces that only allow for direct communication with their near neighbors, a practical approach to distributed control and sensing is to apply a Leader-Follower or Leader-Referenced

formation control method [5] for several reasons. In Unit-Center-Referenced formations any robot move or turn affects the entire formation. For Leader-Referenced formations, the leader moves in any direction, and the followers adjust to move into a pattern position. In Unit-Center-Referenced formations, each robot computes a unit-center independently by averaging the x and y position of all the robots involved in the formation; it then determining its own formation position relative to that center. This means that each robot may have to track a number of other robots. In Leader-Referenced formations a call for tracking is for one robot [17]. The communication between the leader and the followers is through an Ad Hoc wireless networking scheme [28] with one hop distance for maximum end-to-end throughput. In this thesis, a shape control algorithm is proposed, detailed in Chapter 3, to reduce the total number of hop counts required for all transmissions between robots, thus increasing the throughput level.

Due to their efficiency and simplicity, Probabilistic Road Map, PRM, planning algorithms are widely used algorithms for path planning of mobile robots [26][41]. In this thesis, we are implementing a modified and improved version of PRM to plan a path for the leader robot in an unknown environment. For the coordinated robot team movement, we are implementing a feedback-tracking method as a local controller for the followers to track the position of the leader. All robot positions are represented in polar coordinates, where the linear velocity command becomes a function of link throughput and separation distance. Each robot individually will use the calculated velocity to converge to the desired goal point.

It assumed that each robot is equipped with a Laser Range Finder (LRF), mounted on each robot, such that the LRF sweeps the environment ahead of the robot acquiring environment measurements.

Simulation studies are conducted to assess and compare different formation performances under the proposed strategies. The simulation software was developed using MATLAB (2007a) as a network simulator to verify the effectiveness of the above mentioned strategies.

1.5 Contributions

One of the main contributions of this thesis is controlling and maintaining the team formation using wireless network communication constraints. Simulation results show improvement in team performance, where each and every follower robot kept the specified distance to their leader with changing velocity without forcing the leader to slow down for delayed followers to catch up, or increasing the number of exchanging messages between the robots.

An enhancement to the PRM algorithm for path planning in unknown but structured and static environments has been developed, which produced better results in case of local minima. The PRM planner main idea is to initially generate random point samples from the leader view field and connecting them using a simple but very fast motion planner. The connected points are stored with the edges representing the probable paths. Then, the shortest path to the furthest random point in the sensor range aligned with the goal location is chosen. The algorithm repeats the same process until reaching the goal point.

The key point in this development is that in each step of path planning, if an obstacle is detected, the planner limits the local free region for the robot to sample from, so that the robot will avoid local minima and the path will be generated more efficiently.

1.6 Structure of the Thesis

The chapters of this thesis are organized as follows. Chapter 2 focuses on the literature review. Chapter 3 presents a detailed mathematical modeling of the proposed formation control strategy, including trajectory planning, shape forming and the feed back tracking controller configuration. Simulation results and analysis is presented in Chapter 4. Chapter 5 concludes the thesis and proposes future work.

CHAPTER 2

RELATED LITERATURE REVIEW

2.1 Overview

A current important challenge in multi-robot teams is the issue of coordinated motion of multiple robotic systems [57]. The problem considered in this is how to make a team with small number of robots autonomously maintain formation shape using the network links throughput as a reference, while reaching a goal point after navigating through an unknown but structured and static environment.

In research on mobile robot system coordination, both centralized and decentralized control strategies have been studied. Centralized control strategies have the advantage of being able to reach a global optimum solution for tasks such as path planning and reconfiguration [22][23]. However, centralized algorithms are impractical in leader robots failure case, which prevents their implementation in real-time applications. Furthermore, in unknown environments, it is pointless to look for the optimum path planning; avoiding local minima with less number of steps would be considered enough. On the other hand, decentralized algorithms only require local information and can efficiently achieve multi-robot coordination [24], except when it comes to path planning in an unknown environment, since decentralized algorithms requires extensive communication and high computational power.

The system is designed with distributed planning and control to reduce the per-robot cost by centralizing the on-line path planning and distributing the local control and sensing to minimize inter-robot communication.

2.2 Related Work

2.2.1 Formation Control

In formation control, a group of mobile robots has to establish and maintain predetermined but also dynamically reconfigurable shapes, which can be accomplished by controlling each robot's heading and position relative to the team, while allowing the team to move as a whole (the most common formation shapes are hexagon, column, diamond, and wedge [18]). Communication between mobile robots is also essential. For example, using coordination-oriented communication for robot formation control has been extensively studied in the literature [30]. Moreover, in [31] the authors proved that mobility provides certain communication performance improvement.

Several approaches have been proposed to solve the formation control problem. Research in [11] has demonstrated how a set of simple behaviors can be combined depending only on local sensing to produce a global flocking behavior. However this work did not consider the need for communication between robots. Another similar work on a robot soccer-playing team with behavior-based control system has been exhibited in [12].

In [9], a simulated robot team was used to form different shapes, while orienting themselves to a specific robot. The draw back in this approach is the need to more inter-

communications and global knowledge of all robots' positions and headings. In the potential field based technique presented in [8], other variables have been taken into account like the disturbances produced by the interactions between different robots in the system and the interferences caused by obstacles in the environment. However they needed to tune the weight functions to be based on the distance to the center of the potential field. Furthermore, potential field based systems inherently necessitate higher level of heuristics when getting trapped in local minima [54].

In formation control, each robot must know its geometric position relative to the locations of the other robots (or leader robot as in our case). Three primary methods have been identified to accomplish this position configuration [17]:

- Unit-Center-Referenced: each robot computes a unit-center independently by averaging the x and y positions of the entire robot team. Each robot determines its own formation position with respect to that center.
- Leader-Referenced: each robot determines its formation position independently in relation to the leader robot. The leader does not attempt to maintain formation; the other robots are responsible for formation maintenance.
- Neighbor-Referenced: each robot maintains its position with respect to another specified robot.

Leader-Referenced method is used in most applications for several reasons; for example, in Unit-Center-Referenced formations, any robot move or turn affects the entire formation. On the other hand, for Leader-Referenced formations, the leader simply moves in the new direction and the other robots must adjust to move into position [17].

Also, in Unit-Center-Referenced formations, each robot has to track all the other robots where in Leader-Referenced formations only call for the tracking of one robot.

Other approaches have been proposed to implement formation control of a team of mobile robots using more onboard sensing capabilities. As seen in [17], the authors proposed a behavior-based approach for maintaining the formation of a team of unmanned ground vehicles used for military applications. Each vehicle has a mounted camera and a GPS sensor. Although this method can execute formation behaviors efficiently in obstacle avoidance, it requires extensive communication, high computational power and expensive sensors in each robot.

When it comes to formation geometry, two types of formations exist, rigid and non-rigid formations [58]. In rigid formations, the geometry is fully specified and accurately maintained by the motion control of each robot [7][28]. These approaches require switching between geometric shapes when the environment demands it [5]. However, due to obstacle avoidance and environmental conditions, the geometric structure can be distorted and consequentially the actual location of the robots is not considered [6]. That leads at the end of the mission execution to high cost path planning, which could make rigid formation undesirable structure.

In non-rigid formations, some robots positions are not fixed, making the formation more flexible to move, but hard to model though [58].

An interesting approach to rigid formation control although the network is not homogeneous, is that of [27] where the authors proposed using rigid graph theory to

define the formation; they also proposed a gradient control law involving given distances, which would give the formation more flexibility..

In [16], the concept of virtual rigid structures for formation maintenance is applied. Each robot is controlled to maintain a rigid geometric relationship to another robot and to a frame of reference. But since the localization for each robot is computed by a fixed and independent vision-based camera system, the application is constrained in experimental environments.

The authors of [7] presented switching decentralized controllers, where each robot is equipped with only a single camera that stream to a centralized processing unit. The centralized processing unit gathers the pose and velocity information from all the robots; hence, significant sensing capabilities and intense computation are required in these approaches, which increases the system cost for each robot.

Authors in [4] and [29] proposed another local sensing method for robot formation, which depends only on range measurements. In both works, there is no active communication between the leader and the followers. These methods can achieve global formation movement. However, each robot again requires significant sensing capabilities. Authors in [15] presented a control approach in which sophisticated robots formed a chain to assist the simpler followers. The followers use a camera and a color-tracking algorithm to follow the next neighbor. However, in case of failing leader robot, it would be hard for a follower robot to take the leader's place since that the approach is applied to heterogeneous robots.

Most of the approaches described above are related to this thesis. However, because they require that robots have significant sensing (e.g. vision processing) and computation capabilities, they cannot be directly applied to the robot team under consideration.

2.2.2 Path Planning and Trajectory tracking

Most of the approaches have been devoted to develop trajectory tracking control based on the kinematic equations of the mobile robots represented in Cartesian coordinates, but a few of them were in polar coordinates. Authors of [1] used a sliding-mode tracking controller in polar space; they considered the robot dynamics and external disturbances produced by the environment; the controller was proven effective, having fast response, good transient performance and robustness with regard to parameter variations.

In practice, it is not necessary for mobile robots to reach pre specified pose at a specified time, but it is important to follow the geometric path correctly. This kind of tracking is referred to as path tracking. Much work has been done on path tracking using the Cartesian kinematic model, but few researchers have investigated path-tracking problems in polar coordinates. Authors of [2] introduced a Lyapunov-based control approach for path tracking of mobile robots in polar coordinates, while Authors of [3] used back-stepping method to design a path tracking controller, but they all assumed that the linear velocity input always remained constant.

Aside from tracking problems, it has been shown that the stabilization of these systems described in Cartesian space cannot be accomplished by means of a smooth time invariant state feedback control (Brockett's theorem) as pointed out by [20]. Authors of [21] discussed several smooth nonlinear regulation methods in Cartesian coordinates to overcome the stability problem. While the polar coordinates system stabilization has been proved by the work in [2] and [25].

The robots have no a-priori knowledge of the environment, and lack enough sensors and the required memory to explore the assigned area. Former researchers have done much work in this area, for example, the Bug algorithm proposed by [53] is one of the most famous algorithms, and APF algorithm [8] is also effective in real time avoiding obstacles and navigation. Among these methods, each has its own flaws of which the Local Minima problem is the most common one. Because the robot lacks prior knowledge, and since robots cannot predict local minima before detecting the obstacles forming the local minima, trapped robot in local minima may occur [26]. If the planner is not efficient enough, robots attracted by the goal point will take more time to find the right path, which leads to more power consumption due to the high cost of the robots' paths.

In this work, the implemented path-planning algorithm relays on a modified version of Probabilistic Road Maps PRM planner [26] to find a collision-free path.

Since the main aim of PRM is to quickly find a feasible path to the goal point, thus it doesn't care too much on the costs of the generated routes [26][41]. To resolve

these problems, there are already many methods that are raised, and some of them are effectively in use.

During the last two-decade or so, most papers used PRM in this area were only in an already known environments, and only few researchers have tried to use PRM in path planning without a-priori knowledge of the environments [26][41].

The PRM planner avoids local minima entirely when it generates a connected graph in a known environment [41]. In this sense, it is more reliable than the Artificial Potential Fields planner when it comes to path planning in a known environment. On the other hand, Potential Fields naturally stays a safe distance from obstacles, while the random samples of PRM can make the robot move very in a very close distance [26][41]. These random samples also cause the rather unsmooth jumps between points during path execution, while the Artificial Potential Fields method always results in a smooth trajectory (when it does not get stuck in local minima).

But due to its efficiency and simplicity, PRM planning has become great algorithm for path planning of mobile robots. The main idea of a classic PRM planner is to sample at random from a robot's configuration space (which in our case is the half circle area around the robot with radius equal the sensor range), and connect the sampled points to construct a graph at random, search and select the shortest path in this space. In this thesis, we are implementing a modified and improved version of the presented PRM in [26], to plan a path for the leader robot in unknown environment, which is shown to work well with local minima.

CHAPTER 3

MATHEMATICAL MODELING OF FORMATION CONTROL STRATEGY

This Chapter explains the methodology adopted in this work in terms of path planning, shape formation control strategy, used communication protocols and on board robot sensors. The problem to be solved relates to a team of a small number, N , of robots that forms a mobile robot network with assets navigating in an unknown but structured and static environment. Each robot has limited sensor capabilities and computational power. The robot team formation may need to avoid obstacles and sometimes it is required to perform formation switching during complex tasks (e.g., passing a narrow passage) or as a result of a failed robot. The applied formation control, as explained in Sec. 3.3, is the Leader-Referenced formation control, where each follower robot determines its formation position in relation to the leader robot; the leader does not attempt to maintain formation. The followers are responsible for keeping the formation by using a local tracking controller with the leader posing as a target, while the leader is responsible for computing the path plan from the start point to the goal point. This approach has several advantages:

- It is simple, thus, it can be easily modeled and implemented.

- It requires low inter-communication, as it does not require global knowledge of all robots position, since each follower robot needs to know the position of the leader only.
- It is scalable with respect to the number of robots in the team.

Note that, the formation can become disjoint and followers can be left behind [14] since this approach lacks the inter-robot feedback between the leader and the followers. However, by implementing Ad-Hoc Network protocol, Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with a RTS-CTS-DATA-ACK access mechanism this drawback is overcome, since all followers acknowledge the receiving of the leader message. Furthermore, a failing leader in the Leader-Referenced approach can compromise the whole mission. Therefore, an algorithm to elect a new leader is implemented in each and every robot in the team.

A new PRM approach is presented in sec.3.4. The linear velocity of each follower is a function of the separating distance among the robot and the leader. Used control law equations are detailed in Sec.3.5. The leader unicasts its position and heading regularly to the followers through an Ad Hoc communication (Sec.3.6), however, due to obstacle avoidance, some follower robots get delayed from the rest. Even if the follower robots are still able to receive messages from the leader, the time delay will build up gradually and affect their ability to track the leader. Therefore, a weighting factor to control the follower robot velocity is introduced to ensure faster response, which depends on the end-to-end throughput. Calculating end-to-end maximum throughput can be affected by many factors, however, the focus is on only maintaining a one-hop count, or positioning each

follower robot as an intermediate receiver to the leader robot, which can be accomplished using shape formation algorithm, and routing protocols with low overhead. In Ad Hoc communication, robots typically forward each other's packets when the source and destination nodes of a packet are not within direct reach. Two unicast routing protocols are used, which reduce rebroadcasting overhead and energy consumption by using the velocity of the robot to probabilistically propagate ROUTE REQUEST packets [51]. Sec. 3.7 introduces end-to-end throughput calculation and the main factors that affect it.

3.1 Research Assumptions

The following assumptions are considered:

- Each robot is represented as a point (x,y) in the Cartesian coordinate space. This means that velocity control is assumed to be perfect, robot attributes (e.g. odometry errors) and robot dynamics (e.g. left and right wheel velocities) are ignored since they are not considered in the throughput calculations.
- Robots used in the team are identical in their kinematic model and have the same set of sensors. They are preprogrammed with the same control algorithms.
- The Leader-Referenced model in Leader-Follower formation is used, where other robots maintain the desired position with respect to the leader.
- Each robot can individually avoid detected obstacles using its onboard sensors. Each robot considers the other robots as obstacles as long as they are within a certain distance. The follower robots know the leader's position only, but they are not aware of each other's position.

- Localization is considered solved by using the limited onboard sensors to determine the position and orientation of each follower robot with respect to the leader. From this point of view, it is assumed that each robot may efficiently localize itself without considering the self-localization error.
- A robot has no a-priori knowledge of the environment, except that it is structured and static. It is also assumed that each robot does not have enough memory to depict the information gathered by the sensors into a map. Each robot depends only on the received sensor information to avoid obstacles. No learning is considered.

3.2 Sensors

It is assumed that each robot has a wireless communication capability and is equipped with devices that provide point-to-point signal strength measurements. Furthermore, each robot has an onboard Laser Range Finder (LRF), which is used to measure the distance between the robot and the obstacle. The LRF sweeps the environment ahead of the robot and produce continuous range measurements of the environment, with resolution and communication speed enough for 180° field of view, and with time between the transmission and the reception of the laser beam in milliseconds. Scanning is in 2-D at a maximum distance of 10 meters.

3.3 Shape Formation

The Leader-Referenced model under consideration is applicable to N number of robots, with ability to form various formation shapes. In this work, as a study case of the proposed approach, N = 6 robots, with specified distances and bearings is considered.

At the beginning of the task, each robot is given the following information:

- The Formation Shape Matrix that define the distances and bearings of the followers with respect to the leader [19].
- The leader ID.
- The follower ID.
- The start and goal points.
- Participating team members ID.
- Enough memory to store the required algorithms for path planning and shape formation.

After placing the whole team in a random formation, each robot executes a setup procedure. At the end of the setup procedure the leader unicasts a message with *Next Mode* to all follower robots, then, waits for their acknowledgments. Unless all robots are in the right positions, the leader assumes that there are lost follower(s). The leader sends another message to the follower robots with *Lost_Follower Mode* and ID number(s) of the lost robot(s), triggering the local controller of each follower robot to choose a new formation shape, accordingly. Only the robots that can communicate with the leader adjust their position. After the setup stage, and after accomplishing the primary formation configuration, the leader calculates the local path using the PRM method, then starts

moving to the next local goal point sending *Next Mode* messages with its location, bearing and participating follower robot IDs, every T seconds.

3.3.1 Formation Shapes

As a starting point, the main formation shape is a hexagon shape, which is formed by the followers at the setup stage and in an open/obstacle free area as long as there is no lost robot, otherwise, several other formations are considered to keep the followers at one hop distance in case of narrow passages or a lost robot (leader or follower). Examples are shown in Fig.3.1 and Fig.3.2.

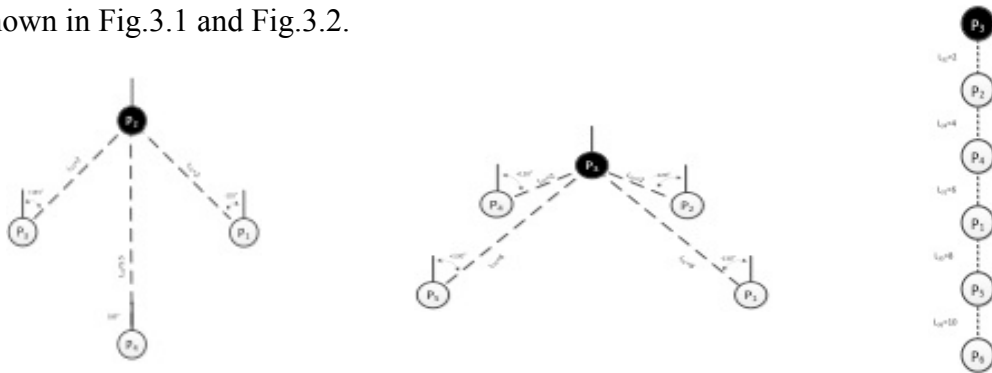


Figure 3.1 Diamond, Wedge and Column Formations

For each formation, each robot has a specific position based on the order of their IDs. Thus, each robot needs to be able to determine its position and angle with respect to the leader. The formations under consideration are:

- Hexagon - the robots travel in a hexagon shape formation, formed at setup stage or while all 6 robots are participating.
- Column - the robots travel one after the other in straight-line formation. The formation is formed when the leader passes through a narrow passage, or when only one follower is left.

- Diamond - the robots travel in a diamond shape formation, as a result of lost robot, and only when 4 robots remaining.
- Wedge - the robots travel in a “V” formation. This formation is applicable for an odd number of robots.

3.3.2 Methodology

To begin with, consider a team of $N = 6$ robots. The team forms a hexagonal formation with configuration vector $P_i = (l_i, \theta_i)$, $P_i \in \mathbb{R}^2$ representing the position of the i^{th} robot, with $i = 1, 2, \dots, N$, as shown in Fig. 3.2. Note that l_i and θ_i are the distance and orientation of robot R_i with respect to the leader ($x_j = l_j \cos \theta_j$, $y_j = l_j \sin \theta_j$). It is assumed that robots are able to sense the proximity of their teammates and/or obstacles within the environment using sensors. Thus, the range and field of the mounted sensor determine the area around P_i .

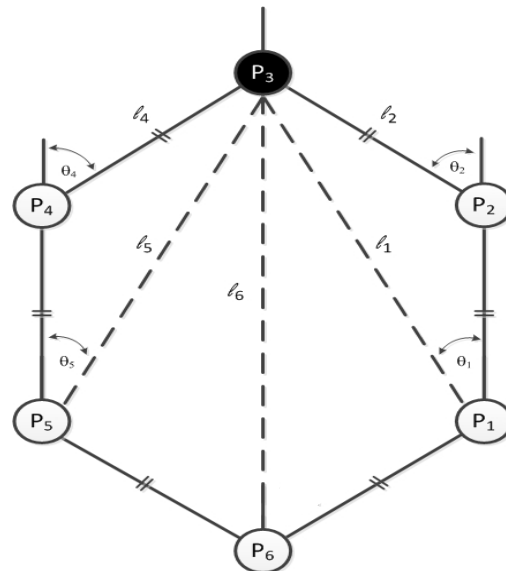


Figure 3.2 Hexagon Shape Formation

For collision and obstacle avoidance purposes, a half circular influence range, such that collision and obstacle avoidance maneuvers are active only when robots are within this range is assumed. When the follower robots find themselves in a non-obstacle free area, they will keep following the leader, but at different angles and distances from the leader compared to the initial ones. This way, the angles between follower robots and leader are not fixed unless the team is moving in an obstacle free area.

In order to represent the formation in Fig. 3.2, the complete team specification is described by means of a formation shape matrix [19] as follows:

$$F_{Hex} = \begin{pmatrix} l_1 & \theta_1 \\ l_2 & \theta_2 \\ 0 & 0 \\ l_4 & \theta_4 \\ l_5 & \theta_5 \\ l_6 & \theta_6 \end{pmatrix}$$

Row i , with $i= 1, \dots, N$, describes the place P_i of each robot in the formation. The formation shape and the Leader-Follower order for the complete team are both described by the formation matrix. Therefore, all the robots in the team must have a-priori knowledge of this matrix, and then during mission execution, each follower robot just needs to know the pose of the leader [47]. This is primarily for maintaining followers at one-hop distance from the leader, to achieve a maximum link throughput, which is going to be the main factor in the controller simulation, and also, will reduce the inter-robot communications.

3.3.3 Formation Control Algorithm

The key idea is simple: every robot in the team positions itself relative to the leader robot. As previously mentioned, the configuration of ($N = 6$) robots is given by the shape matrix, where all robots are a-priori given a comprehensive list of the definitions of all shapes that the group may need to establish. Each robot has a unique ID included in its control code. At the setup stage, or in case of failing robot, all robots will use Algorithm 1, see next, a modified and extended version of the work in [4][47], to produce the formation shapes, accordingly. The robot with the median ID value will be the leader and, thus, in front of the formation. During setup stage or mission execution, each robot calculates its proper position in the formation based on the leader location and the requested shape from the leader. Robots with IDs less than the leader ID adjust and position themselves by the order of their IDs on the right side of the leader, while robots with IDs greater than the leader ID adjust and position themselves by the order of their IDs on the left side of the leader, as shown in Fig. 3.1 and Fig 3.2. The algorithm used as long as no obstacle is detected is presented next.

3.3.4 Algorithm 1, Formation shape control and leader election

1. Team = sort(Team)
2. Check Mode, (Lost_Follower, Lost_Leader, Next, Narrow)
3. If Mode = Lost_Leader
 - i. For odd: Leader ID = median index of(Team)
 - ii. For even: Leader ID = floor(median index of(Team))
4. End if
5. Choose formation matrix accordingly
6. If robot ID > Leader ID

- a. on left side of leader
- b. $[P,H] = F_{matrix}(\text{robot ID} > \text{Leader ID})$
- 7. Else
 - a. right side of leader
 - b. $[P,H] = F_{matrix}(\text{robot ID} < \text{Leader ID})$
- 8. End if

During the algorithm execution, if a robot detected an obstacle, it will interrupt Algorithm1 and will start following the obstacle's border until the robot avoid it entirely. This leads each robot to position itself with respect to the leader at different distance and orientation than the one the formation matrix specified it. When the robot finds itself again in a direct range with the leader, it will resume the algorithm process.

3.3.5 Failing or Lost Robot scenarios

As in many cases, one or more robots may be lost due to sensors failure or battery life, or even in the meaning of physically lost, like being "kidnapped" or fallen in a hole; all these cases can affect the formation of the team. The following cases are considered:

I. Failing Follower(s):

In this case, a similar procedure to Algorithm 1 is used with the check condition depending on whether the number of team members is odd or even. If the number of the remaining members is odd, the new formation becomes a wedge formation, otherwise, the new formation becomes a diamond formation (4 robots), or line formation (2 robots). For example, consider that after the setup stage the leader ID = 3, the members are 6 and formation is hexagon shape. If follower robot 5 failed, the remaining members become 5

robots (odd number) and the next shape will be a wedge, where followers 1 and 2 will be on the right side of the leader, on the left side will be followers 4 and 6. See Fig. 3.3.

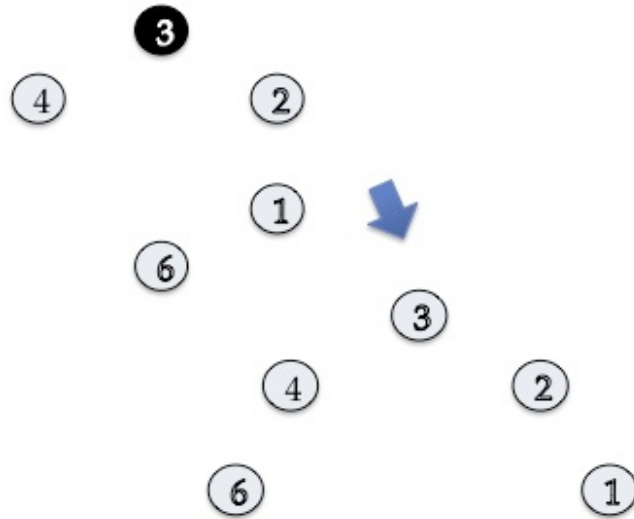


Figure 3.3 Failing Robot #5, wedge formation

II. Failing Leader:

To overcome this problem and to deal with the possibility of a failing leader robot more than once, all robots in the team will be provided with the path planning algorithm as a backup strategy. During mission execution, only the leader makes use of the path planning algorithm.

According to the adopted algorithm, in case the leader robot failed, any other robot can potentially serve as the leader. Who is leading depends on the number of robots and the resulted formation shape.

After a specific period of time, if the followers did not receive an update from the leader, either directly or forwarded by other teammate(s), the followers will enter

Lost_Leader Mode individually, where they have to choose a new formation shape and new leader accordingly.

When the follower enters *Lost_Leader* Mode, each follower robot individually will check the last sent information by the leader and will look for the number and IDs of remaining followers. Each follower robot executes the same code individually, and automatically agrees on the same new leader.

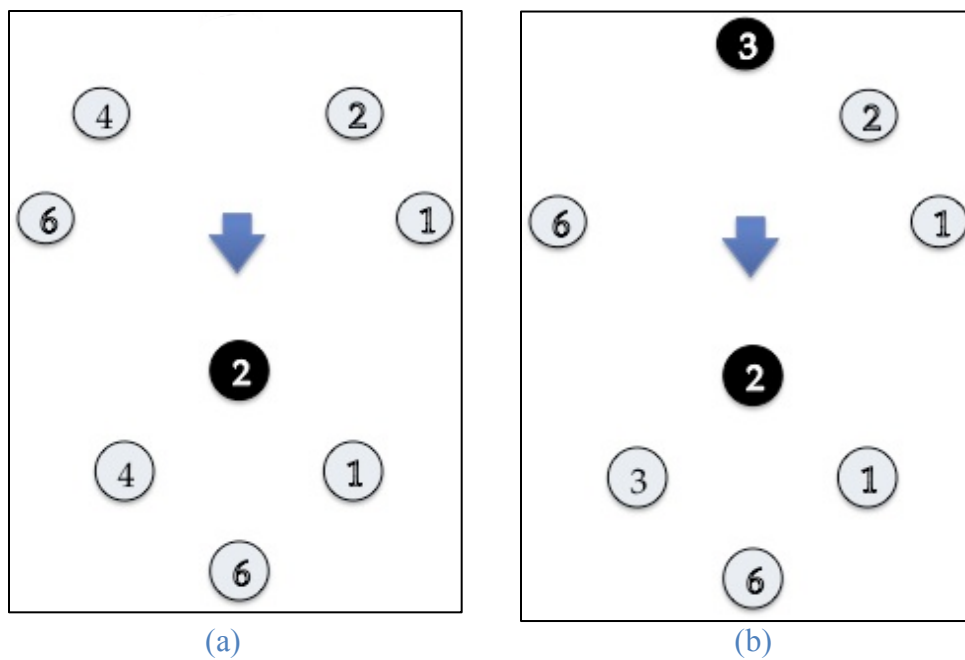


Figure 3.4 Leader election : (a) Failing Leader Robot #3, diamond shape, (b) Failing Follower Robot #4, new leader robot #2, diamond shape

For example, the last message sent by the leader robot (ID = 3) to the followers, before it was considered lost, indicated that the participating follower IDs were 1, 2, 4, 5. Each one of these followers will execute Algorithm 1, and will elect robot 2 as the new leader of the team, plus the new formation shape is diamond, as shown in Fig. 3.4(a).

The new leader, robot 2 will start sending *Next* messages, and will plan a new trajectory to the goal point from its current position, while the rest of the followers shape themselves around the leader, waiting for the new goal position.

Moreover, if the failed follower affects the balance of the formation shape, the robots would elect a new leader, even if the leader did not fail, to recover the formation balance, as illustrated in Fig. 3.4(b).

3.4 On-Line Path planning and Obstacle Avoidance

In this section we present a modified version of the PRM approach [26][41]. The leader robot is a car-like robot represented point, on which a Laser Range Finder is attached to. The robot is given the task to move from a beginning configuration in the 2-D space to the goal point, without a-priori knowledge of the environment. The path-planning algorithm is available and preprogrammed in every robot code, but only the leader executes it.

3.4.1 Methodology

The framework of the PRM planning algorithm consists of two stages [26][41]: roadmap construction (learning) and query. In the learning stage, the algorithm generates random points from the leader view field, then by using the sensor's readings it will keep the points within an obstacle-free area and discard the rest. The algorithm then constructs a probabilistic roadmap by connecting the points using a simple, but very fast motion planner, also known as a local planner [26][41]. Then, the connected points (known as

roadmap) are stored, with the edges representing the possible path. In the query stage, the nearest two points to the start and goal points are selected from the roadmap, known as virtual start and virtual goal, respectively. Because the leader has no a-priori knowledge of the environment except that it is structured and static, the virtual goal point is the furthest random point in the leader sensor range aligned with the goal direction. After selecting these two points, the planner searches the roadmap to find a sequence of edges connecting those points. Only the edges lying in obstacle free parts generate a feasible path for the leader between the virtual start and virtual goal points. When the leader reaches the virtual goal point, it will consider it as a new start point, and then the PRM algorithm repeats the same process till the leader reaches the goal point.

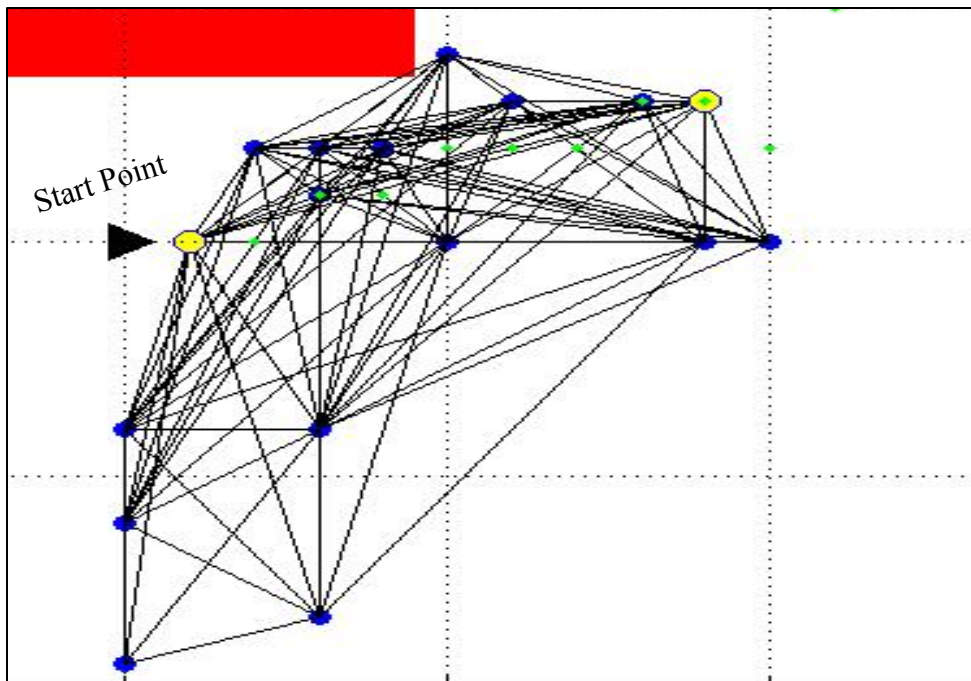


Figure 3.5 Random points in leader's field of view, virtual start and virtual goal points selected

PRM planners [26][41] can efficiently capture the configuration space, C , of large and complex environments. In the problem under consideration, $C \in \mathbb{R}^3$ corresponds to the randomly generated points, x_i and y_i and angle θ of the leader, which define the sensor field range. Here θ is selected at the beginning of each movement to align the robot to the direction of the goal point. This ensures that the leader is moving forward, unless the sensor detects an obstacle.

One of our contributions in this thesis is limiting the local free region for the leader to sample the random points from if an obstacle is detected in 3m range or less, although the sensor has a maximum range of 10m. When an obstacle is detected in 3m range or less of 180° view field (minimum range of 3m is assumed from the robot point to detect an obstacle, considering that every robot type has different dimension), the algorithm first has to decide if the detected obstacle is a front, left, right or corner obstacle, then it will either follow the obstacle border or change θ , hence, the range field used to generate random points will change accordingly, as shown in Fig.3.6. This step is very vital to avoid trapping the leader into an infinite loop or local minima, and ensuring the efficiency of the paths selected by decreasing the path cost for each step. The implemented algorithm, a modification and improved version of the algorithm in [26], is summarized in Algorithm 2, where C_{free} is the obstacle-free subspace of C , and (x_d, y_d) denotes the desired final position.

3.4.2 *Algorithm 2, Modified PRM for Unknown Environments*

1. Goal, Start
2. $C_{free} = []$; Path_Solution=[];
3. Path_Solution = Path_Solution \cup Start

```

4. while xi ≠ xd and yi ≠ yd do
    a. if sensing obstacle in 180° field
        1. While still obstacle in range check
            1. If "left corner" →  $\theta = \theta - 90$ 
            2. If "right corner" →  $\theta = \theta + 90$ 
            3. If "front obstacle" →  $\theta = \theta + 180$ 
            4. If "left or right obstacle" → follow the wall
            5. If "left and right obstacle" → choose mid point
        2. End while
    b. Else
        1. LRF view range is 180° and move forward.
    c. end if
    d. Generate N random points in the sensor range field
    e. C = [(x1-N; y1-N)]
        1. if testPath(p) then
            a. Cfree = Cfree ∪ p
            b. E = generate edges connections with (xi ; yi ) to
                Cfree.
            c. VirtualGoal = nearest point to Goal ∈ Cfree
            d. VirtualStart = nearest point to Start ∈ Cfree
            e. L_path = new local path returned by A* search
                from VirtualStart to VirtualGoal
            f. Path_Solution = Path_Solution ∪ VirtualGoal
        2. Else
            a. Path_Solution = Path_Solution-1, Go back one step
        3. end if
    5. end while

```

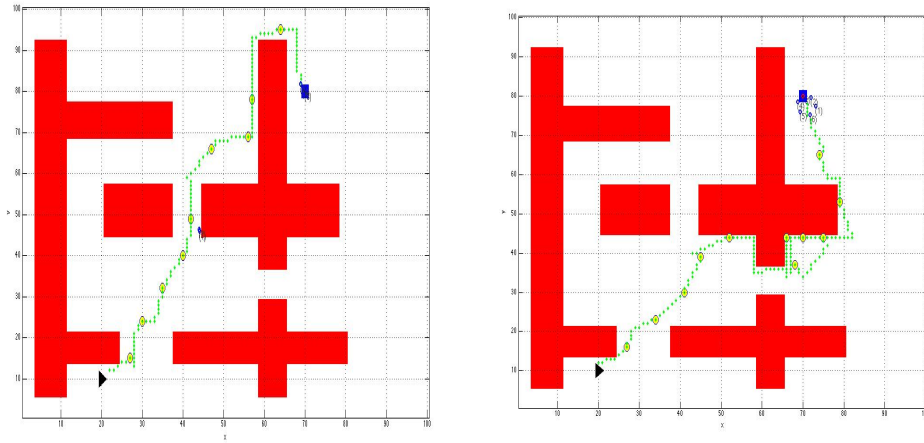


Figure 3.7 Different path trajectories using PRM

From the above algorithm, the generated path from the start point to the goal point is a different trajectory each time the algorithm is executed, as shown in Fig. 3.7.

Consider that the leader robot moves according to the calculated trajectory, without pause and with speed V_L m/s. Every T seconds, the follower robots receive the current location of the leader robot as their target to track, while avoiding obstacles at the same time. After sometime, the formation starts to be loose, and the distances between the leader and followers will differ from F_{Hex} , until some point, the followers would not be able to receive the leader messages directly. If each follower robot kept the distance from the leader fixed, the formation would be rigid and would cause robot collisions. To make the formation more flexible, giving the movement priority to obstacle avoidance while maintaining minimum communication, the throughput level should be calculated for each *Next Mode* message at the follower end, if there is no obstacle in the field-of-sense, each follower robot changes its speed, to compensate for the distance difference, and move to the desired separation. Thus, every T seconds, the followers update their

speeds and destinations while continuously tracking the leader node. This process is repeated until the team reaches the final goal point. Therefore, the major critical task is to derive a control methodology for the followers to compute their desired linear and angular separation with their leader to remain in the defined formation topology.

In this case, the control problem becomes a path tracking control problem for each follower, where each follower plans its path to efficiently position itself relative to its leader by observing the leader's pose. Hence, the tracking controller should be designed for the followers to maintain a certain level of communication link quality with the leader. The goal of the tracking controller is to find the velocities of the followers, based on their links quality with the leader, is the goal of the tracking controller.

3.5 Tracking controller

The objective of the tracking controller, is to calculate the values of the translational and rotational velocities, v_F and ω_F , respectively, of the followers in such a way that the formation/separation errors (linear and angular) converge to zero, and position each follower robot in the desired geometric pattern with the leader robot.

The tracking controller equations used in this work are taken from [25]. To explain the math behind this type of formation controller, consider a simple system consisting of two robots in a leader-follower formation, and assume that the pose vectors of both robots are given in the Cartesian coordinate space as shown in (3.1). After receiving the pose of the leader, the follower will determine its desired pose with respect to the leader, then treat this pose as the next target (goal) as shown in Fig. 3.8:

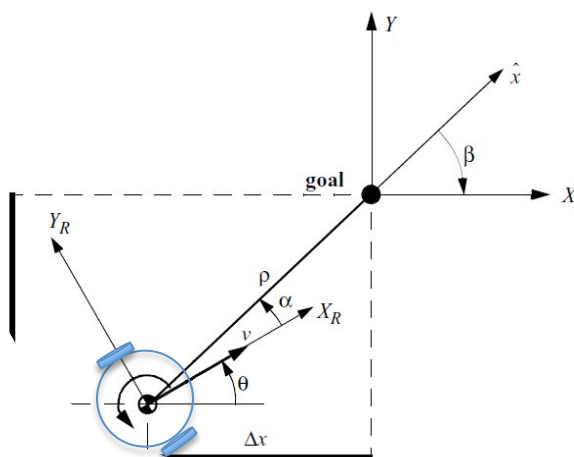


Figure 3.8 Follower kinematics

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (3.1)$$

K is the control matrix, and v and ω would drive the difference error between the current and the desired positions to zero :

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (3.2)$$

The follower robot dynamic model inputs is given by [25] as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.3)$$

As shown in Fig. 3.8, α is the angle between the x_R axis of the robot's reference frame and the line connecting the center of mass of the follower with the target position (goal).

If $\alpha \in I_1$, where:

$$I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2} \right] \quad (3.4)$$

then, the polar coordinates transformation with its origin at the goal position [25].

$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \quad (3.5)$$

$$\alpha = -\theta + \text{atan2}(\Delta y, \Delta x) \quad (3.6)$$

$$\beta = -\theta - \alpha \quad (3.7)$$

Using matrix representation:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos \alpha & 0 \\ \frac{\sin \alpha}{\rho} & -1 \\ -\frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.8)$$

ρ is the distance between the center of mass of the follower and the goal position.

θ is the follower heading.

On the other hand, if $\alpha \in I_2$, where:

$$I_2 = \left(-\pi, -\frac{\pi}{2}\right] \cup \left(\frac{\pi}{2}, \pi\right] \quad (3.9)$$

by setting $v = -v$, we obtain a system described by the following matrix form [25]:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ -\frac{\sin \alpha}{\rho} & 1 \\ \frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.10)$$

v and ω are the linear and angular velocity, respectively as previously said. Now,

to get the closed loop control system equations, we substitute:

$$v = k_\rho \rho \quad (3.11)$$

and:

$$\omega = k_\alpha \alpha + k_\beta \beta \quad (3.12)$$

into (3.8). The velocity equations of the follower robot is given by:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_{\rho}\rho \cos \alpha \\ k_{\rho} \sin \alpha - k_{\alpha}\alpha - k_{\beta}\beta \\ -k_{\rho} \sin \alpha \end{bmatrix} \quad (3.13)$$

For stability issues, work in [25] has proven that the system is stable if:

$$k_{\rho} > 0; \quad k_{\beta} < 0; \quad k_{\alpha} - k_{\rho} > 0 \quad (3.14)$$

To satisfy the previous equation, and for the best results, authors of [25] suggested to set the control parameters $(k_{\alpha}, k_{\beta}) = (8, -1.5)$. As we can see from (3.11), v is a function of the distance ρ between the leader and follower. Since the objective is to maintain the team formation shape while keeping minimum acceptable communication links quality, this may be done when the time delay between the leader robot and the follower robot is controlled. Therefore, the robot velocity is controlled by changing the value of k_{ρ} based on the link quality.

To maintain the formation, hence the network connectivity, each follower robot continuously monitors the end-to-end throughput of the link to the leader robot. When the link throughput drops below a minimum acceptable threshold, the controller increases the velocity control factor to $k_{\rho\max}$, quickly moving the follower robot to its targeted position with respect to the leader robot, until the throughput returns to an acceptable level. On the other hand, if the robot recognizes an increase in its link throughput above the acceptable level, it will attempt to decrease the velocity control factor to $k_{\rho\min}$, or do nothing if the factor already is at $k_{\rho\min}$. This process reduces the delay time a follower robot was caught in because of the link throughput drop, despite how close the follower is to the leader,

which would cause network dis-connectivity in the long run. Furthermore, these processes also ensure that a follower robot is constantly maintaining its distance with respect to the leader robot while keeping the constraints [50]

3.6 Ad-Hoc network

In this work, acceptable communication specifications are chosen according to the DSSS PHY and MAC (IEEE 802.11b, 1999), CSMA/CA RTS CTS access mechanism. MRSR and MRDV are considered as routing protocols [51], which are based on Dynamic Source Routing (DSR) and Ad-Hoc On Demand Distance Vector Routing (AODV) respectively. Refer to [43] for more information on the energy model and measurements for Ad Hoc networks.

3.7 Throughput

As defined in [42], the time average of the number of bits that can be transmitted by each node to its destination is called the per-node or end-to-end throughput. The sum of per-node throughput over all the nodes in a network is called the throughput of the network.

According to [46], mobility has effects on the throughput of neighbor robots. In system under consideration, we keep the network connected by maintaining the distance between the leader and the followers with throughput check. Assuming ideal network conditions, and an error free channel, the theoretical maximum throughput is defined as “the ratio of the spent time in transmission of successfully received packets to the maximum time available for transmission” [36]. Fig.3.9 shows the exchanged data frame

sequence of CSMA/CA protocol in case of RTS CTS mechanism for 802.11b standards [36]. The MAC Service Data Unit (MSDU) size is set equal to 800 bytes in the simulation, and treated as the payload at MAC layer for calculation purposes [36]-[40].

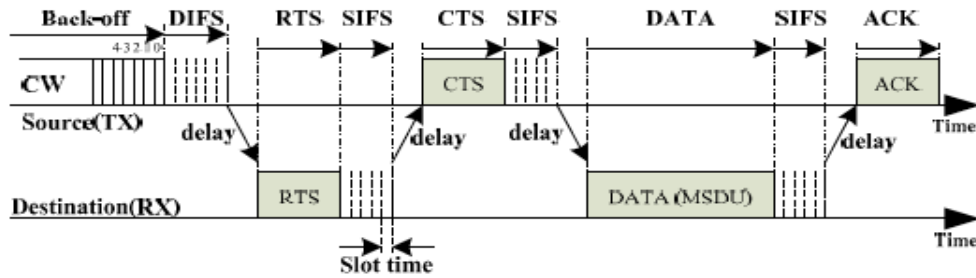


Figure 3.9 Standard Timing of CSMA/CA [36]

Theoretically, the maximum throughput of CSMA/CA RTS CTS protocol in IEEE802.11a/b/g standard is given by [36]:

$$\text{Theory throughput} = \frac{\text{Transmitted data (MSDU)}}{\text{Successful transmission time}} = \frac{8 \times \text{MSDU}(\text{size})}{T_{S[\text{CSMA}|\text{CA}-\text{RTS}|\text{CTS}]}} (\text{bps}) \quad (3.15)$$

Where:

$$T_{S[\text{CSMA}|\text{CA}-\text{RTS}|\text{CTS}]} = T_{\text{RTS}} + 2T_{\text{SIFS}} + 3T_{\text{delay}} + T_{\text{CTS}} + T_{\text{MSDU}(\text{size})} + T_{\text{DIFS}} + T_{\text{Backoff}} \quad (3.16)$$

And $T_{C[\text{CSMA}|\text{CA}-\text{RTS}|\text{CTS}]}$ is the Collision transmission time:

$$T_{C[\text{CSMA}|\text{CA}-\text{RTS}|\text{CTS}]} = T_{\text{RTS}} + T_{\text{delay}} + T_{\text{DIFS}} + T_{\text{Backoff}} \quad (3.17)$$

T_{RTS} , T_{CTS} , $T_{\text{MSDU}(\text{size})}$ and T_{ACK} can be calculated as follows:

$$T_{\text{RTS } 802.11b} = T_{\text{premeable}} + T_{\text{PLCPheader}} + \frac{8L_{\text{RTS}}}{\text{Data Rate}} \quad (3.18)$$

$$T_{\text{CTS } 802.11b} = T_{\text{ACK } 802.11b} = T_{\text{premeable}} + T_{\text{PLCPheader}} + \frac{8L_{\text{ACK}}}{\text{ACK Rate}} \quad (3.19)$$

$$T_{\text{MSDU } 802.11b} = T_{\text{premeable}} + T_{\text{PLCPheader}} + \frac{8(L_{\text{MACheader}} + \text{MSDU})}{\text{Data Rate}} \quad (3.20)$$

$$T_{\text{Backoff}(\text{average})} = \frac{W_{\text{min}} \times T_{\text{slot}}}{2} \quad (3.21), \quad W_{\text{min}} \text{ is minimum contention window}$$

$$T_{delay} = \frac{T_x \text{ to } R_x \text{ distance}}{\text{Radio waves propagation speed}} \quad (3.22)$$

The delay time, T_{delay} depends on the distance between the sending and receiving robots. When the follower robot receives the leader's location, it calculates the throughput of this transmission and decides if it is lower or above the acceptable threshold. With minimum throughput acceptable threshold known to each follower robot, the distance between the sending and receiving robots becomes flexible. Before we consider using the level of throughput as an initiator to formation controller, there are some factors need to be addressed:

3.7.1 Signal Strength and Bit Error Rate

Signal strength between leader and a follower robot is a function of the transmission power, antenna gains, and signal attenuation. In general, the received signal strength can be used as a measure of the connection reliability, while link throughput can efficiently be used to ensure minimum communication link quality. The shadowing path loss model studied in [33] is considered in the simulation, since it considers multi-path propagation effects due to obstacles. It represents more realistic situation than free space and two-ray path loss models. Due to the fading phenomena, at a certain distance, the received power level is a random variable. Therefore, shadowing model can be represented as an Additive White Gaussian Noise (AWGN) in the simulation. According to the radio propagation theory [33], the radio signal attenuates fast when the propagation distance increases, leading to low SNR at the receiving end and low end-to-end throughput.

For the performance of the system we also study the bit error rate (BER), which quantifies the impact of the packet loss. BER of any communication system is defined as the ratio of number of error bits and total number of bits transmitted during a specific period [59]. For any given modulation, the BER depends essentially on the strength of received signal; hence, it is normally expressed in terms of signal to interference and noise ratio (SINR). I also depends on the modulation data rate [59]:

$$SINR = 10\log_{10} \left(\frac{P_{recv}}{N_0 + \sum_{i=1}^N I_i} \right) \quad (3.23)$$

For DPSK modulation technique:

$$BER(SINR) = 0.5e^{-\frac{E_b}{N_0}} \quad (3.24)$$

$$\frac{E_b}{N_0} = SINR \cdot \frac{1}{\# \text{ bit ber packet}} \quad (3.25)$$

E_b is energy per bit and N_0 is noise power.

3.7.2 Routing Protocols:

Two unicast routing protocols are applied. These protocols designed for use in Ad Hoc networks formed by mobile multi-robot teams [51]: Mobile Robot Distance Vector (MRDV) and Mobile Robot Source Routing (MRSR). Both protocols present an efficient routing technique by minimizing needed overhead and power consumption.

These protocols are considered as modifications to reactive routing protocols (DSR and AODV). Proactive protocols such as DSDV [44] and OLSR [45], continuously exchange routing table updates to maintain routes that lead to high-energy drain. Reactive protocols avoid the need to actively maintain routes until they need to transfer data. That

is why they are preferred in robot networks with limited resource and the need to communicate only occasionally in the duration of executing their mission.

The MRSR is based on Dynamic Source Routing (DSR) [48]; when the sender and receiver are not in range, the technique reduces routing overhead and energy consumption [51] by limiting the forwarding of ROUTE REQUEST packets if robots are moving. Once the follower robot receives the ROUTE REQUEST, it forwards the ROUTE REQUEST with a probability of p_r , where p_r is calculated using a combination of the follower robot current velocity v and the distance weighting factor γ of the robot as shown below [51]:

$$p_r = \min\left(1, \left(\frac{1}{v}\right)^\gamma\right) \quad (3.23)$$

γ is a function of the remaining distance for the robot to reach the goal point. While this is an unknown factor, γ is usually very large, hence, p_r is very small.

The second unicasting protocol is MRDV, which is based on the AODV routing protocol [49]. MRDV utilizes the use of hop-by-hop routing and destination-based sequence numbers with another Ad Hoc routing protocol DSDV [44]. MRDV shares on-demand behavior with the MRSR and probabilistically forwards ROUTE REQUEST packets. However, MRDV stores routing information as one entry per destination in the routing tables, in contrast to MRSR, which caches multiple entries per destination. In both protocols, if a route discovery fails, a normal route discovery (without probabilistic rebroadcast) is sent out to ensure connectivity [51].

3.8 Discussion

3.8.1 Shape Formation

- In case of lost connection between the leader and one of the followers due to hardware or technical difficulties, after waiting a specific period of time, the follower enters *Lost_Leader* mode and elects a new leader. If after the specified time passes with no messages from the new leader, the follower robot realizes the problem and either stops, or calculates the trajectory path to the goal point (become a leader) and starts moving.
- The formation shape is considered for maximum $N = 6$ robots. In case of $N > 6$, the method still works as long as the formation matrix is defined accordingly. The setup configuration could be a hexagon shape too, to cover as large as area as possible, but at the same time, to have maximum throughput for each follower robot, which is not the case for the column shape.
- One of the advantages of the Leader-Referenced approach is scalability and tolerance to new member additions. In this work, the effect of adding more robots to the team is not considered. Note that for large N unicasting is not a practical data transmission.
- In case of 3-D formations, the considered formation control algorithm in this work can still be implemented if the team continuously maintained the exact height, although, new constraints should be introduced when it comes to the obstacle avoidance maneuver. When considering different altitudes the formation matrix must be exchanged to consider spherical coordinates, at the least.

3.8.2 On-Line Path Planning

- In PRM path planning, the only factor that affects the mission execution time is the path itself. The leader robot moves with a constant velocity without pausing or waiting for the rest of the team to catch up. Therefore, the size of the team has no effect on the chosen path or the execution time.
- At the end of the mission, when the leader reaches the goal point and has no more pose information to send to the followers, all the follower robots in the maintained formation will stop automatically.

3.8.3 Ad Hoc Network and Throughput Calculations

- The leader communicates with the followers through unicasting instead of broadcasting, to reduce the number of forwarded messages especially when the used unicasting protocols are tailored for this purpose.
- Considering each follower robot has different separation distance from the leader, the throughput minimum threshold is slightly different from follower to another; however, they all are about 0.5333 Mb/s, calculated in ideal conditions.

CHAPTER 4

SIMULATION STUDIES

A MATLAB-based simulator has been developed for simulation studies. By using some of the Robotics Toolbox functions [34], the *Formation Module* is constructed to simulate Leader-Follower robot formations, where the leader robot follows the path provided by the *Path_Trajectory Module* from *start* point to *goal* point, while five followers receive leader position updates (*Next Mode* message), and try to track it using the end-to-end throughput as a system input keeping a specific formation shape. The *Ad_Hoc Module* simulator used is based on [35]. The mission environment is 100mX100m obstacle map with three different configurations. See Fig.4.1 as an example. Because of the obstacle inflation, there is an extra 0.5 m added to the border of the obstacle, which is not included in the map, as shown in Fig. 4.3 (a), robot 6 is actually following the obstacle border, not inside it!

4.1 On-line Path Planning (PRM)

We implemented the local tracking controller in simulation with position updates as in (3.13), mobility control inputs given by (3.11) and (3.12), where the leader and supporting team move from start point to goal point in unknown environment with minimum inter communication and low sensing capabilities.

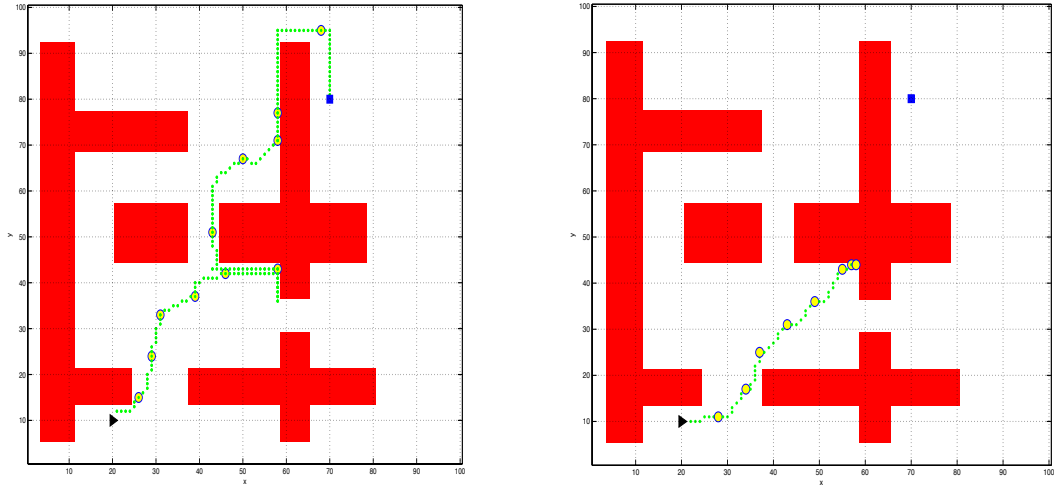


Figure 4.1(a) Enhanced PRM finds the goal point in short time. (b) Original PRM [26] did not solve local minima problem within same time.

When applying Algorithm 2 from Chapter 3, the leader robot moves to the goal point with minimum number of steps, or at least did not get stuck in a local minima for a long time. The basic idea of this path planning is to calculate and follow the shortest path to goal, in other words, giving the priority to path optimization. If local minima were in the way, the algorithm switches priority to effectively guide the leader out of the local minima even if the alternative means generating new random points in the opposite direction of the goal. Fig. 4.1(a) illustrates an example of applying Algorithm 2.

The black triangle and blue square denote the start and goal points, respectively. The yellow circles indicate the next Virtual Start and Virtual Goal for the Leader, which are at 10m maximum distance. In this case, when it comes to shortest distance to the goal, the leader may get trapped in the right upper corner as shown above, then the algorithm steers the leader away. It may take several seconds for the leader to find its way out of

that corner. In the original version of the algorithm [26], if the robot is stuck in a local minima while giving short distance a priority, implementing line 26 in the algorithm alone (where leader takes a step back on the calculated path, when there is no more connected and obstacle free steps) may consume all the random points connected to the initial point that were generated in the front 180° view field, and the robot may find itself out of choices and stops at the end, as shown in Fig. 4.1(b).

The draw back from this enhanced algorithm is that the planner is unable to find the way out in case a robot is stuck in a three wall obstacle, with less than 10m width, as shown in Fig. 4.2.

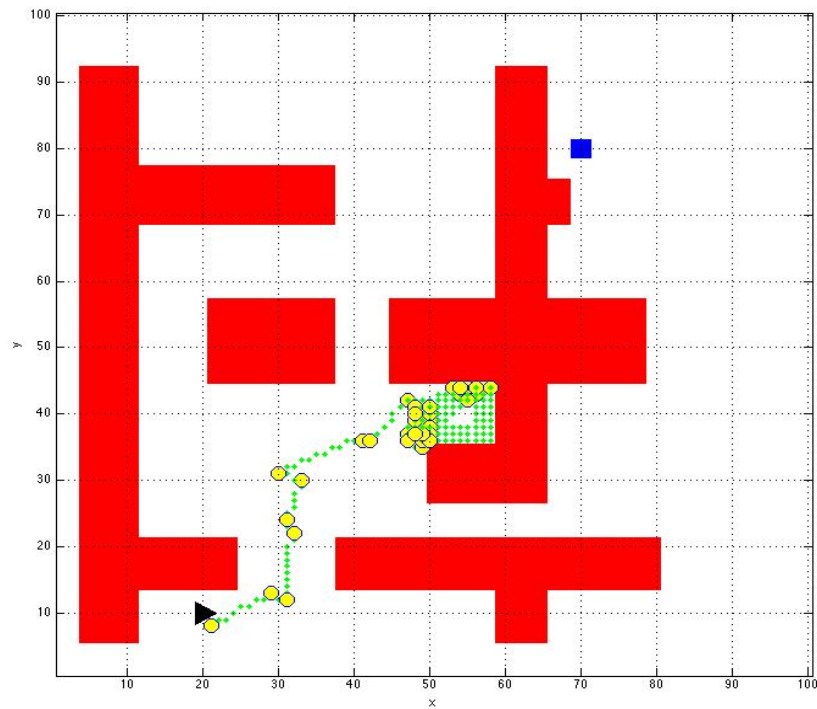


Figure 4.2 Planner got stuck in 3-Walls obstacle.

4.2 Throughput Calculation

The follower robots must maintain the distance from the leader at a specified speed with a constant control factor. Because of any changes in the environment, the follower robot may get delayed trying to arrive at the last location sent by the leader. This may cause the communication link to break triggering *Lost_Follower* mode message from the leader to the rest of follower robots. This may, in turn, change the team formation accordingly. The local controller simulator gives each follower robot the ability to detect any drop in link throughput, and to react by increasing its velocity, thus, getting closer to the leader robot until reaching an acceptable throughput level. If the speed is minimum while throughput is within an acceptable range, the followers maintain the speed. Note that speed is either minimum or maximum.

Given power limitations, it is essential to keep the coordination communication between the followers and the leader robot to the minimum. In this simulation, acceptable communication specifications are chosen according to the DSSS PHY and MAC (IEEE 802.11b, 1999), CSMA/CA RTS CTS access mechanism. The simulation is conducted using the MRSR and MRDV routing protocols [51], which are based on Dynamic Source Routing (DSR) and Ad-Hoc On Demand Distance Vector Routing (AODV) respectively. Both protocols are compared and evaluated based on end-to-end delay, while keeping the MAC service data unit size fixed of MSDU = 800 bytes.

The following figures illustrate the throughput levels for a data rate of 1 Mbps with constant and variable velocity control factor. The throughput minimum threshold is slightly different from a follower to another, depending on its location in the team

formation. However, they are all about 0.5333 Mb/s, calculated under ideal conditions. Because of their lower control overhead, when it comes to performance comparison between the two protocols, both MRDV and MRSR have almost a similar performance. MRSR has a slightly lower overhead than MRDV due to the use of caching. However, MRSR also has a slightly higher delay than MRDV due to the use of larger packets that include source routes [51].

Using (3.15), all the times are calculated assuming a full RTS, CTS, DATA, ACK exchange. These calculated times also include an estimate of the time spent backing off during contention. We also plot the BER and the level of SNR as a function of receiving power, to indicate link connectivity and packet loss, with receiver SNR threshold at 32 db.

The simulation is implemented using three different maps, map1, map2 and map3. Assuming ideal mission conditions in these maps, like if all followers track the leader perfectly without pausing, miss calculating next position or having any effects on the throughput, all the followers arrive at the goal point with occasional throughput drops. Therefore, to test the controlling algorithm, we simulated a delay time for the follower robots, like when they avoid an obstacle while trying to keep the formation shape for example. Also, for comparison purposes between both protocols, a random path trajectory has been calculated by the leader for each map separately, then saved to a .mat file to be used several times with that map in both cases of k_ρ . ($k_{\rho_{\max}} = 3$. Higher values of k_ρ can cause the controller to oscillate).

4.2.1 Map1

- **Using the MRSR Protocol:**

First we applied the simulation using low and constant controller factor k_p . The follower robots tracked the leader perfectly until second 60 when the delay period was imposed. Then robot 6 started to fall behind and finally stopped. The rest of the team changed the formation to wedge shape as shown in Fig. 4.3(a). The throughput level at follower 6 end point started to drop due to the increasing separation distance. Since the controller at this point is not reacting to the change in the throughput, the follower will maintain an increasing velocity based only on the separation distance, leading to higher BER and lower SNR as shown in Fig. 4.4(a) and Fig. 4.5(a).

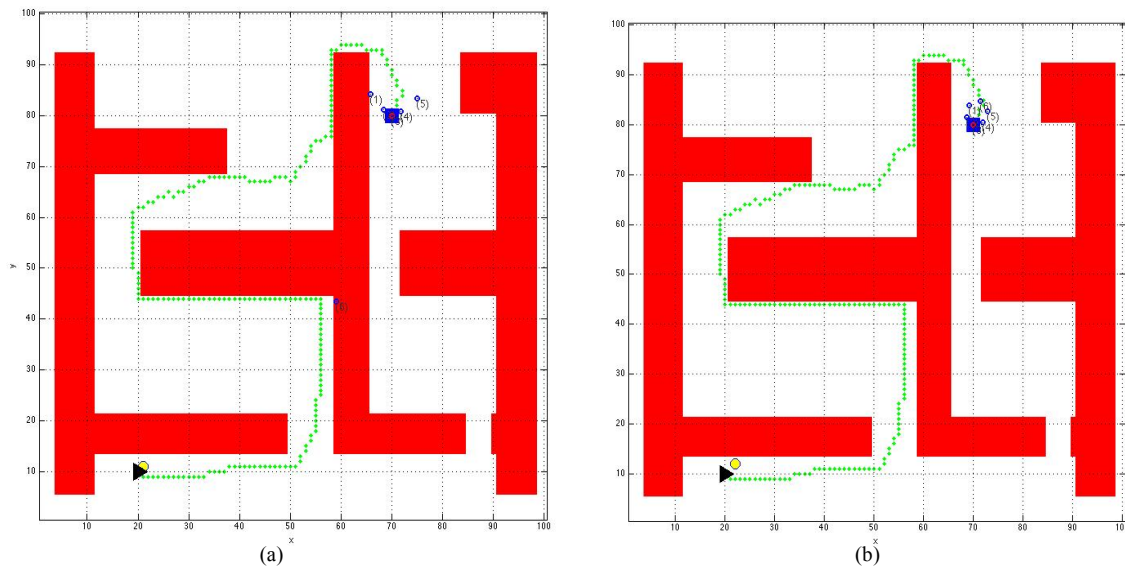


Figure 4.3 Mission executions in map1 with (a) Constant low k_p . (b) Variable k_p

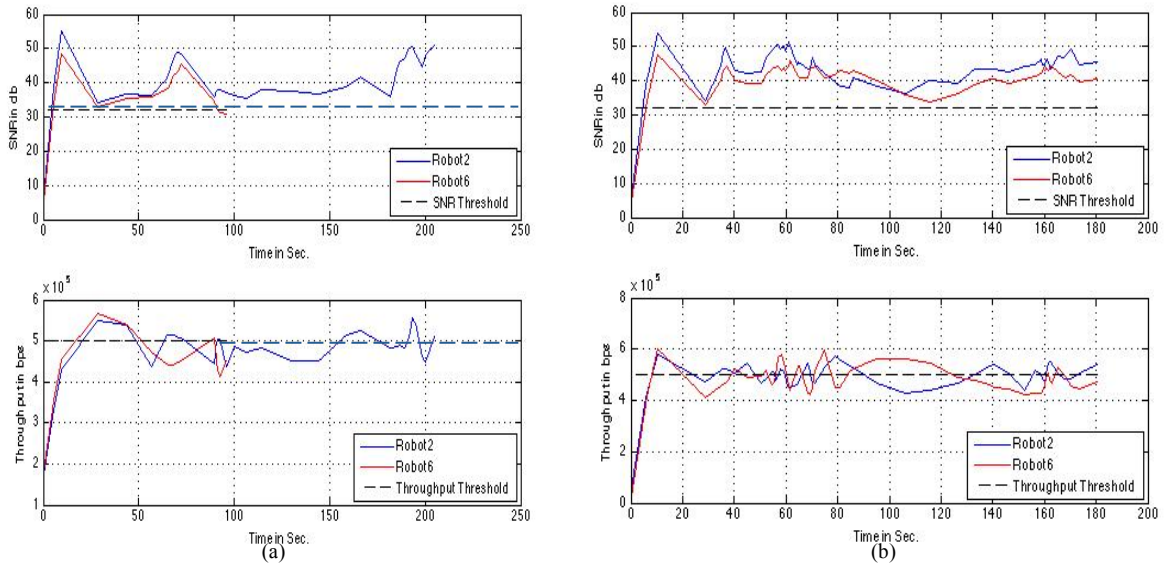


Figure 4.4 SNR and Throughput levels with (a) Constant low k_ρ . (b) Variable k_ρ

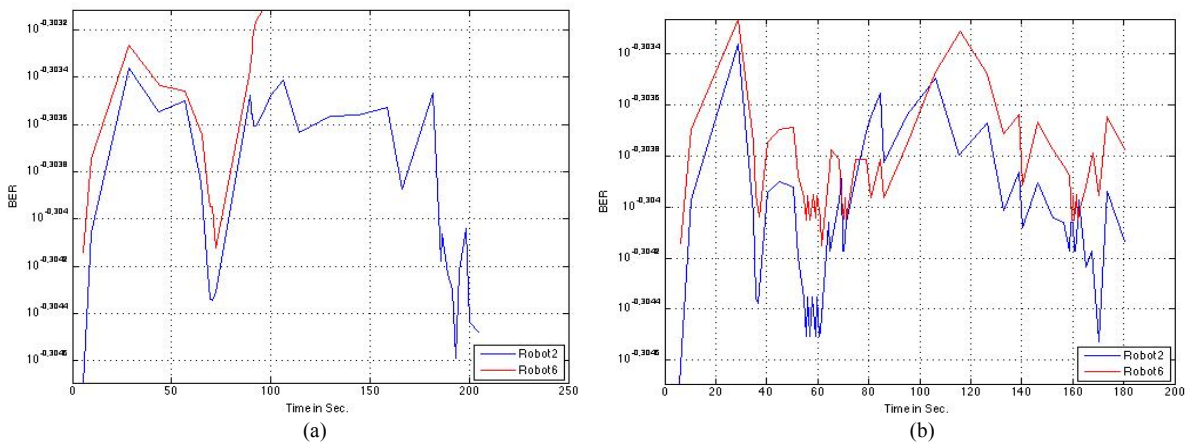


Figure 4.5 BER levels with (a) Constant low k_ρ . (b) Variable k_ρ

Then, we applied the same path trajectory and with the same delay period, but with variable controller factor k_ρ as a function of the measured throughput. The entire team successfully arrived at the goal point as shown in Fig. 4.3(b). We can see from Fig. 4.4(b) that whenever the level of throughput dropped below the threshold level, it will increase almost immediately due to the moving the follower faster and closer to

the leader. Same results inferred from Fig. 4.5(b), where decreasing the separation distance between the leader and the followers decreases the BER levels.

- **Using the MRDV Protocol:**

Following the same procedure when using the MRSR protocol, the simulation produced the same results for this map, map1. Robot 6 was lost in the case of low constant k_ρ , then the formation performance improved when using variable k_ρ , Fig 4.6 - 4.8 illustrate the simulation results.

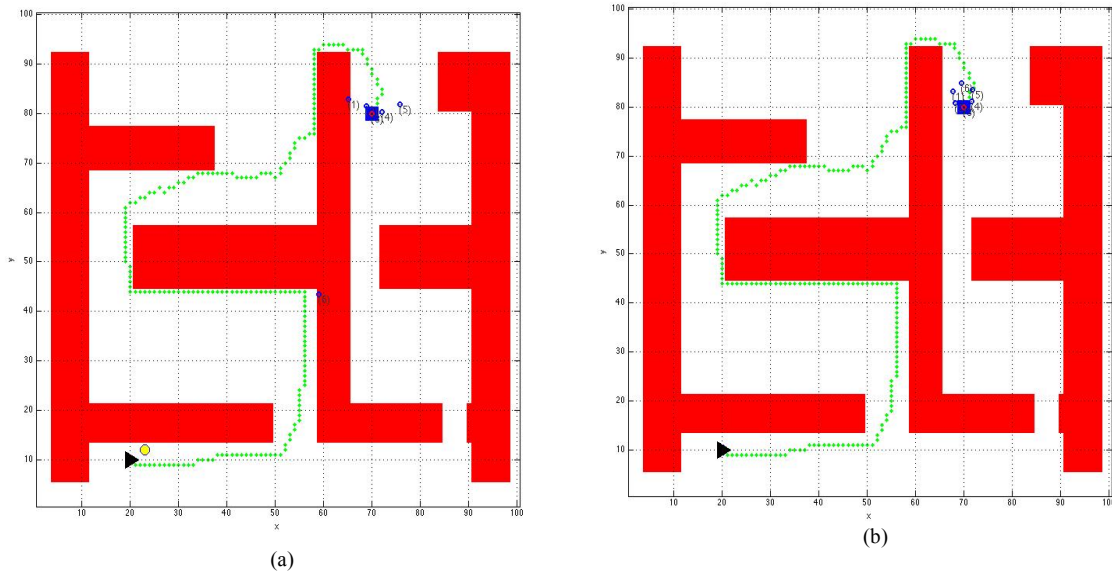


Figure 4.6 Mission executions in map1 with (a) Constant low k_ρ . (b) Variable k_ρ

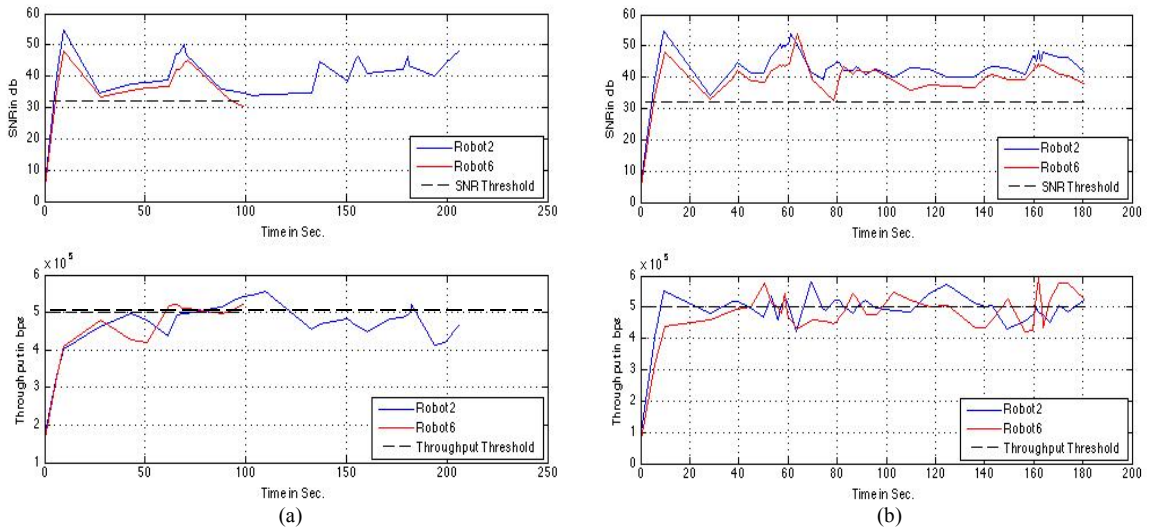


Figure 4.7 SNR and Throughput levels with (a) Constant low k_ρ . (b) Variable k_ρ

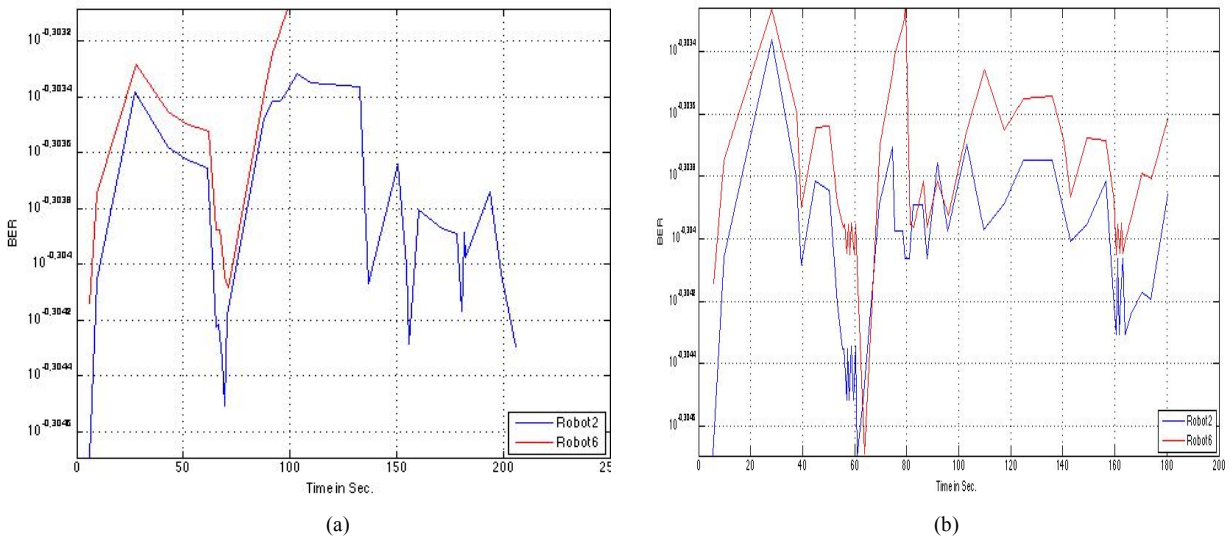


Figure 4.8 BER levels with (a) Constant low k_ρ . (b) Variable k_ρ

For map1, both MRSR and MRDV protocols have the same performance, and both gave the same results.

4.2.2 Map2

- **Using the MRSR Protocol:**

When we ran the same simulation for the second map, but of course with different path trajectory (calculated by the leader, then, saved to a .mat file to be used several times with this map). Follower robot 1, 5 and 6, were moving with changing speed at constant k_p until second 60 when the delay period imposed. They were trying to avoid colliding with an obstacle and paused to avoid other followers while finishing the turn, they got delayed, and did not catch up with the leader then stopped. Therefore, they are considered as *Lost_Followers* in the next message sent from the leader to the remaining team. Each and every follower executes the reformation code based on the participated members count. In this case, the new shape is wedge, as shown in Fig.4.9 (a).

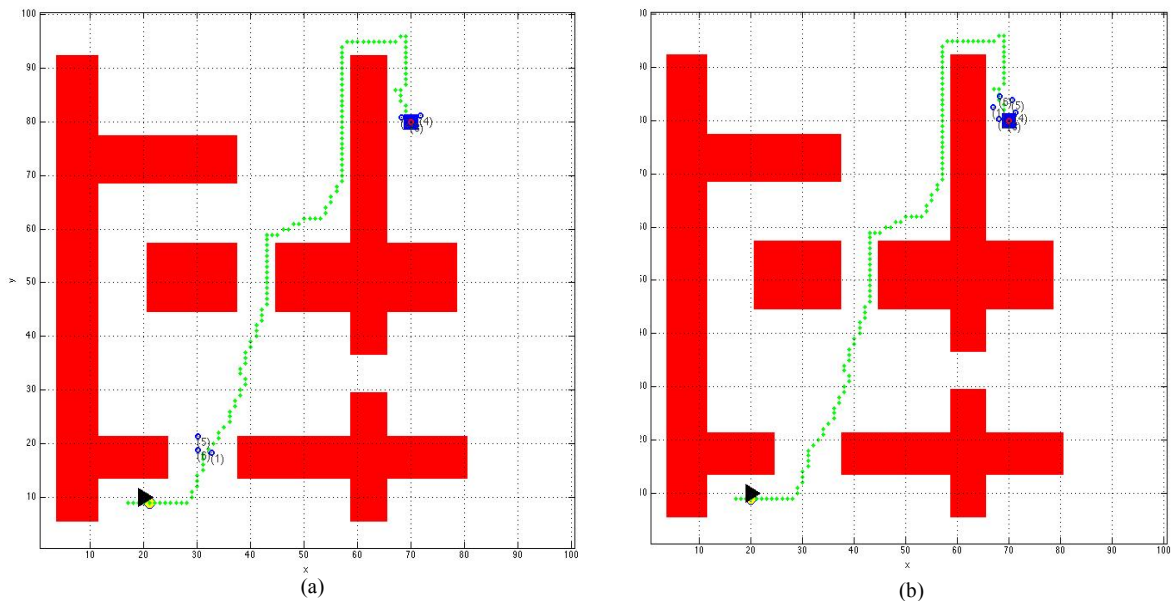


Figure 4.9 Mission executions in map1 with (a) Constant low k_p . (b) Variable k_p

From Fig. 4.10(a), follower robot 2 end point link throughput was under the minimum threshold as soon as the leader started moving, and because of the constant velocity control factor k_p , after the pause delay, the link throughput kept decreasing but the link did not break though, since the received power was not below the SNR threshold. Although, follower robot 2 succeeded in reaching the goal, the overall throughput was less than acceptable.

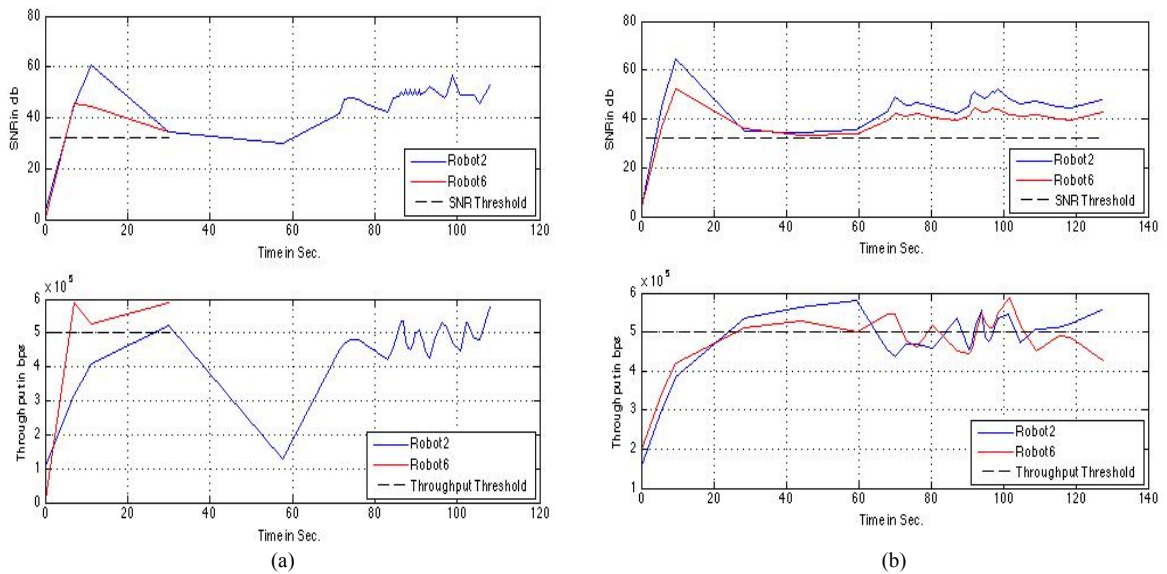
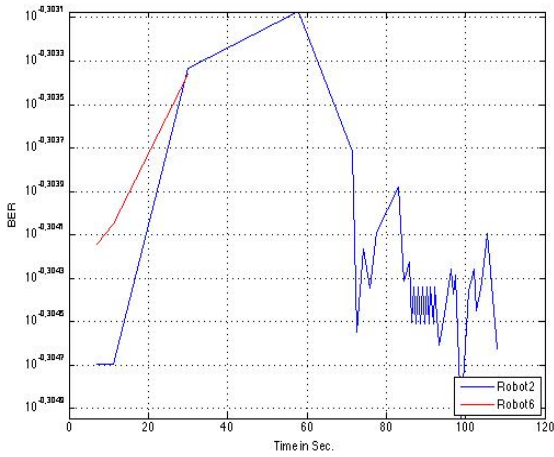
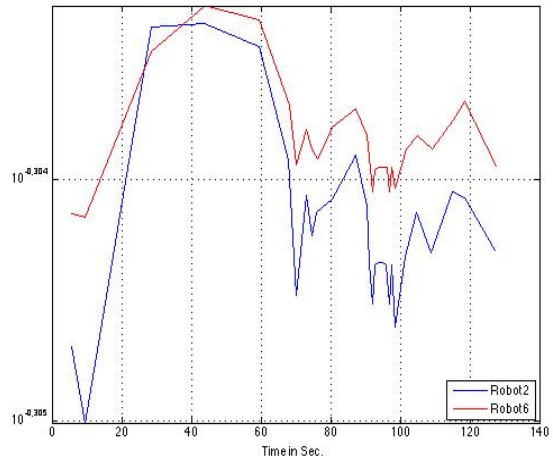


Figure 4.10 SNR and Throughput levels with (a) Constant low k_p . (b) Variable k_p

In Fig. 4.10(b), both SNR and throughput levels sustained almost a stable level when the controller changed the output velocity based on the link throughput for each follower robot. Even BER results, as shown in Fig.4.11(b), were acceptable for robot 6 (which was lost in the previous simulation), and did not have high levels after introducing the delay period to the simulation.



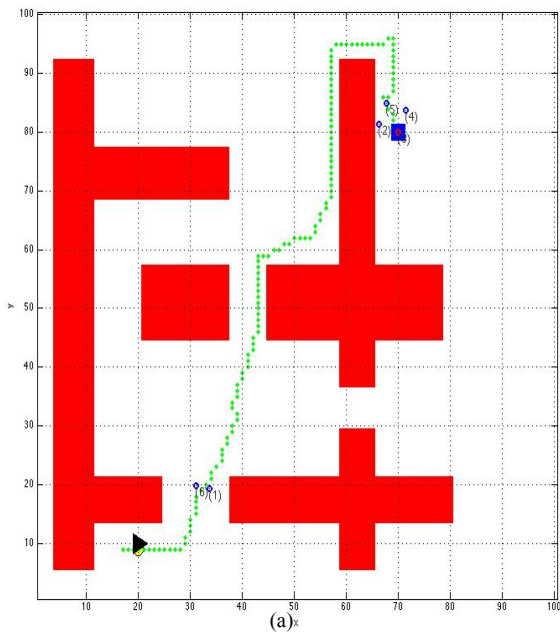
(a)



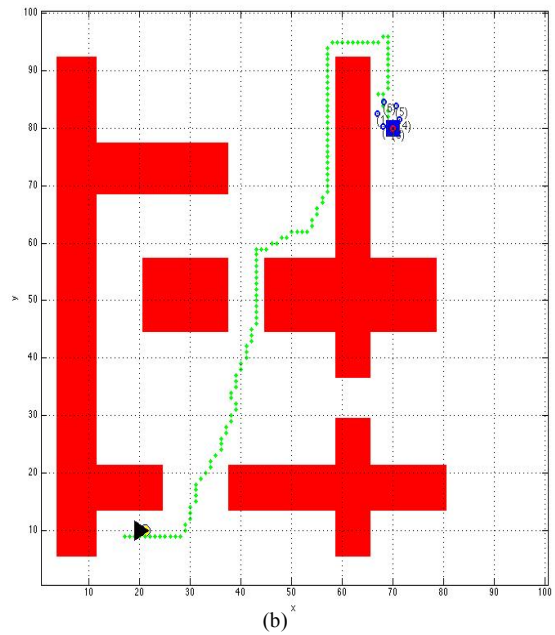
(b)

Figure 4.10 BER levels with (a) Constant low k_ρ . (b) Variable k_ρ

- Using the MRDV Protocol:



(a)



(b)

Figure 4.11 Mission executions in map1 with (a) Constant low k_ρ . (b) Variable k_ρ

We conducted the same simulation steps as when using the MRSR routing protocol, and the results showed a noticeable difference when the leader lost two followers instead of three as the case of using the MRSR protocol, as shown in Fig. 4.12(a). Even for followers close to leader like follower robot 2, the throughput levels were not above the threshold, and BER level was high most of the mission time as shown in Fig.4.13 (a) and

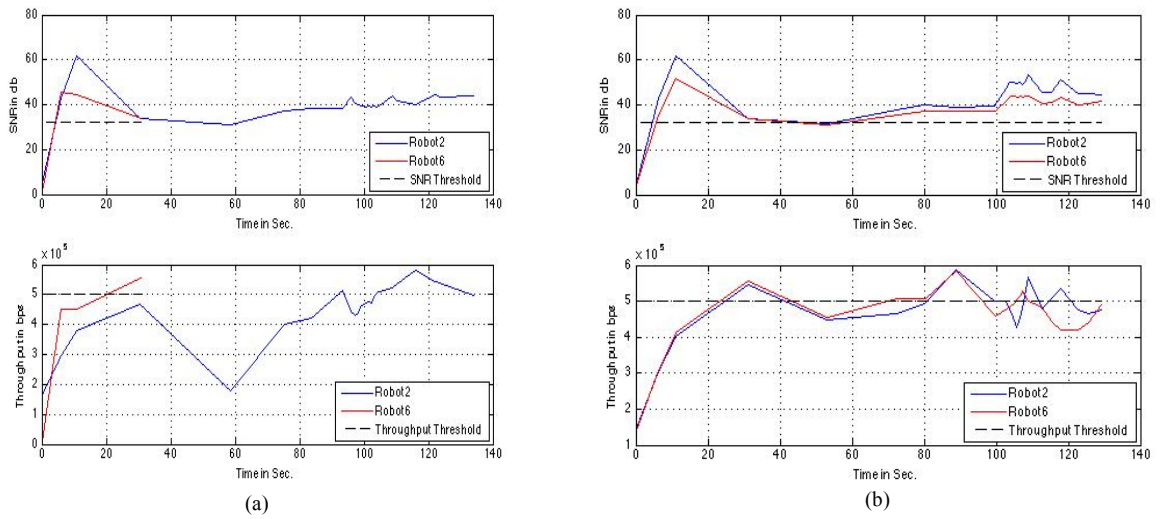


Figure 4.12 SNR and Throughput levels with (a) Constant low k_ρ . (b) Variable k_ρ

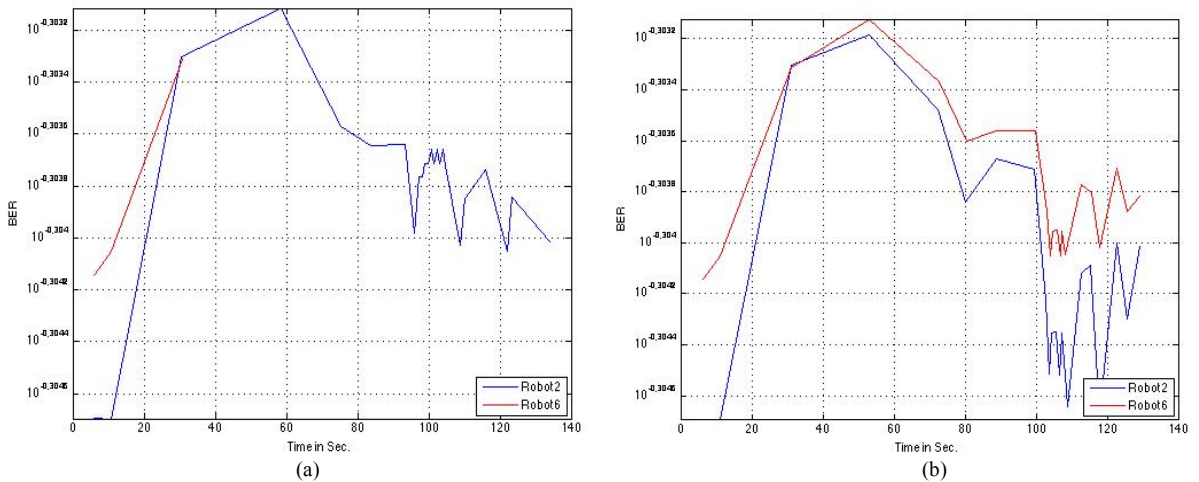


Figure 4.14 BER levels with (a) Constant low k_ρ . (b) Variable k_ρ

Fig.4.14 (a).

When using map2 as the mission environment, performance of both protocols was not the same. Using the MRDV protocol has better performance with constant and variable controller factor.

4.2.3 Map3

- **Using the MRSR Protocol:**

In this environment, MRSR protocol performed better than MRDV protocol. First, using the MRSR protocol and constant k_ρ , the formation lost only three robots, as in Fig.4.15 (a), where when using the MRDV and same settings, four follower robots were

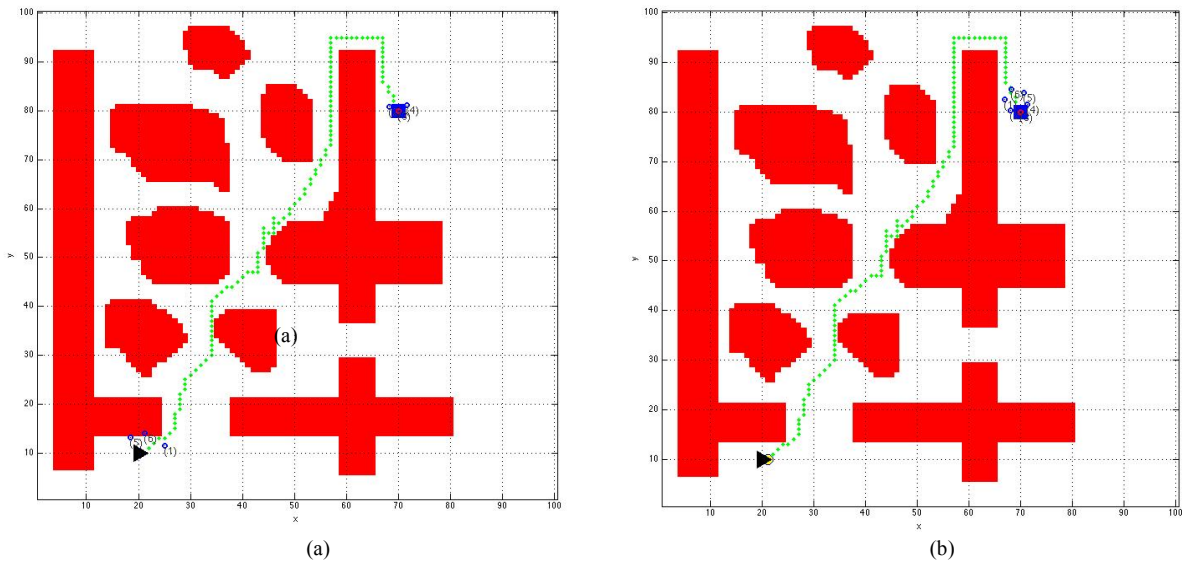


Figure 4.15 Mission executions in map1 with (a) Constant low k_ρ , (b) Variable k_ρ

lost, as shown in Fig.4.18 (a). Using a variable k_ρ has a big impact on the followers' performance, as shown in Fig. 4.16 (b), follower 6, with the lowest throughput at constant low k_ρ , has the most offset in throughput when using variable k_ρ factor, and even when

the throughput was under the acceptable level, the controller kept the throughput stabilized for longer periods.

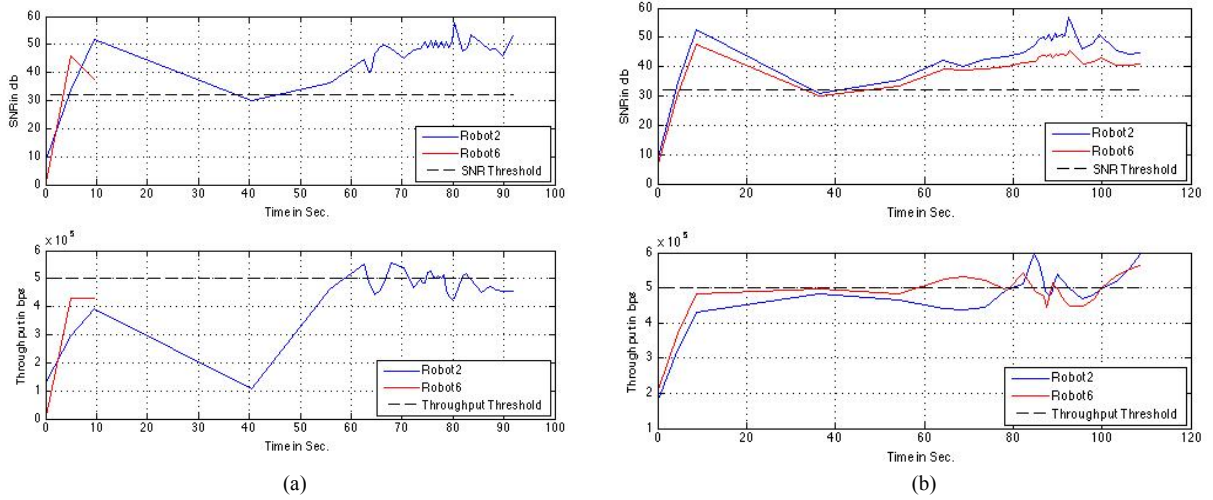


Figure 4.16 SNR and Throughput levels with (a) Constant low k_ρ . (b) Variable k_ρ

Furthermore, the BER level for robot 6 was improved. As in Fig. 4.17 (b), BER kept decreasing after the imposed delay at second 60, hence, the controller factor as a function of the throughput proved better performance using the MRSR protocol.

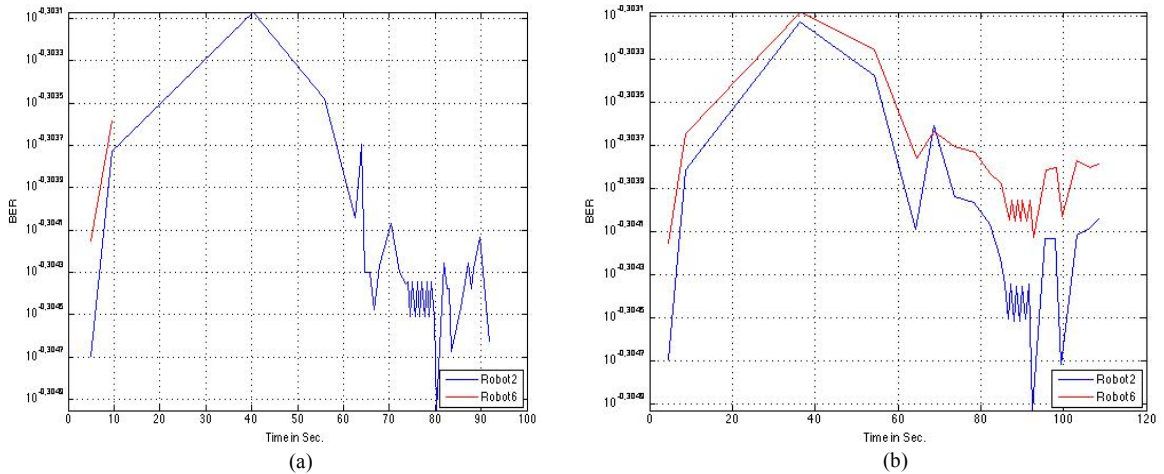


Figure 4.13 BER levels with (a) Constant low k_ρ . (b) Variable k_ρ

- Using the MRDV Protocol:

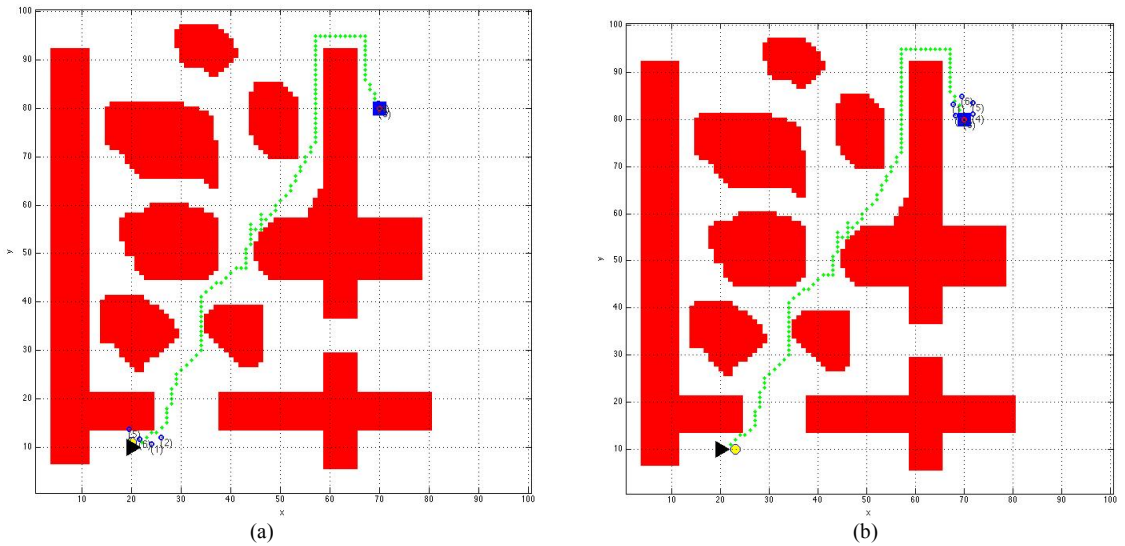


Figure 4.14 Mission executions in map1 with (a) Constant low k_p . (b) Variable k_p

So far, this is the worst-case scenario in the simulation. Four follower robots were lost, 1, 2, 5 and 6, and because of that, the resulting formation shape is a column shape, as shown in Fig. 4.18 (a). Since the follower robots 2 and 6 were both lost, Fig. 4.19(a) and Fig. 4.20(a) show the measured levels for a short time.

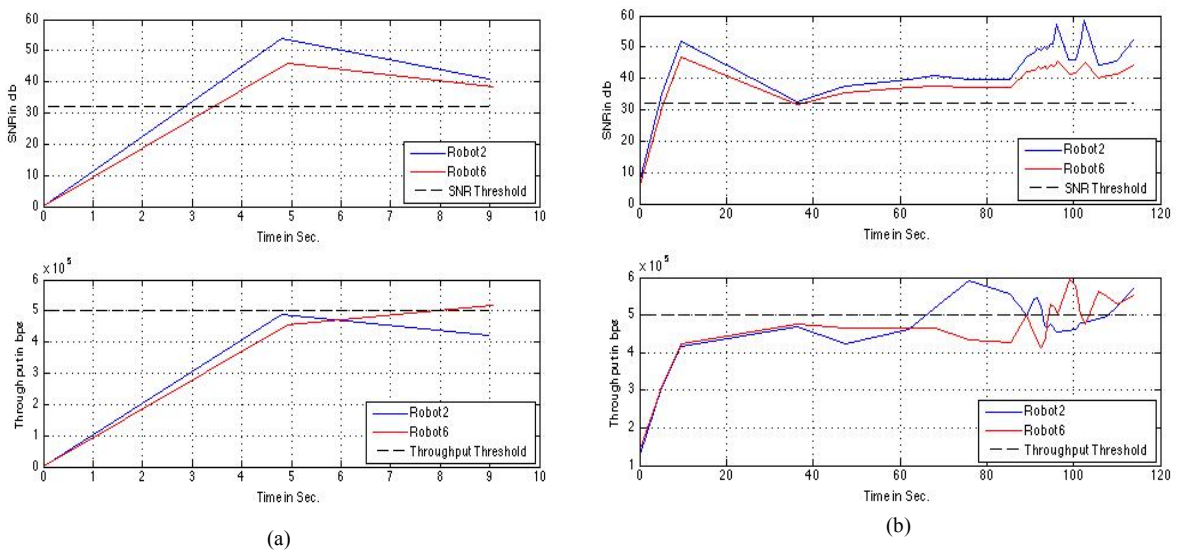


Figure 4.19 SNR and Throughput levels with (a) Constant low k_p . (b) Variable k_p

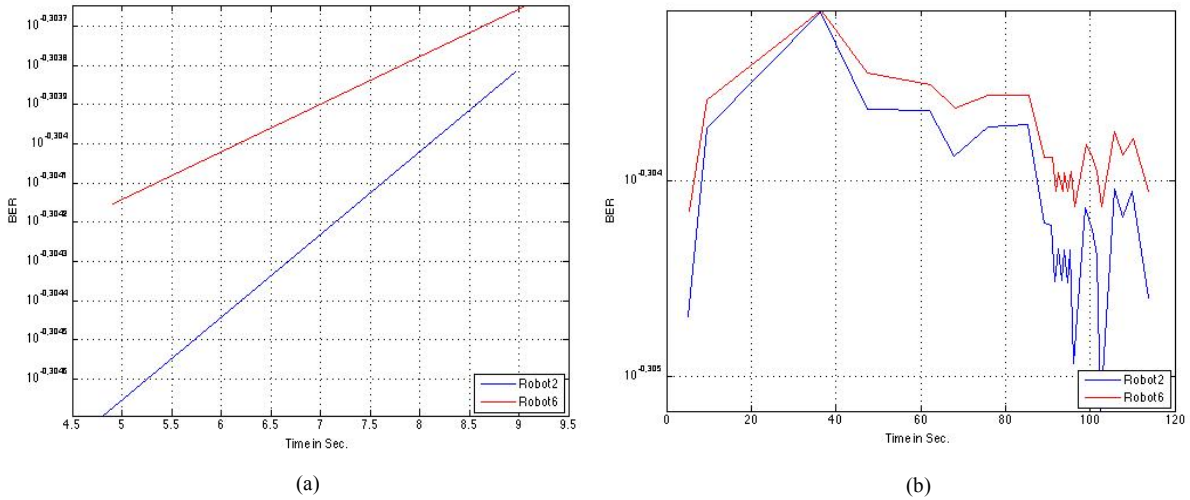


Figure 4.20 BER levels with (a) Constant low k_ρ . (b) Variable k_ρ

In general, MRSR protocol performance has better results than MRDV. When the local controller change the velocity of the followers according to links throughput, MRDV has a slightly higher overhead than MRSR, leading to lower throughput in the long run.

4.3 Failing Robot Scenarios

As mentioned in Chapter 3, to keep followers a one-hop distance from the leader, in case of lost follower or leader, to achieve a maximum link throughput, each robot local controller has the ability to change the followers shape formation in case of lost follower(s), or a missing leader. Also, it has the ability to elect another leader and execute the Path_Trajectory Module that calculates and plan a new path for the new leader to follow.

The leader can detect a lost follower(s) after not receiving ACK(s) for the *Next* message, if that is the case, the leader sends a *Lost_Follower* message, which include the leader pose and remaining followers' IDs. Fig. 4.21 is an example on changing the team formation shape from hexagon to diamond when two followers were lost.

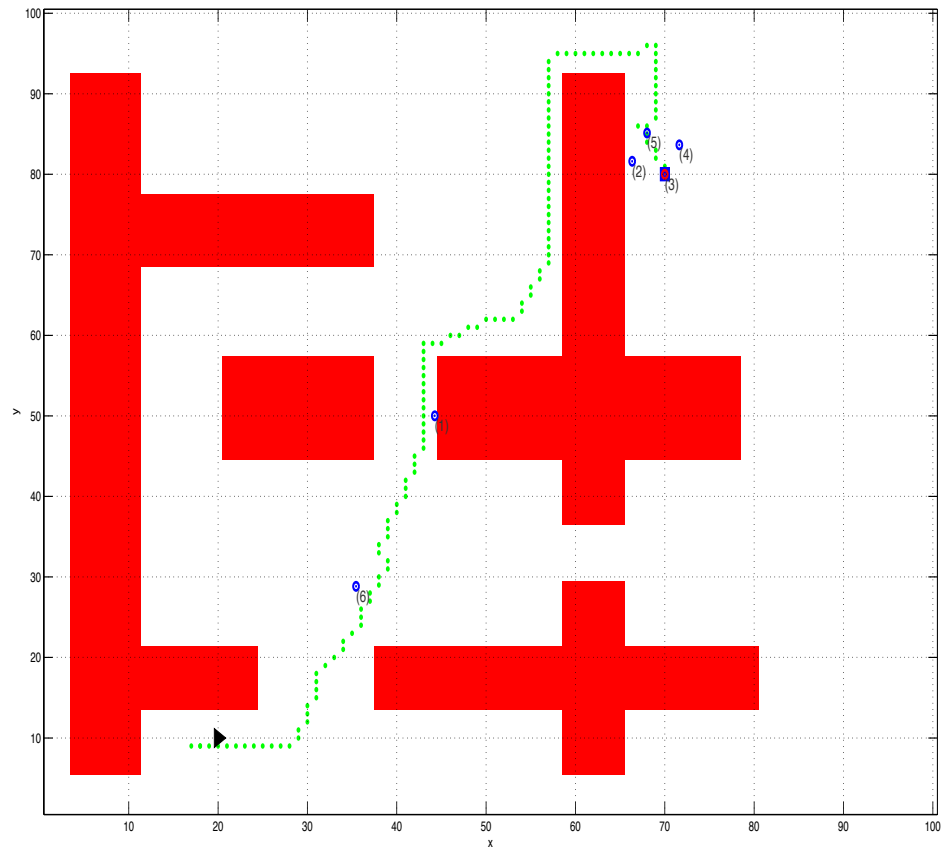


Figure 4.21 Lost Followers 6 & 1, Diamond Shape

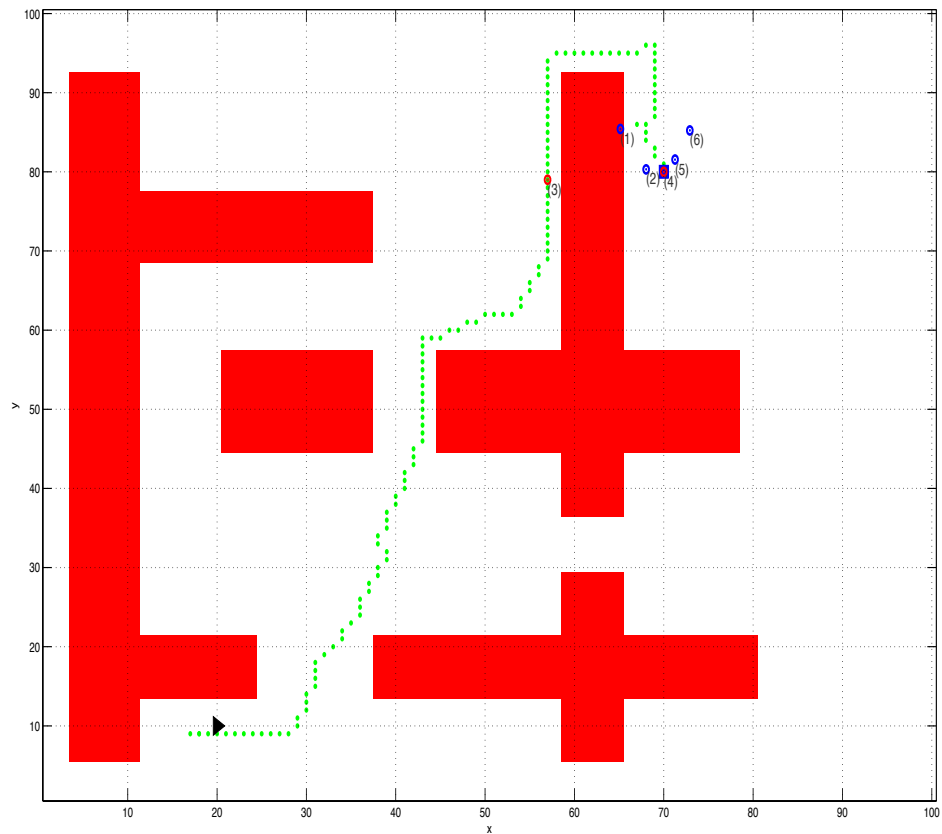


Figure 4.22 Lost leader case, follower 4 become leader

When the followers reach the next targeted point, and after $T = 2s$, if they did not receive a *Next* message from the leader, they will enter the *Lost_Leader* Mode, and according to the team IDs the have from the last received message, every follower elects the same new leader, and form the same new shape. For example, in Fig.4.22, after the leader, robot 3, is stopped and did not communicate with the followers, and according to Algorithm 1 in Chapter 3, follower 4 becomes the new leader of five-robot team, with wedge formation shape. If the leader was considered lost because of miscommunication, but still have a battery life, it will continue moving and finish the task by itself.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

This thesis studies the formation control problem for a team of mobile robots operating in an unknown but structured and static environment, with limited sensing capabilities and minimum inter robot communication.

One of the main contributions of this thesis is controlling and maintaining the team formation using wireless network communication constraints. Through simulation results, improvement in team performance has been exhibited, where each and every follower kept the specified distance to their leader with changing velocity without forcing the leader to slow down so delayed followers would catch up, or increasing number of exchanging messages between them.

Also, an enhancement to the PRM algorithm for path planning in unknown environment has been developed, which produced better results in case of local minima. It successfully guided the planner to generate short and feasible routes in a short time, and led the robot out of the local minima effectively.

However there are still some drawbacks. If the robot enters a small closed room (less than 10m in width), the probabilistic planning approach can not help the robot to turn back. Instead, the robot will keep going in circles until it finds the room entrance and leave the room. Moreover, by continuously choosing random sample points, PRM can

bring the robot dangerously close to an obstacle - this can be addressed by adding an obstacle bias in the sensor measuring calculation. These random samples also cause the slightly unnatural jumps between points during path execution. Therefore, more limits can be added to the PRM algorithm to ensure a smoother trajectory.

The results have demonstrated that the separation distance alone may not guarantee the robot enough speed to reach a targeted (goal) position while maintaining acceptable link quality level. The link data throughput was introduced as another factor for formation tracking control, where each robot continuously monitors the end-to-end throughput. From simulation results, it has been shown that by incorporating link throughput constraints into the problem formulation, we gain the ability to satisfy the main physical task while maintaining the necessary level of network connectivity. The reliability of this approach was tested using two different routing protocols and in 3 different environments.

In general, the MRSR protocol performance has better results compared MRDV, when the local controller change the velocity of the followers according to the link throughput, since MRDV has a slightly higher overhead than MRSR.

REFERENCES

- [1] Dongkyoung Chwa; Seo, J.H.; Pyojae Kim; Jin-Young Choi, "Sliding mode tracking control of nonholonomic wheeled mobile robots," *American Control Conference, 2002. Proceedings of the 2002* , vol.5, no., pp.3991,3996 vol.5, 2002
doi: 10.1109/ACC.2002.1024553
- [2] Aicardi, M.; Casalino, G.; Bicchi, A.; Balestrino, A., "Closed loop steering of unicycle like vehicles via Lyapunov techniques," *Robotics & Automation Magazine, IEEE* , vol.2, no.1, pp.27,35, Mar 1995
doi: 10.1109/100.388294
- [3] Wu, W.; Chen, H.; and Wang, Y.; 1999, "Back stepping Design for Path Tracking of Mobile Robots," *Proceedings of the IEEE International Conference Robots and Systems*, pp. 1822-1827.
- [4] Fredslund, J.; Mataric, M.J., "A general algorithm for robot formations using local sensing and minimal communication," *Robotics and Automation, IEEE Transactions on*, vol.18, no.5, pp.837,846, Oct 2002
doi: 10.1109/TRA.2002.803458
- [5] Fierro, R.; Song, P.; Das, A.; Kumar, V., "Cooperative control of robot formations", in: Murphey, R.; Pardalos P. (Eds.), "Cooperative Control and Optimization," Kluwer Academic Press, Hingham, MA, 2002.
- [6] Ogren, P.; Fiorelli, E.; Leonard, N.E., "Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment *Automatic Control, IEEE Transactions on* , vol.49, no.8, pp.1292,1302, Aug. 2004
doi: 10.1109/TAC.2004.832203
- [7] Das, A.K.; Fierro, R.; Kumar, V.; Ostrowski, J.P.; Spletzer, J.; Taylor, C.J., "A vision-based formation control framework," *Robotics and Automation, IEEE Transactions on* , vol.18, no.5, pp.813,825, Oct 2002
doi: 10.1109/TRA.2002.803463
- [8] Barnes, L.; Valavanis, K., "Swarm formation control utilizing elliptical surfaces and limiting functions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* , vol.39, no.6, pp.1434,1445, Dec. 2009
doi: 10.1109/TSMCB.2009.2018139
- [9] Sugihara, K.; Suzuki, I., "Distributed algorithms for formation of geometric patterns with

- many mobile robots,” *J. Robot. Syst.*, vol. 13, no. 3, pp. 127–139, 1996.
- [10] Ando, H.; Suzuki, I.; Yamashita, M., “Formation and agreement problems for synchronous mobile robots with limited visibility,” in *Proc. IEEE Int. Symp. Intelligent Control*, Monterey, CA, Aug. 1995, pp. 453–460
- [11] Mataric, M. J., “Designing and understanding adaptive group behavior,” in *Adaptive Behavior*. Cambridge, MA: MIT, Dec. 1995, vol. 4, pp. 1:50–1:81
- [12] Werger, B. B., “Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams,” *Artif. Intell.*, vol. 110, pp.293–320, 1999.
- [13] Werger, B. B.; Mataric, M. J., “Robotic ‘food’ chains: Externalization of state and program for minimal-agent foraging,” in *From Animals to Animats 4: Proc. 4th Int. Conf. Simulation of Adaptive Behavior*, Maes, Mataric, Meyer, Pollack, and Wilson, Eds., Cape Cod, MA, Sept. 1996, pp. 625–634.
- [14] Ren, W.; Sorensen, N., “Distributed coordination architecture for multi-robot formation control”, *Robotics and Autonomous Systems*, Volume 56, Issue 4, 30 April 2008, Pages 324-333, ISSN 0921-8890
- [15] Parker, L. E.; Kannan, B.; Tang, F.; Bailey, M., “Tightly-coupled navigation assistance in heterogeneous multi-robot teams,” *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* , vol.1, no., pp.1016,1022 vol.1, 28 Sept.-2 Oct. 2004
doi: 10.1109/IROS.2004.1389486
- [16] Lewis, M.A.; Tan, K-H, “High precision formation control of mobile robots using virtual structures,” *Autonomous Robots*, vol. 4, no. 4, pp. 387-403, 1997
- [17] Balch, T.; Arkin, R. C., “Behavior-based formation control for multi-robot teams,” *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926-939, 1998
- [18] Orqueda, Omar Armando Adrian, Ph.D. dissertation, “Vision-based control of multi-agent systems”, Oklahoma State University, 2007, 202 pages; 3245822
- [19] Bicho, E.; Monteiro, S., “Formation control for multiple mobile robots: a non-linear attractor dynamics approach,” in *2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, October 27-31 2003, pp. 2016–2022
- [20] Brockett, R. W., “Asymptotic Stability and Feedback Stabilization,” *Differential Geometric Control Theory* , Vol. 27, Birkhauser, Boston, MA, USA, pp. 181-208, 1983.
- [21] Dixon, W. E.; Dawson, D. M.; Zergeroglu, E.; Behal, A., “Nonlinear Control of

- Wheeled Mobile Robots, Springer, New York, USA, 2001.
- [22] Bellingham, J.S.; Tillerson, M.; Alighanbari, M.; How, J.P., "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol.3, no., pp.2816,2822 vol.3, 10-13 Dec. 2002
doi: 10.1109/CDC.2002.1184270
- [23] Bellingham, J.; Richards, A.; How, J.P., "Receding horizon control of autonomous aerial vehicles," *American Control Conference, 2002. Proceedings of the 2002*, vol.5, no., pp.3741,3746 vol.5, 2002
doi: 10.1109/ACC.2002.1024509.
- [24] Lawton, J.; Young, B.; Beard, R., "A decentralized approach to formation maneuvers" *Automatic Control, IEEE Transactions on*, 19(6):933_941, December 2003.
- [25] Siegwart, R.; Nourbakhsh, I. R. "Introduction to Autonomous Mobile Robots". Bradford Co., Scituate, MA, USA. 2004.pp80-88
- [26] Lee, Z.; Chen, X., "Path planning approach based on probabilistic roadmap for sensor based car-like robot in unknown environments," *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol.3, no., pp.2907,2912 vol.3, 10-13 Oct. 2004
doi: 10.1109/ICSMC.2004.1400774
- [27] Olfati-Saber, R., "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, pp. 401– 420, 2006.
- [28] Fordigh, Magnus; Johansson, Per; Larsson, Peter, "Wireless Ad Hoc Networking - the Art of Networking without a Network." *Ericsson Review* 4 (2000): 248-63.
- [29] Jaydev P. Desai; Ostrowski, J.; Kumar, V., "Controlling formations of multiple mobile robots", *IEEE International Conference on Robotics*.
- [30] Dunbar, W. B.; Murray, R. M., "Model predictive control of coordinated multi-vehicle formations," *Proceedings of the 41st IEEE CDC, Las Vegas, Nevada, USA, 2002*.
- [31] Mataric, M. J., "Behavior-based control: Examples from navigation, learning, and group behavior," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, pp. 323–336, 1997
- [32] Lumelsky, V.; Stepanov, A., "Path planning strategies for a point mobile automation moving amongst unknown obstacles of arbitrary shape," *Algorithmica*, 1987
- [33] A. Goldsmith, A. Nin, "Wireless Communications", Cambridge: Cambridge

University Press, 2005. Print.

- [34] Corke, P.I., "Robotics, Vision & Control", Springer 2011, ISBN 978-3-642-20143-1.
- [35] Wireless Network Simulator in MATLAB, Available: <http://wireless-matlab.sourceforge.net>
- [36] Sarthong, J.; Sittichivapak, S., "Near Theoretical Maximum Throughput Limits of CSMA/CA RTS CTS Protocol in IEEE 802.11 Wireless Networks Using Active Node Back-off Algorithm", International Conference on Circuits, System and Simulation IPCSIT vol.7 (2011) © (2011) IACSIT Press, Singapore
- [37] IEEE Std. 802.11a Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer Extension in the 5 GHz Band, IEEE802.11a WG September 1999.
- [38] IEEE Std. 802.11b Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-Speed Physical Layer Extension in the 2.4 GHz Band, IEEE802.11a WG, September1999.
- [39] IEEE Std. 802.11g Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: Amendment 4 Further Higher Data Rate Extension in the 2.4 GHz Band, IEEE802.11a WG, June
- [40] Balador, A.; Movaghar, A.; Jabbehdari, S.; Kanellopoulos, D., "A Novel Contention Window Control Scheme for IEEE 802.11 WLANs", ETE Tech Rev 2012; 29:202-12
- [41] Kavraki, L.E.; Svestka, P.; Latombe, J.-C.; Overmars, M.H., "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," Robotics and Automation, IEEE Transactions on , vol.12, no.4, pp.566,580, Aug 1996, doi: 10.1109/70.508439.
- [42] Li, P.; Fang, Y.; Li, J. , "Throughput, Delay, and Mobility in Wireless Ad Hoc Networks," *NFOCOM, 2010 Proceedings IEEE , vol., no., pp.1,9, 14-19 March 2010*
doi: 10.1109/INFCOM.2010.5461902

- [43] Feeney, L. M.; Nilsson, M., "Investigating the energy consumption of a wireless network - interface in an Ad Hoc networking environment," *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol.3, no., pp.1548,1557 vol.3, 2001
doi: 10.1109/INFCOM.2001.916651
- [44] Kozat, U.; Tassiulas, L., "Throughput capacity of random Ad Hoc networks with infrastructure support", In Proceedings of ACM MobiCom, San Diego, CA, USA, June 2003.
- [45] Li, P.; Fang, Y., "Impacts of topology and traffic pattern on capacity of hybrid wireless networks" *Mobile Computing, IEEE Transactions on*, 8(12):1585–1595, December 2009.
- [46] Grossglauser, M.; Tse, D., "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 477-486, Aug. 2002.
- [47] Monteiro, S.; Bicho, E., "Robot formations: Robots allocation and leader-follower pairs," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, vol., no., pp.3769,3775, 19-23 May 2008, doi: 10.1109/ROBOT.2008.4543789
- [48] Johnson, D. B.; Maltz, D. A., "Dynamic Source Routing in Ad Hoc Wireless Networks". Kluwer Academic, 1996.
- [49] Perkins, C. E.; Royer, E. M., "Ad Hoc on-demand distance vector routing," in *Proc. of IEEE WMCSA*, February 1999.
- [50] Hsieh, M.A.; Cowley, A.; Kumar, V.; Taylor, C.J., "Towards the deployment of a mobile robot network with end-to-end performance guarantees," *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, vol., no., pp.2085-2090, 15-19 May 2006,
doi: 10.1109/ROBOT.2006.1642012
- [51] Das, S.M.; Hu, Y.C.; Lee, C.S.G.; Yung-Hsiang Lu, "Mobility-aware Ad Hoc routing protocols for networking mobile robot teams," *Communications and Networks, Journal of*, vol.9, no.3, pp.296,311, Sept. 2007, doi:

10.1109/JCN.2007.6182857.

- [52] Kavraki, L.E.; Svestka, P.; Latombe, J.-C.; Overmars, M.H., "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol.12, no.4, pp.566,580, Aug 1996, doi: 10.1109/70.508439
- [53] Lumelsky V.;Stepanov; A., "Path planning strategies for a point mobile automation moving amongst unknown obstacles of arbitrary shape," *Algorithmica*, 1987
- [54] Vadakkepat, P.; Tan K.C.; Ming-Liang, W., "Evolutionary artificial potential fields and their application in real time robot path planning" *Proc. of the 2000 Congress on Evolutionary Computation (July 2000)*, pp. 256–263
- [55] Yuan Yan; Mostofi, Y., "Robotic Router Formation in Realistic Communication Environments," *Robotics, IEEE Transactions on*, vol.28, no.4, pp.810,827, Aug. 2012, doi: 10.1109/TRO.2012.2188163
- [56] Xi Chen; Wei Zhuang; Jindong Tan, "The maintaining of communication links quality in unknown environment," *Information and Automation, 2009. ICIA '09. International Conference on*, vol., no., pp.218,223, 22-24 June 2009, doi: 10.1109/ICINFA.2009.5204924
- [57] Desai, J. P.; Ostrowski, Jim P.; Kumar, V., "Modeling and control of formations of nonholonomic mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 905–908, 2001
- [58] Anderson, B. D O; Changbin Yu; Fidan, B.; Hendrickx, J.M., "Rigid graph control architectures for autonomous formations," *Control Systems, IEEE*, vol.28, no.6, pp.48,63, Dec. 2008, doi: 10.1109/MCS.2008.929280
- [59] Proakis, J.; Salehi,M., "Digital Communications", McGraw-Hill Education, Nov 6, 2007