

**SAFE OPEN-LOOP STRATEGIES FOR HANDLING  
INTERMITTENT COMMUNICATIONS IN  
MULTI-ROBOT SYSTEMS**

A Thesis  
Presented to  
The Academic Faculty

By

Siddharth Mayya

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Electrical and Computer Engineering

School of Electrical and Computer Engineering  
Georgia Institute of Technology  
May 2016

Copyright © 2016 by Siddharth Mayya

**SAFE OPEN-LOOP STRATEGIES FOR HANDLING  
INTERMITTENT COMMUNICATIONS IN  
MULTI-ROBOT SYSTEMS**

Approved by:

Dr. Magnus Egerstedt, Advisor  
*Professor, School of ECE*  
*Georgia Institute of Technology*

Dr. Yorai Wardi  
*Professor, School of ECE*  
*Georgia Institute of Technology*

Dr. Fumin Zhang  
*Associate Professor, School of ECE*  
*Georgia Institute of Technology*

Date Approved: 29<sup>th</sup> April 2016

## **ACKNOWLEDGMENTS**

I would like to thank my advisor Dr. Magnus Egerstedt for introducing me to the fascinating world of multi-agent robotics and guiding my research for the last two years. I also thank all members of the GRITS Lab for their continued help throughout this research effort. Finally, I thank my family for their support and encouragement over the last two years.

# TABLE OF CONTENTS

<b>ACKNOWLEDGMENTS</b> . . . . .	iii
<b>LIST OF FIGURES</b> . . . . .	v
<b>SUMMARY</b> . . . . .	vii
<b>CHAPTER 1 INTRODUCTION</b> . . . . .	1
1.1 Multi-Robot Systems . . . . .	1
1.2 Problem Description . . . . .	3
1.3 Solution Strategy . . . . .	4
<b>CHAPTER 2 BACKGROUND AND PREVIOUS WORK</b> . . . . .	6
2.1 Reachable Sets of Unicycle Robots . . . . .	7
2.2 Minimum Volume Ellipsoids . . . . .	12
<b>CHAPTER 3 REACHABILITY ANALYSIS</b> . . . . .	14
3.1 Problem Formulation . . . . .	14
3.2 The Reachable Set . . . . .	15
3.3 Bounding Shapes for $\mathcal{R}(t)$ . . . . .	16
<b>CHAPTER 4 SAFE TIME HORIZON COMPUTATION</b> . . . . .	31
4.1 Problem Formulation . . . . .	31
4.2 Defining the Safe Time . . . . .	33
4.3 Motion Strategy . . . . .	34
4.4 Safety Guarantees . . . . .	35
4.5 Computing the Safe Time . . . . .	37
4.6 Simulations . . . . .	40
<b>CHAPTER 5 EXPERIMENTAL SETUP AND RESULTS</b> . . . . .	44
5.1 Experimental Setup . . . . .	44
5.2 Methodology . . . . .	46
5.3 Results . . . . .	47
<b>CHAPTER 6 CONCLUSIONS</b> . . . . .	51
<b>APPENDIX A STATE TRANSFORMATIONS: UNICYCLE ROBOTS</b> . . . . .	52
<b>APPENDIX B CIRCULAR TRAJECTORY COMPUTATION</b> . . . . .	53
<b>REFERENCES</b> . . . . .	54

## LIST OF FIGURES

Figure 1	Unicycle Robot Dynamics . . . . .	8
Figure 2	Reachable Sets for a Dubins's car [1] . . . . .	10
Figure 3	Reachable Set $\mathcal{R}(t)$ . . . . .	17
Figure 4	Examples of Bounding Shapes . . . . .	18
Figure 5	Convex Hull of $\mathcal{R}(t)$ . . . . .	19
Figure 6	Approximating the involute of a circle with the arc of a circle in the first quadrant . . . . .	20
Figure 7	Simplification of $conv(\mathcal{R}(t))$ . . . . .	21
Figure 8	Minimum Area Ellipse enclosing $\mathcal{K}(t)$ . . . . .	28
Figure 9	Evolution of $\mathcal{R}(t)$ , $conv(\mathcal{R}(t))$ , $\mathcal{K}(t)$ , $\xi(\mathcal{K}(t))$ for $t = 2, 5, 12, 25$ . . . . .	30
Figure 10	Updates times $t_k$ at which commands are transmitted to the robots. . . . .	31
Figure 11	Trajectory of robot $i$ after communication failure: $v_i(t_k) = 0.25$ , $\omega_i(t_k) = -0.9$ . . . . .	33
Figure 12	Rationale behind safe time computation. The red robot can move in an open-loop fashion until its path intersects the reachable set of another robot. . . . .	35
Figure 13	Computational Burden vs. Network Size . . . . .	40
Figure 14	Multi-robot Patrol with Safe Times . . . . .	41
Figure 15	Single robot communication failure. The red curve indicates the path taken by the robot and the blue ellipse is the reachable set of a neighboring robot. . . . .	42
Figure 16	Double robot communication failure. Each robot executes its last received velocity for the respective safe time horizon. . . . .	43
Figure 17	The design of a Khepera III Robot . . . . .	45
Figure 18	The Khepera III Robot . . . . .	45
Figure 19	Four Khepera Robots patrolling a corridor . . . . .	48
Figure 20	The safe time algorithm reduces the disruption in motion caused by communication failures. . . . .	49

Figure 21 The robot on the right experiences a short duration communication failure. The safe time algorithm prevents jerky motion and allows the robot to continue moving. . . . . 49

Figure 22 Mutiple simultaneous communication failures are handled effectively by the safe time algorithm. . . . . 50

## SUMMARY

The objective of this thesis is to develop a strategy that allows robots to safely execute open-loop motion patterns for pre-computed time durations when facing interruptions in communication. By computing the time horizon in which collisions with other robots are impossible, this method allows the robots to move safely despite having no updated information about the environment. As the complexity of multi-robot systems increase, communication failures in the form of packet losses, saturated network channels and hardware failures are inevitable. This thesis is motivated by the need to increase the robustness of operation in the face of such failures. The advantage of this strategy is that it prevents the jerky and unpredictable motion behaviors which often plague robotic systems experiencing communication issues. To compute the safe time horizon, the first step involves constructing reachable sets around the robots to determine the set of all positions that can be reached by the robot in a given amount of time. In order to avoid complications arising from the non-convexity of these reachable sets, analytical expressions for minimum area ellipses enclosing the reachable sets are obtained. By using a fast gradient descent based technique, intersections are computed between a robot's trajectory and the reachable sets of other robots. This information is then used to compute the safe time horizon for each robot in real time. To this end, provable safety guarantees are formulated to ensure collision avoidance. This strategy has been verified in simulation as well as on a team of two-wheeled differential drive robots on a multi-robot testbed.

# CHAPTER 1

## INTRODUCTION

Over the last few years, the coordinated control of multi-robot systems has garnered a lot of research interest. This is partly due to the lowering cost and increasing computing power of embedded processors and partly due to the vast array of unanswered research questions in the field. In a team of robots performing a coordinated task, communication is often an integral part of the closed loop control mechanism [2]. In fact, in many scenarios such as extra-terrestrial exploration, high precision manufacturing and on multi-robot testbeds, the robots rely on communicating with a centralized decision maker for their motion commands. As the team sizes of the robots grow and the deployment conditions become increasingly harsh, communication failures are inevitable.

This thesis is motivated by the need to robustly handle intermittent communications that occur in multi-robot systems. To this end, we develop a safe open-loop motion strategy that allows a robot to move even in scenarios where communication frequently breaks down. Furthermore, in this thesis, we consider multi-robot systems which rely on a centralized decision making device for their motion commands. This architecture is similar to what is commonly used in multi-robot testbeds. We therefore, use multi-robot testbeds as an example to introduce some of the terminologies and concepts that will be used in this thesis. This chapter gives a general introduction to multi-robot systems and discusses multi-robot testbeds in greater detail (Section 1.1). This is followed by a brief description of the problem (Section 1.2), and an outline of the solution strategy (Section 1.3).

### 1.1 Multi-Robot Systems

A multi-robot system is defined as a collection of robots performing some global task using local interaction rules. Teams of robots may be required to cover an area effectively [3], build formations of a specific shape and size [4], take sensor measurements [5] etc.

Deploying a team of robots offers many advantages like improved reliability, redundancy and effectiveness in completing tasks. For example, in a disaster rescue mission, sending out a team of robots to scour for survivors is more advantageous compared to sending out a single robot. All these applications share a fundamental attribute- the presence of a signal exchange network typically in the form of a wired or wireless communication channel. The amount of information being shared depends on how coordinated the task is [2]. It is therefore crucial to design strategies to improve the robustness of multi-robot algorithms against faulty communication channels, which is one of the motivations for the work presented in this thesis. Before we outline the specific problems that will be solved in this thesis, a discussion on multi-robot testbeds is necessary to establish context and introduce some terminologies.

### **1.1.1 Multi-Robot Testbeds**

Experimental verification is an integral and essential part in the research cycle of any multi-robot algorithm. A majority of the verification is done on simulations due to constraints on cost, complexity and time. However, implementation of control algorithms on actual hardware is necessary to prove real-world viability. Multi-robot testbeds have become an integral part of multi-agent robotics research in the recent years due to their ability to facilitate efficient and accurate verification of algorithms (for e.g. [6], [7]). Several successful multi-robot testbeds have been developed for serving different classes of robots. Some notable ones are the micro-UAV testbed at University of Pennsylvania [8], a hovercraft vehicle testbed at Caltech [9], and the recent remote-access testbed at Georgia Tech [10]. A multi-robot testbed must ensure that a wide range of control algorithms can be faithfully executed on a large number of robots in a safe and robust manner. This typically entails the use of a tracking system for motion control. The tracking system is used to retrieve the global positions of the robots using an overhead detection device. This position data is then used by a centralized host computer to close the position control feedback loop. The host then transmits the desired velocity commands to the robots via a wireless communication

channel. The robots then execute these velocity commands.

In addition to robust tracking, a multi-robot testbed must also ensure that the physical assets are protected by guaranteeing collision avoidance. Since users are allowed to take control of the physical hardware on the testbed, it is important to add measures to detect and prevent potential collisions. On the other hand, while ensuring collision avoidance, it is important to not influence the motion of the robots to an extent that the control objectives are not met [11]. A number of algorithms and techniques have been developed to address this issue for teams of robots (e.g. [12], [13]). Many of these techniques can be directly applied on multi-robot testbeds. As a result of these advancements, multi-robot testbeds have become a state-of-the-art research instrument allowing flexible execution of control algorithms while at the same time ensuring faithful and safe robot operations.

## 1.2 Problem Description

Wireless communication is an essential part of most multi-robot control algorithms and may include inter-robot data exchange, commands sent from a central host to the robots and diagnostic measurements sent from the robots to the central host. With large number of robots, there is an increase in the traffic flowing through the communication network and occasional failures are expected. These may be caused due to transmitted packets getting lost [14] or corrupted [15], saturation of the communication channels [16] or hardware failures on the robots or the host. This raises the following question:

*What should a robot do in case it stops receiving velocity commands from the host?*

A existing strategy used to handle such an eventuality is to stop the robot immediately. While this behavior preserves safety, it leads to undesirable motion behaviors when communication failures are frequent and intermittent. It might cause the robots to move in a jerky fashion as intermittent velocity commands are received. Furthermore, this strategy does not reflect robustness in the sense that it does not allow for a graceful degradation

of performance during such an unexpected failure. To address all of these problems, we propose a motion strategy that allows robots to robustly and safely handle communication failure.

### 1.3 Solution Strategy

In this thesis we answer the following question:

*What is the maximum time for which a robot can move in a provably safe manner after experiencing communication failure in a multi-robot system?*

We develop a technique to compute such a time horizon and use it to build an open-loop motion strategy for the robots. This time horizon is called the safe time horizon.

As a first step in computing the safe time, reachable sets are constructed around the robots. We consider unicycle robots as they are the most popular type of ground robots [17]. However the concept of safe times can be extended to robots of any type. A reachable set consists of all possible locations that a robot can reach at a future time. Since reachable sets for unicycle robots are non-convex, efficient ellipsoidal approximations are computed. To this end, analytical expressions for the minimum area ellipses enclosing the convex hull of reachable sets of unicycle robots are derived. In the second step, the safe time of each robot is computed by performing intersection tests between the robot's trajectory and the ellipsoidal reachable sets of other robots. After the host has computed the safe time for all the robots, it transmits these times to the robots along with the velocity commands. If a robot loses communication, it can continue to move in an open-loop fashion for the duration of the last received safe time. In this time window, collision avoidance is guaranteed because as long as the robot stays outside the reachable sets of other robots, it cannot collide with them. It is proved that, with this strategy, the safety guarantees provided by the multi-robot control algorithm are preserved and are extended to situations where the robots are moving without direct control from the host. After confirming the robustness of the strategy in simulation, experimental verification is performed on a team of two-wheeled

differential drive robots which are executing a multi-robot patrol algorithm on a multi-robot testbed. The experimental setup, methods and results are given in detail.

## CHAPTER 2

### BACKGROUND AND PREVIOUS WORK

Developing an open-loop motion strategy as described in Chapter 1 requires novel control strategies and tools for describing and analyzing the nature of the problem. In order to allow a robot to move without having direct control on its motion, a rigorous analysis of the future positions of all robots in the plane must be performed. Reachability analysis is a popular way to compute all the possible future states that a dynamical system can be in, given its current state and input constraints. The structure of reachable sets has been extensively studied for unicycle dynamics models. For these systems, the problem of computing the reachable set has been shown to be closely related to the problem of finding the minimum-time paths that the robot can take between two points in the state space [18]. Two particularly common dynamics models used to explore the structure of these paths are the *Dubin's car* model and the *Reeds-Shepp car* model.

As discussed in Chapter 1, the non-convex nature of the reachable set entails that it must be approximated by a simpler shape to allow for cheap set representations, rotations and intersection tests. The concept of enclosing complex shapes with simpler ones is not new and a large amount of literature exists on the topic. Such shapes are often called Bounding Shapes in the Computer Science literature [19] where they are commonly used. Here, we shall focus on the literature specific to computing ellipsoidal approximations. The reasons for choosing ellipses as the choice of bounding shape are discussed in Chapter 3.

In this chapter, we will present an introduction to the structure of time-optimal paths and reachable sets for the *Dubin's car* model and the *Reeds-Shepp car* model. This is followed by a discussion on some ellipsoidal calculus techniques. Note, however, that this chapter is not meant to be a thorough review of these topics. Only concepts that are directly applied in this thesis are elaborated.

## 2.1 Reachable Sets of Unicycle Robots

An important concept in control theory is that of a reachable set. The reachable set of a system is defined as the set of all the states which can be attained by this system given its dynamics and constraints on the input. It can be formally defined as follows.

**Definition 1.** *Given a dynamical system whose state evolves according to the following differential equation,*

$$\dot{x} = f(x, u) \quad (1)$$

$$u \in \mathcal{U} \quad (2)$$

where  $\mathcal{U}$  is the set of admissible controls, the reachable set at time  $t$  can be defined as

$$\mathcal{R}(t; x_0) = \bigcup_{u(\cdot) \in \mathcal{U}} \mathcal{X}(t, x_0, u(\cdot))$$

where  $\mathcal{X}(t, x_0, u(\cdot))$  represents the solution to the differential equation given in Equation (1).

In particular we are interested in investigating reachability for unicycle robots. Unicycle robots are characterized by their position in the 2-D plane and their orientation with respect to the horizontal axes, as shown in Figure 1. The equations of motion for such a robot can be written as

$$\begin{aligned} \dot{x} &= v \cos(\phi) \\ \dot{y} &= v \sin(\phi) \end{aligned} \quad (3)$$

$$\dot{\phi} = \omega$$

$$|v| \leq v_m, \quad |\omega| \leq \omega_m$$

Denote  $z = (x, y) \in \mathbb{R}^2$ . This is also called the kinematic car model.

To travel to a point on the boundary of its reachable set, a unicycle robot must travel “as fast as it can” to reach the point. If this was not true, a point even farther way would be

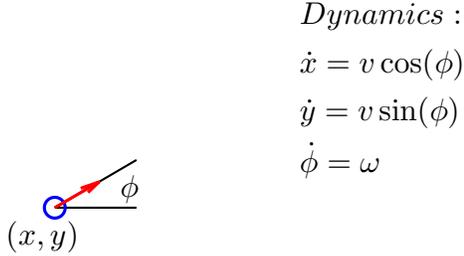


Figure 1: Unicycle Robot Dynamics

reachable in the same amount of time. Hence, the problem of computing the reachable set is closely linked to solving a minimum-time optimal control problem. This can be illustrated as follows.

**Definition 2.** Define  $V(z_f; z_0, \phi_0)$  as the solution of a minimum time optimal control problem formulated as

$$V(z_f; z_0, \phi_0) = \min_{u \in \mathcal{U}} J(t_f) = \int_0^{t_f} 1 dt = t_f$$

given the initial position and orientation  $(z_0, \phi_0)$  and the desired final position  $z_f$  of the unicycle robot. The final orientation of the robot  $\theta_f$  is a free variable.  $t_f$  is the final time which is the minimum time in which this position can be reached. Note that,  $t_f$  is also a free variable in this formulation.

Then, we can define the reachable set of the unicycle robot as,

$$\mathcal{R}(t; z_0, \theta_0) = \{z' \in \mathbb{R}^2 : V(z'; z_0, \theta_0) \leq t\} \quad (4)$$

Before analyzing the geometry of the reachability sets for the dynamics given in Equation 3, we investigate the structure of these sets for simpler motion models.

### 2.1.1 Dubin's Car Model

**Definition 3.** *A Dubin's car [20] moves in the forward direction with a constant velocity and has an upper bound on the curvature of the path it can take. Its dynamics can be described by the following equations:*

$$\begin{aligned}\dot{x} &= \cos(\phi) \\ \dot{y} &= \sin(\phi) \\ \dot{\phi} &= \omega \quad |\omega| \leq 1\end{aligned}\tag{5}$$

*The term  $\omega$  acts as a control input, specifying the instantaneous angular velocity of the vehicle.*

In the paper [18], published in the American Journal of Mathematics, Lester Eli Dubins considered the problem of finding the minimum length continuously differentiable path connecting two given points provided that the curvature was bounded and the outgoing and incoming direction at the first and second point was specified. A very common interpretation to this problem concerns finding the shortest path between two points taken by the Dubin's car for which the starting and ending directions are specified. The main theorem from this publication, allows an interpretation of this problem in the context of a minimum-time optimal control problem. It is worth noting that here, since the velocity of the car is constant and is 1, the shortest path corresponds to the time-optimal path. It was shown that the time-optimal paths for the Dubin's car are concatenations of three different path segments: a straight line segment and two circular arcs of maximum curvature.

A landmark paper by Cockayne and Hall [21] constructed the reachable sets for the Dubin's car model. It was established in this paper, that any point on the boundary of the reachable set can be attained using piecewise control with at most one switch. In particular, it was established that only four different control actions  $(-1, 0)$ ,  $(1, 0)$ ,  $(-1, 1)$ ,  $(1, -1)$  are required to reach the boundary. By varying the switch time between the control actions and the control sequences themselves, the structure of the reachable sets for a Dubin's car

can be obtained. Figure 2 shows the construction of the reachable set of a Dubin's car for different time horizons. For each time horizon, the reachable set has its own scale. In the

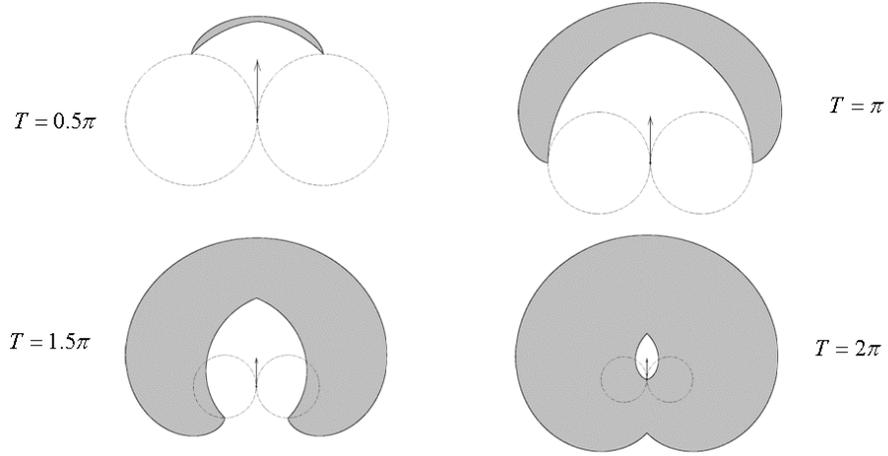


Figure 2: Reachable Sets for a Dubin's car [1]

next section, we investigate another very important motion model called the Reeds-Shepp car model.

### 2.1.2 Reeds-Shepp Car Model

In 1990, James Reeds and Lawrence Shepp considered a dynamics model which extended the Dubin's car model to include forward and backward motions [22]. Formally, this model can be defined as follows.

**Definition 4.** *A Reeds-Shepp car can move both forward and backward with a fixed velocity. It can change its direction of motion instantaneously. This motion can be described using the following differential equations.*

$$\begin{aligned}
 \dot{x} &= v \cos(\phi) \\
 \dot{y} &= v \sin(\phi) \\
 \dot{\phi} &= \omega \quad |v| = 1, |\omega| \leq 1
 \end{aligned} \tag{6}$$

In [22], Reeds and Shepp showed that the shortest path of the car between two points (with incoming and outgoing directions specified) could always be achieved by means of

specific trajectories, namely concatenations of at most five types of paths (in this case too, the time-optimal paths are the same as the shortest paths). These paths are either circular arcs or straight lines. By parameterizing each trajectory with the switching times between the paths, they further classified all these concatenations into 48 families of paths. This family of trajectories together, suffice to join any two states by the shortest path. These results were derived without making the use of optimal control theory and were mainly based on geometric arguments. We now formalize the definition of a sufficient family of paths.

**Definition 5.** *A collection of trajectories  $\mathcal{F}$  is sufficient for time-optimality if, given any two points  $a, b$  in the state space, there exists a path in  $\mathcal{F}$  that goes from  $a$  to  $b$  and is time-optimal.*

Sussmann and Tang in [23] used modern geometric tools combined with optimal control to reduce the sufficient family of trajectories for the Reeds-Shepp car from 48 to 46. To prove the existence of time-optimal trajectories between arbitrary points, they had to first convexify the control set of the Reeds-Shepp model. This is because no theorems regarding the existence of optimal controls are directly applicable to the Reeds-Shepp car model owing to its non-convex velocity set ( $|\nu| = 1$ ). They convexify the control set of the model by replacing it by the convex hull ( $|\nu| \leq 1$ ). This model is called the Convexified Reeds-Shepp (CRS) model. Since the input control set for this model is compact and convex, one can now apply standard optimal control results to the problem. Using Pontryagin's maximum principle, techniques from Lie algebra and envelope theory, the authors single out a family of trajectories  $\mathcal{F}$  which is sufficient for time-optimality for the CRS Model. The authors note that any trajectory that is time-optimal for the CRS model and is a valid Reeds-Shepp path, is also time-optimal for the Reeds-Shepp model. The sufficient family of trajectories  $\mathcal{F}$  happen to be admissible to the Reeds-Shepp model and hence provide a solution to the Reeds-Shepp time-optimality problem.

Further building on the results from [23], Soueres et al. in [24] construct the set of

reachable positions for a Reeds-Shepp car model. To do this, they solve the “Free Final Direction Problem” which concerns the computation of the time-optimal path to a point in the plane without being concerned about the final orientation of the robot. They show that the number of sufficient paths required to solve this problem is 3. By integrating over the final positions of these trajectories, they obtain geometric representations of the reachable sets.

## 2.2 Minimum Volume Ellipsoids

Given any convex body  $K$  in  $\mathbb{R}^n$ , there exists a unique ellipsoid of minimum volume circumscribing it [25]. Before presenting the details of this result, we define an ellipsoid in a formal manner.

**Definition 6.** *An ellipsoid  $E$  in an  $n$ -dimensional Euclidean Space  $\mathbb{R}^n$  can be obtained by applying a regular affine transformation on the unit ball  $\mathbb{B}_n$ . Define  $c \in \mathbb{R}^n$  as the center of the ellipsoid and a non-singular symmetric matrix  $A \in \mathbb{R}^{n \times n}$ . Then, the ellipsoid can be obtained as follows,*

$$E = c + A\mathbb{B}_n = \{c + Ax : x \in \mathbb{R}^n, \|x\| \leq 1\} \quad (7)$$

Such a matrix  $A$  has a unique polar decomposition given by  $A = WZ$  where  $W$  is a positive definite matrix and  $Z$  is an orthogonal matrix. So,  $E = c + WZ\mathbb{B}_n$ . But the unit ball is invariant under an orthogonal transformation so  $E = c + W\mathbb{B}_n$ . We now make the following change of variables:

$$x = c + Wu$$

$$u = W^{-1}(x - c)$$

So,  $E = \{x : x \in \mathbb{R}^n, \|W^{-1}(x - c)\|^2 \leq 1\}$ , and defining  $H := W^{-2}$ , we get,  $E = \{x \in \mathbb{R}^n : (x - c)^T H(x - c) \leq 1\}$ . We denote this ellipse as  $E(c, H)$ .

Given any convex body  $K$  in  $\mathbb{R}^n$ , the existence of a minimum volume ellipsoid circumscribing  $K$  can be proved using compactness arguments. The uniqueness of these enclosing

ellipsoids was proved in [26] by Danzer, Laugwitz and Lenz and then independently in [27] by Zaugskin. The circumscribing ellipsoid is often called the *Löwner ellipse* named after Charles Löwner who extensively applied uniqueness properties to obtain results. We denote the unique minimum volume ellipsoid corresponding to  $K$  as  $\xi(K)$ . The circumscribing ellipsoid  $\xi(K)$ , has important applications in optimization and statistics. The ellipsoid algorithm of Khachiyan [28] for linear programming, increased the interest in the problem of obtaining the ellipsoid for any set of points. Following this, there have been a number of algorithms that have emerged to numerically compute the minimum volume ellipsoid enclosing a set of points. Some of the papers discussing these algorithms are [29], [30] and [31].

In [25], Fritz John directly tackled the problem of obtaining the minimum volume ellipsoid circumscribing a convex body  $K$  by introducing a semi-infinite programming problem and then obtaining the optimality conditions.

If the convex set satisfies certain symmetry properties, we can simplify the computation of the circumscribing ellipsoids and in some cases even obtain analytical expressions for the minimum volume ellipsoids, as shown in [32].

## CHAPTER 3

### REACHABILITY ANALYSIS

In this chapter, a geometric representation of the reachable sets of unicycle robots is outlined. These expressions follow from a direct extension to the results presented in [23] and in [24]. Motivated by the non-convexity of the reachable set, we search for simpler geometric shapes which can enclose the reachable set. A list of desirable properties for such shapes is listed in Section 3.3. It is argued that ellipses are an excellent choice of shape for enclosing the reachable set. Analytical expressions for the minimum area ellipse enclosing the reachable set are computed in Section 3.3.3. Since all the robots are identical, we perform the analysis for a single robot.

#### 3.1 Problem Formulation

For this chapter and the following chapters, we assume, without loss of generality, that the unicycle robots have the following dynamics:

$$\begin{aligned}\dot{x} &= v \cos(\phi) \\ \dot{y} &= v \sin(\phi) \\ \dot{\phi} &= \omega \\ |v| &\leq 1, \quad |\omega| \leq 1\end{aligned}\tag{8}$$

The maximum linear and angular velocity is normalized to 1. Details regarding the state space and time scale transformation required to transform the dynamics in Equation 3 to Equation 8 is provided in Appendix A. Let  $z = (x, y) \in \mathbb{R}^2$ . Denote  $(z_0, \phi_0)$  as the initial state of the robot at time  $t = 0$ . The reachable set  $\mathcal{R}(t; z_0, \phi_0)$  at time  $t$ , is the set of all geometric positions in the  $x - y$  plane that can be reached at time  $t$  by a robot starting at  $(z_0, \phi_0)$ .

For  $z_0 = (0, 0), \phi_0 = 0$ , denote the reachable set as  $\mathcal{R}(t)$ . Since the dynamics shown in

Equation 8 is drift-free and owing to the nature of the appearance of  $\phi$  in the equations, the structure of the reachable set does not depend on  $z_0$  and  $\phi_0$ .

$$\mathcal{R}(t, z_0, \phi_0) = z_0 + \Pi_{\phi_0} \mathcal{R}(t) \quad (9)$$

Here,  $\Pi_{\phi_0}$  represents a rotation by  $\phi_0$ . This expression indicates that the reachable set for a robot at any position and orientation in the state space, can be obtained by applying suitable rotations and translations to the reachable set  $\mathcal{R}(t)$ . Therefore, the study of reachable sets can be restricted to the case when the robot is positioned at the origin and oriented along the positive x-axis ( $z_0 = (0, 0), \phi_0 = 0$ ).

### 3.2 The Reachable Set

In this section, we mathematically describe the structure of the reachable set  $\mathcal{R}(t)$  which is the same as the reachable set of a Reeds-Shepp car model (see Equation 6).

To illustrate this, we present a result which has been proved in [23].

**Definition 7.** *For a robot with dynamics given in Equation 8, given any two points  $a, b \in \mathbb{R}^2$  along with the initial and final directions, there exists a time-optimal trajectory from  $a$  to  $b$  which consists of the following path segments:*

1. *Circular Arcs corresponding to the inputs  $|v| = 1$  and  $|\omega| = 1$ .*
2. *Straight Lines corresponding to inputs  $|v| = 1$  and  $\omega = 0$ .*

*The same family of trajectories is also sufficient for time-optimality for a Reeds-Shepp car.*

In [24], the authors use this family of trajectories to obtain the reachable set for a Reeds-Shepp car. They do so, by computing the shortest path among all the paths linking an initial position with a fixed direction of the car, to a final position with a free direction. The same results apply to the unicycle robot we are considering, since the family of time-optimal paths is the same.

**Definition 8.** For a unicycle robot with dynamics given in Equation 8, the reachable set  $\mathcal{R}(t)$  is symmetric about the  $x$  and  $y$  axis. Define  $C_a^{++}, C_b^{++}, C_c^{++}$  as sets which constitute the boundary of the reachable set in the first quadrant.

$$\begin{aligned}
C_a^{++}(t) &= p \in \mathbb{R}_{++}^2 : \begin{cases} p_x &= \cos u - (2 + d) \sin u \\ p_y &= \sin u + (2 + d) \cos u - 1 \end{cases}, u \in S_a, d = t - \pi/2 - u \\
C_b^{++}(t) &= p \in \mathbb{R}_{++}^2 : \begin{cases} p_x &= -2 \sin w + \sin(w + v) \\ p_y &= 2 \cos w - \cos(w + v) - 1 \end{cases}, w \in S_b, v = t - w \\
C_c^{++}(t) &= p \in \mathbb{R}_{++}^2 : \begin{cases} p_x &= \sin q + s \cos q \\ p_y &= -\cos q + s \sin q + 1 \end{cases}, q \in S_c, s = t - q
\end{aligned} \tag{10}$$

where  $S_a, S_b, S_c$  are given by,

$$\begin{aligned}
S_a &= \{u \in [0, \pi/2) : d = t - \pi/2 - u \geq 0\} \\
S_b &= \{w \in [0, \pi/2) : w < v = t - w < \pi/2\}
\end{aligned} \tag{11}$$

$$S_c = \{q \in [0, \pi/2] : s = t - q \geq 0\} \tag{12}$$

Using these definitions, the boundary of the reachable set in the first quadrant can be defined as,

$$\partial\mathcal{R}(t)^{++} = C_a^{++}(t) \cup C_b^{++}(t) \cup C_c^{++}(t)$$

$\mathbb{R}_{++}^2$  denotes the closed first quadrant. By symmetry we can construct the boundary of the entire reachable set. (Figure 3)

### 3.3 Bounding Shapes for $\mathcal{R}(t)$

In order to compute safe time horizons, we need to perform intersection tests between a robot's position and the reachable sets of other robots. Since  $\mathcal{R}(t)$  is non-convex, checking

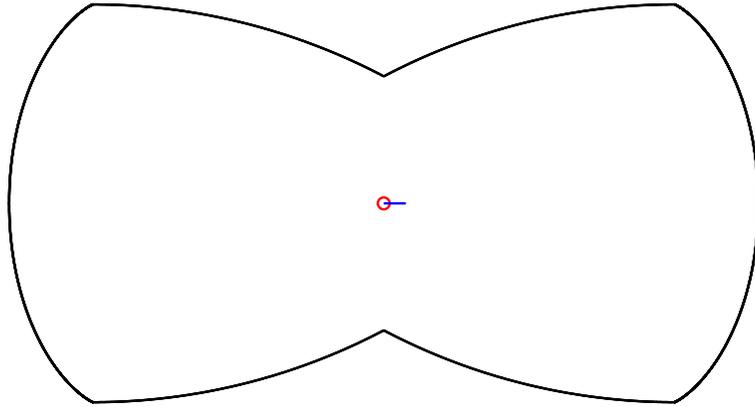


Figure 3: Reachable Set  $\mathcal{R}(t)$

for intersections in real-time can be computationally expensive. Furthermore,  $\mathcal{R}(t)$  does not allow for a finite representation, which can incur significant memory usage as the number of robots increase [33]. Hence, there is a motivation to find simpler geometric shapes which enclose the reachable set. Such techniques are commonly used in computer science to enable real time collision detection (for e.g. [34], [35], [36]) and for efficient intersection tests between objects [37].

### 3.3.1 Why Ellipses?

A bounding shape is a single simple shape encapsulating one or more objects of more complex nature. We aim to choose a bounding shape having the following desirable properties:

- Finite/Cheap representation
- Inexpensive to compute
- “Tight” fitting
- Easy to rotate and transform

- Inexpensive intersection tests

The use of bounding shapes can result in significant performance gain, and the elimination of complex intersection tests with non-convex objects. Some commonly used bounding shapes are boxes, polygons, circles and ellipses (Figure 4).

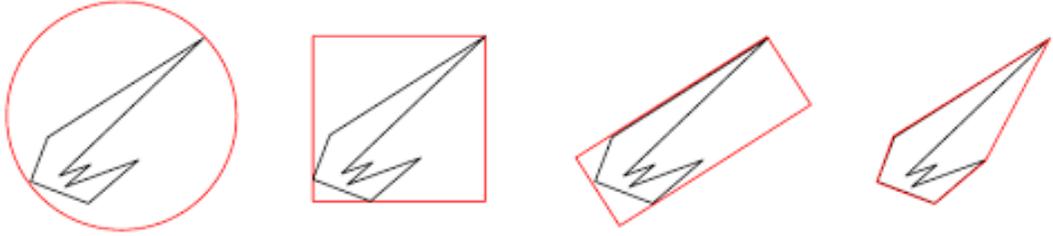


Figure 4: Examples of Bounding Shapes

As shown in Section 2.2, ellipses can be represented by their center and a transformation matrix. Computing minimum area (volume) ellipses (ellipsoids) enclosing convex bodies is a well studied topic and several algorithms have been developed to compute such shapes for an arbitrary convex body (e.g. [28], [29]).

We show that, owing to the symmetry properties of  $\mathcal{R}(t)$ , analytical expressions for the minimum area ellipse enclosing the convex hull of  $\mathcal{R}(t)$  can be computed. This is computationally very beneficial since no iterative numerical technique is required to compute the ellipse at run-time. Furthermore, in Section 3.3.3.4, we show that this ellipse converges asymptotically to the convex hull of  $\mathcal{R}(t)$  as time grows larger. In other words, the approximation becomes better as time increases. All these factors together, make ellipses the prime choice for the bounding shape enclosing reachable sets of unicycle robots.

### 3.3.2 Convex Hull of $\mathcal{R}(t)$

Since the computation of the ellipse requires that the underlying set be convex (see Section 2.2), we first obtain an expression for the convex hull of  $\mathcal{R}(t)$  (Figure 5). The convex hull of a set  $K$  is defined as the smallest convex set that contains  $K$ .

**Definition 9.** *The convex hull of  $\mathcal{R}(t)$  is denoted by  $\text{conv}(\mathcal{R}(t))$  and can be expressed as the*

convex combination of all the points in  $\mathcal{R}(t)$ .

$$\text{conv}(\mathcal{R}(t)) = \left\{ \sum_{i=0}^{\infty} \theta_i p_i : \forall p_i \in \mathcal{R}(t), \theta_i \geq 0, \sum_{i=0}^{\infty} \theta_i = 1, i = 1, 2, \dots \right\}$$

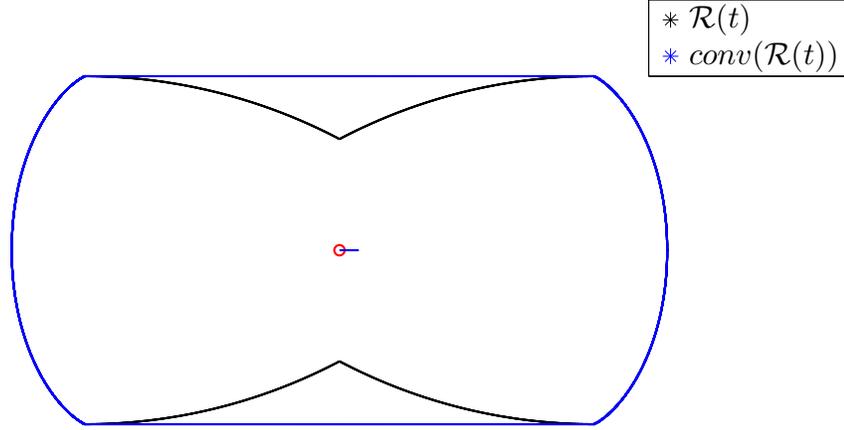


Figure 5: Convex Hull of  $\mathcal{R}(t)$

### 3.3.2.1 Simplifying $\text{conv}(\mathcal{R}(t))$

We now use direct geometric arguments to obtain a simplified representation of  $\text{conv}(\mathcal{R}(t))$ . The curved portion of the boundary of  $\text{conv}(\mathcal{R}(t))$  in the first quadrant is the part of an involute of a circle (see curve  $C_c^{++}$  in Equation 10) whose structure presents difficulties when deriving an analytical expression for the minimum area ellipse enclosing the set. To simplify computations, this curved boundary can be replaced by the arc of a circle. Figure 6 illustrates this, by juxtaposing the two curves in the first quadrant. By performing this replacement in all four quadrants,  $\text{conv}(\mathcal{R}(t))$  can be approximated by a simpler set  $\mathcal{K}(t)$  which is the part of a circle centered at the origin and bounded between two hyper-planes which are parallel to the  $x$  axis. In the following proposition, it is proved that  $\mathcal{K}(t)$  encloses  $\text{conv}(\mathcal{R}(t))$  (Figure 7). Furthermore it is shown that, as time grows larger, the dissimilarity between  $\mathcal{K}(t)$  and  $\text{conv}(\mathcal{R}(t))$  goes to zero asymptotically. To measure the dissimilarity between two sets, the Jaccard distance is used as a metric [38]. The Jaccard Distance

between two sets  $X, Y \in \mathbb{R}^2$  using the area measure is given as

$$d_J(X, Y) = 1 - \frac{A(X \cap Y)}{A(X \cup Y)} \quad (13)$$

where  $A(\cdot)$  denotes the area of the set.

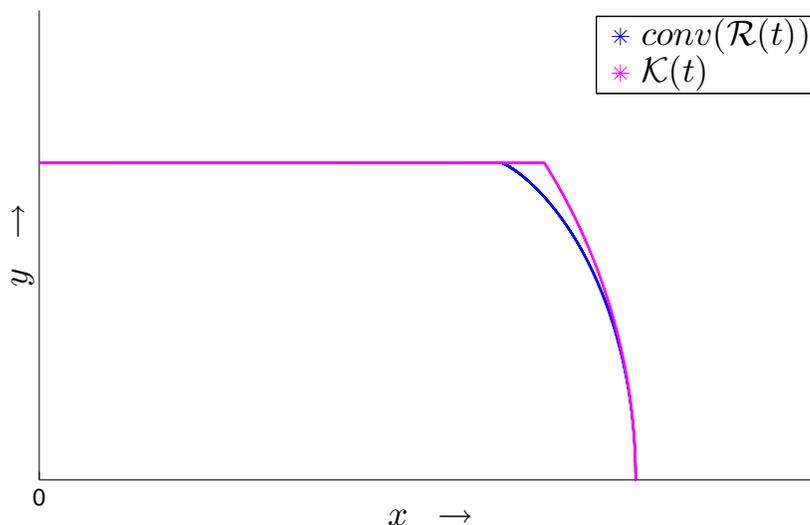


Figure 6: Approximating the involute of a circle with the arc of a circle in the first quadrant

**Proposition 1.** Define a set  $\mathcal{K}(t)$ ,

$$\mathcal{K}(t) = p \in \mathbb{R}^2 : \|p\| \leq t, |p_y| \leq \begin{cases} 1 - \cos(t), & \text{if } 0 < t \leq \pi/2 \\ t - \pi/2 + 1, & \text{if } t > \pi/2 \end{cases} \quad (14)$$

Then,  $\text{conv}(\mathcal{R}(t)) \subseteq \mathcal{K}(t)$  and

$$\lim_{t \rightarrow \infty} d_J(\text{conv}(\mathcal{R}(t)), \mathcal{K}(t)) = 0 \quad (15)$$

where  $d_J$  is the Jaccard distance.

*Proof.* Let  $p(t, q)$  denote a point on the curve  $C_c^{++}$  (see Equation 10). From Equation 11, we know that the parameter  $q$  lies in the interval  $[0, \min(t, \pi/2)]$ . The squared distance of this

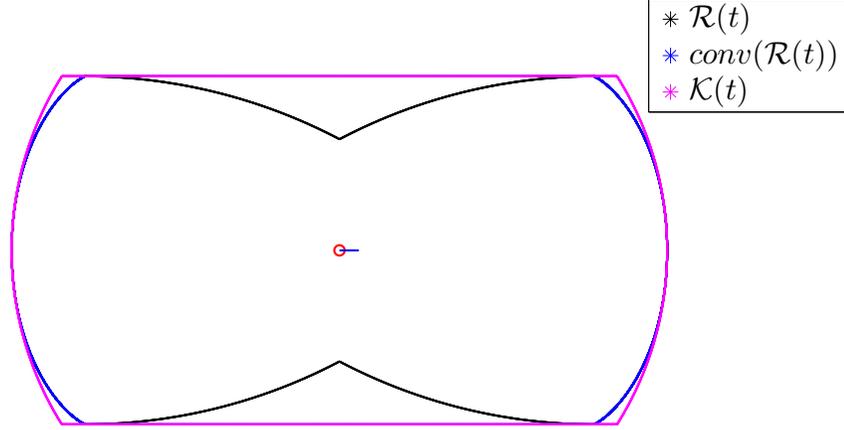


Figure 7: Simplification of  $\text{conv}(\mathcal{R}(t))$

curve from the origin is given as,

$$\|p(t, q)\|^2 = (\sin(q) + (t - q) \cos(q))^2 + (-\cos(q) + (t - q) \sin(q) + 1)^2 \quad (16)$$

$$= (t - q)^2 + 2 - 2 \cos(q) + 2(t - q) \sin(q) \quad (17)$$

where  $q \in [0, \min(t, \pi/2)]$ . Taking the derivative of  $\|p(t, q)\|^2$  with respect to the parameter  $q$  we get,

$$\begin{aligned} \frac{\partial \|p(t, q)\|^2}{\partial q} &= -2(t - q) + 2(t - q) \cos(q) \\ &= 2(t - q)(\cos(q) - 1) \leq 0, \forall q \in [0, \min(t, \pi/2)] \end{aligned} \quad (18)$$

Note that  $\|p(t, 0)\|^2 = t^2$ . Hence, this curve always has a distance of less than or equal to  $t$  from the origin and can be bounded by a circular arc of radius  $t$  (see Figure 5). Furthermore, note that

$$\max_{q \in [0, \min(t, \pi/2)]} p_y(t, q) = \begin{cases} 1 - \cos(t), & \text{if } 0 < t \leq \pi/2 \\ t - \pi/2 + 1, & \text{if } t > \pi/2 \end{cases} \quad (19)$$

From Equations 18 and 19, it follows that  $\text{conv}(\mathcal{R}(t)) \subseteq \mathcal{K}(t)$ . This proves the first part of the proposition.

Next, we show that as  $t$  grows larger, the dissimilarity between the two sets goes to zero. Since we are interested in an asymptotic result, only values of time greater than  $\pi/2$  are considered.

Since  $\text{conv}(\mathcal{R}(t)) \subseteq \mathcal{K}(t)$ ,  $\text{conv}(\mathcal{R}(t)) \cap \mathcal{K}(t) = \text{conv}(\mathcal{R}(t))$  and  $\text{conv}(\mathcal{R}(t)) \cup \mathcal{K}(t) = \mathcal{K}(t)$ . So, the Jaccard distance can be computed as,

$$d_J(\text{conv}(\mathcal{R}(t)), \mathcal{K}(t)) = 1 - \frac{A(\text{conv}(\mathcal{R}(t)))}{A(\mathcal{K}(t))} \quad (20)$$

Since both the sets are symmetric with respect to both  $x$  and  $y$  axis, we compute the area in the first quadrant,

$$A(\text{conv}(\mathcal{R}(t))) = \int_0^{t-\pi/2+1} p_x dp_y \quad (21)$$

where  $p \in C_c^{++}$  and is reproduced here for convenience,

$$\begin{aligned} p_x &= \sin(q) + (t - q) \cos(q) \\ p_y &= -\cos(q) + (t - q) \sin(q) + 1 \\ q &\in [0, \pi/2] \end{aligned}$$

The integral in Equation 21 can be written as,

$$A(\text{conv}(\mathcal{R}(t))) = \int_0^{\pi/2} (t - q) \sin(q) \cos(q) + (t - q)^2 \cos^2(q) dq \quad (22)$$

This is solved to obtain,

$$A(\text{conv}(\mathcal{R}(t))) = \frac{1}{48}(12\pi(t^2 - 1) + t(48 - 6\pi^2) + \pi^3) \quad (23)$$

We perform a similar analysis to obtain the area of  $\mathcal{K}(t)$  in the first quadrant.

$$A(\mathcal{K}(t)) = \int_0^{t-\pi/2+1} \sqrt{(t^2 - y^2)} dy$$

This gives,

$$A(\mathcal{K}(t)) = \frac{t^2}{2} \left( \sin^{-1} \frac{t - \pi/2 + 1}{t} + \frac{1}{2} \sin(2 \sin^{-1} \frac{t - \pi/2 + 1}{t}) \right) \quad (24)$$

Denote  $c = \pi/2 - 1$  and  $c/t = \delta(t)$ . Note that  $\delta(t) > 0 \forall t$ .

$$A(\mathcal{K}(t)) = \frac{t^2}{2} \left( \sin^{-1}(1 - \delta(t)) + \frac{1}{2} \sin(2 \sin^{-1}(1 - \delta(t))) \right) \quad (25)$$

Now, we take the limits of these expressions as time grows larger. As  $t \rightarrow \infty$ , the term  $t^2$  in Equation 23 dominates and so,

$$\lim_{t \rightarrow \infty} A(\text{conv}(\mathcal{R}(t))) = \frac{\pi}{4} t^2 \quad (26)$$

Furthermore, as  $t \rightarrow \infty$ ,  $\delta(t) \rightarrow 0$  in Equation 25. So,

$$\lim_{t \rightarrow \infty} A(\mathcal{K}(t)) = \frac{\pi}{4} t^2 \quad (27)$$

Plugging this into Equation 20,

$$\lim_{t \rightarrow \infty} d_J(\text{conv}(\mathcal{R}(t)), \mathcal{K}(t)) = 1 - \frac{\lim_{t \rightarrow \infty} A(\text{conv}(\mathcal{R}(t)))}{\lim_{t \rightarrow \infty} A(\mathcal{K}(t))} = 0 \quad (28)$$

This completes the proof. □

Simplifying  $\text{conv}(\mathcal{R}(t))$  using a justified approximation  $\mathcal{K}(t)$  will be useful in the next section where the structure of  $\mathcal{K}(t)$  will play an integral part in the computations.

### 3.3.3 Minimum Area Ellipse

In this section, expressions for the minimum area ellipse enclosing  $\mathcal{K}(t)$  are derived. The first part obtains the structure of the ellipse and the nature of the contact points between the ellipse and  $\mathcal{K}(t)$ . Following this, a non-linear programming problem is solved to obtain continuous time representations of the ellipse.

#### 3.3.3.1 Structure of the Ellipse

An ellipse (and more generally, an ellipsoid) can be represented uniquely by its center  $c$  and a transformation matrix  $H$ :  $E(c, H) = \{x \in \mathbb{R}^2 : (x - c)^T H (x - c) \leq 1\}$ . See Section 2.2 for a discussion on the construction of ellipses. Fritz John in [25] showed that for every convex body  $K \in \mathbb{R}^n$ , there exists a unique circumscribing ellipsoid of minimum volume. In our

case, we seek to find a minimum area ellipse circumscribing the convex set  $\mathcal{K}(t) \in \mathbb{R}^2$ . This can be represented by the semi-infinite programming problem,

$$\begin{aligned} \min_{c,H} & -\log \det(H) \\ \text{s.t.} & (z - c)^T H (z - c) \leq 1, \forall z \in \mathcal{K}(t) \end{aligned} \quad (29)$$

We denote the minimum volume ellipse as  $\xi(\mathcal{K}(t))$ .

To obtain the structure of the ellipse, we study and exploit the symmetry properties of the convex body  $\mathcal{K}(t)$ .

**Definition 10.** Denote  $\mathcal{O}(K)$  as a set of affine transformations  $T(x) = a + Ax$  which leave a set  $K \in \mathbb{R}^n$  unchanged.

$$\mathcal{O}(K) = \{T(x) = a + Ax : T(K) = K\} \quad (30)$$

This set is called the automorphism group of  $K$ . It has been shown in [26] that, if  $K$  is convex, the circumscribing minimum volume ellipsoid inherits the symmetry properties of  $K$ .

Applying this definition to our case,

$$\mathcal{O}(\mathcal{K}) \subseteq \mathcal{O}(\xi(\mathcal{K})) \quad (31)$$

Using this result, we have the following lemma.

**Lemma 1.** The extremal ellipse  $\xi(\mathcal{K}(t))$  has the form  $E(c, H)$  where  $c = 0$  and  $H = \text{diag}(A, B)$  for some  $A > 0, B > 0$ .

*Proof.* We know that  $\mathcal{K}(t)$  is symmetric about both the  $x$  and  $y$  axis and hence,  $-I_n \in \mathcal{O}(\mathcal{K}(t)) \subseteq \mathcal{O}(\xi(\mathcal{K}(t)))$ .

Also,  $T \in \mathcal{O}(\xi(\mathcal{K}(t))) \implies Tc = c$ . Choosing  $T = -I_n$ , we get  $c = 0$ .

We know from [32], that  $T \in \mathcal{O}(\xi(\mathcal{K}(t))) \implies T^T H T = H$ . The structure of  $H$  appears by applying  $T = -I_n$ . □

### 3.3.3.2 Contact Points

Contact points belong to the boundary of both  $\mathcal{K}(t)$  and  $\xi(\mathcal{K}(t))$  and satisfy certain properties. The following result by Fritz John [25] helps us understand the nature of these points and outlines some of their properties.

**Definition 11.** *If an ellipsoid  $E(c, H)$  is the minimum volume ellipsoid for a convex body  $K \in \mathbb{R}^n$ , then there exist points  $\{u_i\}_1^k$  and multipliers  $\lambda = (\lambda_1, \dots, \lambda_k) > 0$ ,  $0 \leq k \leq n(n+3)/2$  such that,*

$$\begin{aligned} 0 &= \sum_{i=1}^k \lambda_i (u_i - c) \\ u_i &= \partial K \cap \partial \xi(K), i = 1, \dots, k \\ K &\subseteq E(c, H) \end{aligned} \tag{32}$$

*The points  $\{u_i\}_1^k \in \partial K \cap \partial \xi(K)$  are called the contact points of  $\xi(K)$ . Furthermore,*

$$c \in \text{conv}(\{u_i\}_1^k) \tag{33}$$

*This immediately implies that, the contact points of  $\xi(K)$  are not contained in any closed halfspace whose bounding hyperplane passes through the center of  $\xi(K)$ .*

Using these facts, we have the following result,

**Lemma 2.** *The number of contact points of  $\xi(\mathcal{K}(t))$  is 4.*

*Proof.* Since  $\mathcal{K}(t)$  and  $\xi(\mathcal{K}(t))$  are symmetric about both the x,y axis (see Lemma 1) and since the number of contact points cannot be greater than 5, the possible number of contact points is 2 or 4. First we analyze the case of two contact points. They must lie on either the  $x$  axis or on the  $y$  axis (otherwise symmetry in all the 4 quadrants is not possible). But, from Definition 11, we know that the contact points cannot lie in any closed halfspace whose bounding hyperplane passes through the center of  $\xi(\mathcal{K}(t))$ . In both the situations, the  $x$  and  $y$  axis itself are hyperplanes which divide the space such that, both the contact points lie in a closed halfspace. Hence, the number of contact points is 4.  $\square$

### 3.3.3.3 Deriving the ellipse

In this section, we derive expressions for the ellipse  $\xi(\mathcal{K}(t))$ . As shown in Section 3.3.3.1,  $\xi(\mathcal{K}(t))$  is an ellipse of the form  $E(c, H)$  with  $c = 0$  and  $H = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$ . Results from 3.3.3.1 and 3.3.3.2 are used to state the following lemma.

**Lemma 3.** *The semi-infinite programming problem given in Equation 29 can be re-formulated as a non-linear programming problem:*

$$\begin{aligned} \min_A & -\log A - \log \frac{1 - A(t^2 - \alpha(t)^2)}{\alpha(t)^2} \\ \text{s.t.} & A \leq \frac{1}{t^2} \end{aligned} \quad (34)$$

$$\text{where } \alpha(t) = \begin{cases} 1 - \cos(t) & \text{if } 0 < t \leq \pi/2 \\ t - \pi/2 + 1 & \text{if } t > \pi/2 \end{cases}.$$

*Proof.* The feasibility condition  $\mathcal{K}(t) \subseteq E(0, H)$  can be translated into the condition that the quadratic function,

$$g(y) = A(t^2 - y^2) + B(y^2) - 1$$

is non-positive on the interval  $|y| \leq \alpha(t)$ .

From Section 3.3.3.2, we know that there must be atleast two points at which the function  $g(y)$  is zero. Since  $g(y)$  is a quadratic function, it follows that it must take zero values at the endpoints of  $y = \pm\alpha(t)$  (in no other situation can it take non-negative values in the given interval). So,

$$g(y) = \lambda(y - \alpha(t))(y + \alpha(t)), \text{ for some } \lambda \geq 0 \quad (35)$$

By equating the coefficients, the following constraints emerge,

$$B - A = \lambda \geq 0$$

$$At^2 - 1 + \lambda\alpha(t)^2 = 0$$

Eliminating  $\lambda$  from the equations, the two constraints are reduced to  $B \geq A$  and  $At^2 - 1 + (B - A)\alpha(t)^2 = 0$ . Expressing  $B$  in terms of  $A$  from the second constraint, the desired problem formulation is obtained.  $\square$

The next theorem solves the non-linear programming problem outlined above to obtain analytical expressions for the minimum area ellipses enclosing the reachable sets of unicycle robots.

**Theorem 1.** *The minimum area enclosing ellipse  $\xi(\mathcal{K}(t))$  has the form  $E(c, H)$ , where  $c = 0$  and  $H = \text{diag}(A, B)$ , where  $A > 0$ ,  $B > 0$  are given as follows:*

1. *If  $0 < t \leq \pi/2$ , then*

$$A = \frac{1}{2(t^2 - \alpha(t)^2)} \text{ and } B = \frac{1}{\alpha(t)^2} \quad (36)$$

*where  $\alpha(t) = 1 - \cos(t)$ .*

2. *If  $\pi/2 < t \leq (1 + \frac{1}{\sqrt{2}})(\pi - 2)$ , then*

$$A = \frac{1}{2(t^2 - \alpha(t)^2)} \text{ and } B = \frac{1}{\alpha(t)^2} \quad (37)$$

*where  $\alpha(t) = t - \pi/2 + 1$ .*

3. *If  $t > (1 + \frac{1}{\sqrt{2}})(\pi - 2)$ ,*

$$A = \frac{1}{t^2} \text{ and } B = \frac{1}{t^2} \quad (38)$$

*Proof.* We first search for optimal values of  $A$  in the interior of the constraint set, i.e. when  $A < \frac{1}{t^2}$ . Denoting the cost function in Equation (34) as  $f(A)$ ,

$$\begin{aligned} \nabla f(A) &= 0 \\ \implies \nabla f(A) &= -\frac{1}{A} + \frac{t^2 - \alpha(t)^2}{1 - A(t^2 - \alpha(t)^2)} = 0 \\ \implies A &= \frac{1}{2(t^2 - \alpha(t)^2)} \end{aligned}$$

For the feasibility condition to hold true,

$$t^2 > \alpha(t)^2$$

which is true for all values of  $t$  in the interval  $(0, \pi/2]$ . For  $t > \pi/2$ , this holds whenever the function  $t^2 + 4(1 - \pi/2)t + 2(1 - \pi/2)^2$  takes non-positive values. This is true for  $0 < t \leq (1 + \frac{1}{\sqrt{2}})(\pi - 2)$ . For all values of  $t > (1 + \frac{1}{\sqrt{2}})(\pi - 2)$ , the minimum in the feasible set is obtained when  $A = 1/t^2$ . This completes the proof.  $\square$

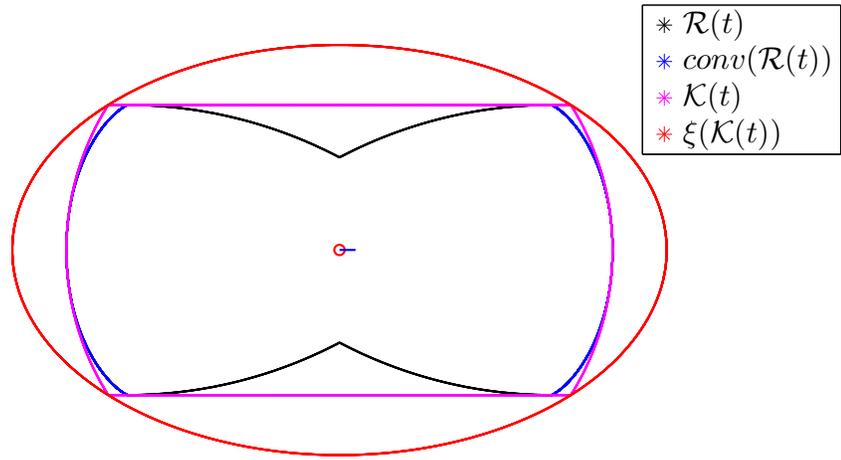


Figure 8: Minimum Area Ellipse enclosing  $\mathcal{K}(t)$

### 3.3.3.4 Convergence Results

In the previous section, analytical expressions for the minimum area ellipse enclosing  $\mathcal{K}(t)$  were developed. In this section we investigate the dissimilarity between the set  $\mathcal{K}(t)$  and the minimum area ellipse enclosing it. This is done using the Jaccard distance and is similar to the analysis done in Section 3.3.2.1. The following theorem is one of the main results in this thesis.

**Theorem 2.** *Let  $\mathcal{R}(t)$  denote the reachable set of a unicycle robot with dynamics given in Equation 8,  $\text{conv}(\mathcal{R}(t))$  denote its convex hull,  $\mathcal{K}(t)$  denote an approximation of the convex*

hull as defined in Proposition 1 and let  $\xi(t)$  be the minimum area ellipse enclosing  $\mathcal{K}(t)$ . Then,

$$\lim_{t \rightarrow \infty} d_J(\text{conv}(\mathcal{R}(t)), \xi(t)) = 0$$

*Proof.* We first compute  $d_J(\xi(t), \mathcal{K}(t))$ . Since  $\mathcal{K}(t) \subseteq \xi(t)$ ,  $\mathcal{K}(t) \cup \xi(t) = \xi(t)$  and  $\mathcal{K}(t) \cap \xi(t) = \mathcal{K}(t)$ . So,

$$d_J(\xi(t), \mathcal{K}(t)) = 1 - \frac{A(\mathcal{K}(t))}{A(\xi(t))} \quad (39)$$

Again, due to the asymptotic nature of the result, we only consider the case when  $t > \pi/2$ . For these values of time,  $\xi(t)$  is a circle of radius  $t$ . Also, since both  $\xi(t)$  and  $\mathcal{K}(t)$  are symmetric about the  $x$  and  $y$  axis, we only compute areas in the first quadrant. So,  $A(\xi(t)) = \pi t^2/4$ . From Equation 25, we know the expression for  $A(\mathcal{K}(t))$ .

$$A(\mathcal{K}(t)) = \frac{t^2}{2} \left( \sin^{-1}(1 - \delta) + \frac{1}{2} \sin(2 \sin^{-1}(1 - \delta)) \right)$$

$$\text{where } \delta(t) = c/t$$

Taking the limit,

$$\lim_{t \rightarrow \infty} \delta(t) = 0 \implies \lim_{t \rightarrow \infty} A(\mathcal{K}(t)) = \pi \frac{t^2}{4}$$

which is equal to the area of  $\xi(t)$  in the first quadrant. Thus,

$$\lim_{t \rightarrow \infty} d_J(\mathcal{K}(t), \xi(t)) = 0 \quad (40)$$

Since the Jaccard distance is a metric distance, it obeys the triangle inequality,

$$d_J(\text{conv}(\mathcal{R}(t)), \xi(t)) \leq d_J(\text{conv}(\mathcal{R}(t)), \mathcal{K}(t)) + d_J(\mathcal{K}(t), \xi(t))$$

Applying Equation 28,

$$\lim_{t \rightarrow \infty} d_J(\text{conv}(\mathcal{R}(t)), \xi(t)) = 0$$

This completes the proof. □

Figure 9 shows the evolution of  $\mathcal{R}(t)$ ,  $\text{conv}(\mathcal{R}(t))$ ,  $\mathcal{K}(t)$  and  $\xi(\mathcal{K}(t))$  for different values of time. The time scale for each figure is different. Within a relatively small time frame, all four sets converge and become circular in shape. This can be attested by the asymptotic nature of the convergence.

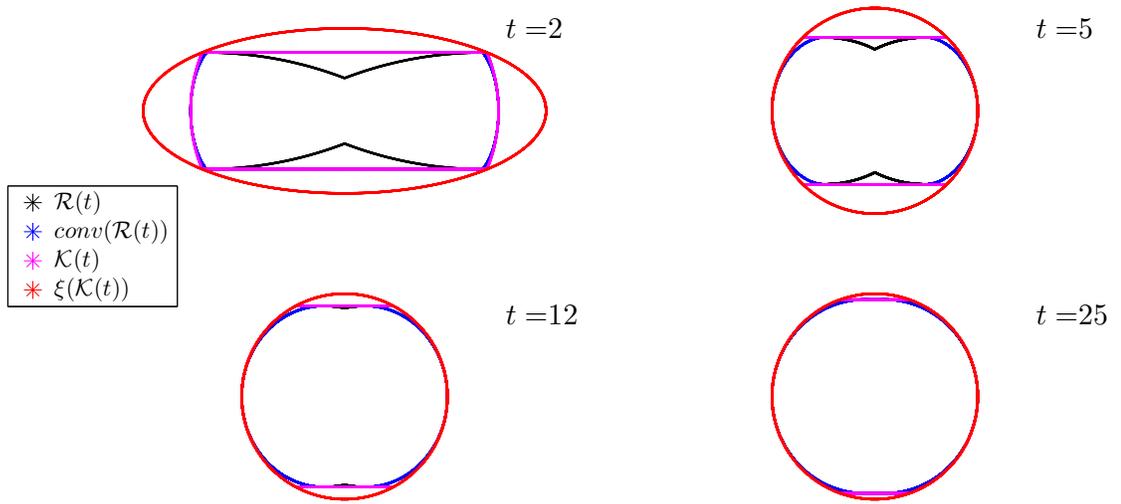


Figure 9: Evolution of  $\mathcal{R}(t)$ ,  $\text{conv}(\mathcal{R}(t))$ ,  $\mathcal{K}(t)$ ,  $\xi(\mathcal{K}(t))$  for  $t = 2, 5, 12, 25$ .

# CHAPTER 4

## SAFE TIME HORIZON COMPUTATION

In the previous chapter we computed the exact structure of the reachable sets for unicycle robots and obtained analytical expressions for ellipsoidal approximations of these sets. In this chapter, we formally define the safe times (Section 4.2), incorporate them in the motion strategy of the robots (Section 4.3), prove the safety guarantees they provide (Section 4.4) and explain mechanisms to compute them (Section 4.5). This is followed by a discussion on the computational burden of the algorithm.

### 4.1 Problem Formulation

Consider a team of  $N$  robots in an enclosed 2D space. Each robot is a unicycle robot with dynamics as specified in Equation 8. Let  $z_i(t) = (x_i(t), y_i(t))$  denote the position of robot  $i$  at time  $t$ . The robots do not have sensing or decision making capabilities and solely rely on a central host for their motion commands. The host has access to the global positions of the robots. At regular intervals of time  $t_k = k\delta, k \in \mathbb{N}$ , the host transmits data to all the robots via a wireless communication channel where  $1/\delta$  is called the update frequency (Figure 10). Each data packet includes the desired velocities and the safe times of the robot. Denote the velocity command sent to robot  $i$  as  $u_i(t_k) = (v_i(t_k), \omega_i(t_k))$  and the safe time as  $s_i(t_k)$ .

The safe time for each robot is bounded by a pre-determined value  $L$ . This value can be viewed as a design parameter and typically depends on the size of the testbed and the number of robots. This is also important for numerical stability of the computation algorithm in special cases, for e.g. when there is only a single robot on the test bed.

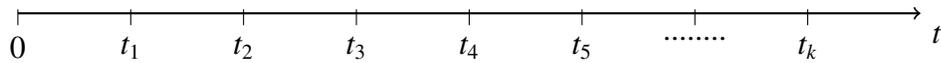


Figure 10: Updates times  $t_k$  at which commands are transmitted to the robots.

The parameter  $L$  can be used to impose a graph structure on the team of robots and introduce the notion of a neighbor.

**Definition 12.** A robot  $j$  is considered a neighbor of robot  $i$  if the distance between them can be covered in less than the maximum safe time. Let  $N_i(t)$  denote the neighborhood set of robot  $i$  at time  $t$ ,

$$N_i(t) = \{j \in \{1, \dots, N\}, j \neq i : \|z_i(t) - z_j(t)\| < 2Lv_m\}$$

where  $L$  is the upper bound on the safe time and  $v_m$  is the maximum linear velocity that the robots can achieve.

This implies that if robot  $i$  and robot  $j$  are not neighbors based on this definition, they cannot collide within the maximum safe time  $L$ .

We now describe the behavior of the robots after experiencing communication failure.

**Definition 13.** After a robot experiences communication failure, it executes its last received velocity command repeatedly for the duration of the safe time horizon. Thus, the trajectory of the robot in this circumstance will be circular if  $\omega \neq 0$  and will be a straight line if  $\omega = 0$  (Figure 11). Let  $Z_f(\mu, z_i(t_k))$  denote the position of the robot  $i$  along this trajectory,

$$Z_f(\mu, z_i(t_k)) = \begin{cases} z_i(t_k) + \frac{v_i(t_k)}{\omega_i(t_k)} \begin{pmatrix} \sin(\omega_i(t_k)\mu + \phi_i(t_k)) - \sin(\phi_i(t_k)) \\ \cos(\phi_i(t_k)) - \cos(\omega_i(t_k)\mu + \phi_i(t_k)) \end{pmatrix}, & \text{if } \omega_i(t_k) \neq 0 \\ z_i(t_k) + v_i(t_k) \begin{pmatrix} \cos(\phi_i(t_k)) \\ \sin(\phi_i(t_k)) \end{pmatrix}, & \text{if } \omega_i(t_k) = 0 \end{cases} \quad (41)$$

where  $t_k$  is the time of the last received command and  $\mu$  is the time elapsed since the communication failure.

Since  $Z_f(\mu, z_i(t_k))$  denotes the position of a robot after experiencing communication failure, it will play a crucial role in computing and analyzing the safe times. Refer to Appendix B for the derivation of  $Z_f(\mu, z_i(t_k))$ .

We now have the necessary definitions and expressions to formally define the safe time.

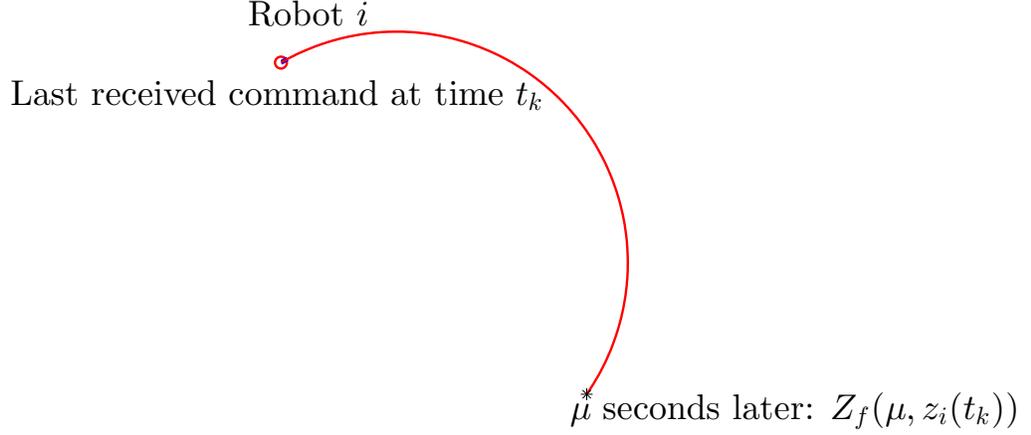


Figure 11: Trajectory of robot  $i$  after communication failure:  $v_i(t_k) = 0.25, \omega_i(t_k) = -0.9$

## 4.2 Defining the Safe Time

As discussed in Chapter 1, the formal definition of the safe time horizon is motivated by the following design question:

*What is the maximum time for which a robot can move in a provably safe manner after experiencing communication failure in a multi-robot system?*

**Definition 14.** Let  $s_i(t_k)$  denote the safe time horizon of robot  $i$  computed at time  $t_k$ . It can be defined as,

$$s_i(t_k) = \max_{\lambda} \int_0^{\lambda} 1 \, d\lambda$$

$$Z_f(\mu, z_i(t_k)) \cap \xi(\mathcal{R}(\mu; z_j(t_k), \phi_j(t_k))) = \emptyset, \forall \mu \in [0, \lambda], \forall j \in N_i$$

$$\lambda \leq L$$

where  $\mathcal{R}(\mu; z_j(t_k), \phi_j(t_k))$  is the reachable set for robot  $j$  for a time horizon of  $\mu$  seconds,  $\xi(\cdot)$  denotes the minimum area ellipse enclosing the reachable set and  $\emptyset$  denotes the empty set.

Thus, the safe time can be thought of as the longest amount of time for which the trajectory of the robot does not intersect the reachable sets of its neighbors. We are now ready to formulate the motion strategy of the robots.

### 4.3 Motion Strategy

This section discusses how the safe time can be incorporated into the motion strategy of the robots. As discussed earlier, the host transmits  $(u_i(t_k), s_i(t_k))$  to all the robots  $i \in \{1, \dots, N\}$  at regular time intervals  $t_k$ . The communication link between a robot  $i$  and the host is represented by the variable  $c_i$  which is defined as

$$c_i(t_k) = \begin{cases} 1, & \text{if } (u_i(t_k), s_i(t_k)) \text{ was received} \\ 0, & \text{if } (u_i(t_k), s_i(t_k)) \text{ was not received} \end{cases} \quad (42)$$

From this definition, it is clear that  $c_i$  changes value only when  $t = t_k$  for some  $k$ . The following algorithm outlines the motion strategy that the robots employ. For robot  $i$ :

```

while true do
  if  $c_i(t_k) = 1$  then
    Execute  $u_i(t_k)$ 
  else
    Execute  $u_i(t_{k-1})$  for  $s_i(t_{k-1})$  seconds then Stop Moving
  end if
   $k = k + 1$ 
end while

```

Figure 12 illustrates the rationale behind the computation of the safe time. The robot experiencing communication failure is shown in red. The other robots are depicted using grey circles. The reachable sets of these robots are depicted as growing blue ellipses. The red robot can continue to move safely until its path intersects the reachable set of another robot. After this, the robot stops moving.

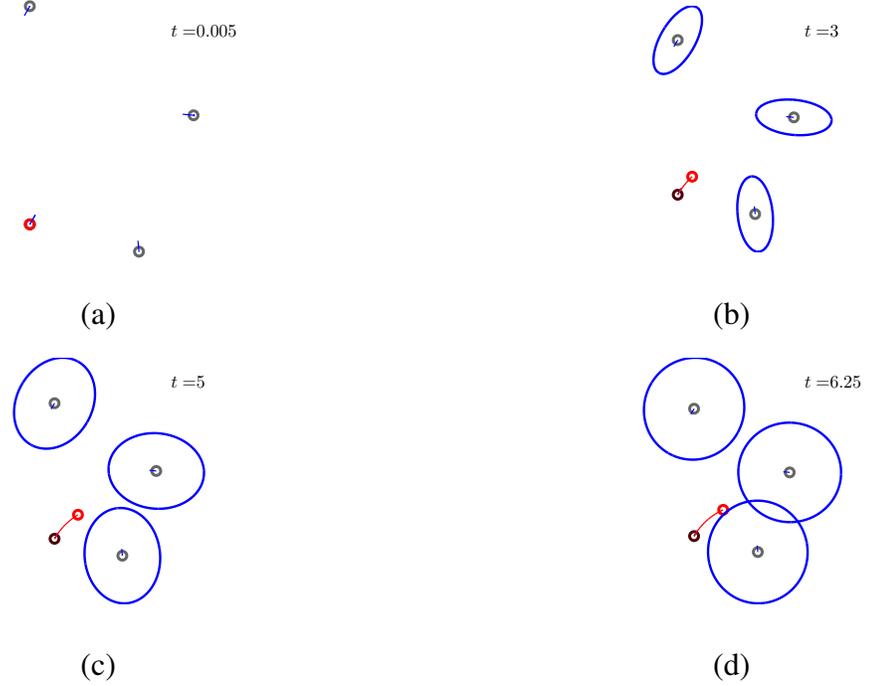


Figure 12: Rationale behind safe time computation. The red robot can move in an open-loop fashion until its path intersects the reachable set of another robot.

#### 4.4 Safety Guarantees

Before stating the safety guarantees provided by the safe time algorithm, some assumptions are made on the capabilities of the control algorithm running on the host. It is expected that collision avoidance between communicating robots is ensured by the control algorithm. The following definition formally describes the assumptions.

**Definition 15.** *The control algorithm running on the host ensures collision avoidance among communicating robots and between communicating robots and stationary obstacles.*

If  $c_i(t_k) = 1 \quad \forall i \in \{1, \dots, n\}$ ,  $u_i(t_k)$  guarantees that,

$$\|z_i(t_k) - z_j(t_k)\| > 0 \implies \|z_i(t_{k+1}) - z_j(t_{k+1})\| > 0, \forall i, j \in \{1, \dots, N\}, i \neq j \quad (43)$$

If  $z_O$  denotes the position of a stationary obstacle, then,

$$\|z_i(t_k) - z_O\| > 0 \implies \|z_i(t_{k+1}) - z_O\| > 0, \forall i \in \{1, \dots, N\} \quad (44)$$

However, safety is not guaranteed in situations where the robots are moving but are not under direct control of the host. We are interested in this specific case because when executing the safe time algorithm, the robots continue to move even when they are not receiving new commands from the host.

In the next theorem, it is proved that the safe time based motion strategy ensures that, in the event of a communication failure, a robot can continue to move for the duration of the safe time, without colliding with any other robot. Thus, safety is not compromised even if the robot is moving without any direct commands from the host.

**Theorem 3.** *If  $c_i(t_m) = 0, \forall m > k$ ,*

$$\begin{aligned} \|z_i(t_k) - z_j(t_k)\| > 0 &\implies \|z_i(t_k + \mu) - z_j(t_k + \mu)\| > 0, \\ \forall \mu \in [0, s_i(t_k)], \forall j \in \{1, \dots, N\}, j \neq i \end{aligned}$$

*Proof.* Since  $\|z_i(t_k) - z_j(t_k)\| > 0 \forall j \in N_i, s_i(t_k) > 0$ . We have already proved that if  $j \notin N_i$ , then a collision is not possible between robot  $i$  and  $j$  within the safe time (Definition 12).

Let  $j \in N_i$ . From the definition of  $s_i(t_k)$ , we know that,

$$\begin{aligned} z_i(t_k + \mu) &\notin \xi(\mathcal{R}(\mu, z_j(t_k), \phi_j(t_k))) \\ \forall \mu \in [0, s_i(t_k)], \end{aligned} \tag{45}$$

From the definition of reachable sets,

$$\begin{aligned} z_j(t_k + \mu) &\in \xi(\mathcal{R}(\mu, z_j(t_k), \phi_j(t_k))) \\ \forall \mu \in [0, s_i(t_k)] \end{aligned}$$

From these two statements, it is clear that

$$z_i(t_k + \mu) \neq z_j(t_k + \mu) \quad \forall \mu \in [0, s_i(t_k)] \tag{46}$$

Hence,

$$\|z_i(t_k + \mu) - z_j(t_k + \mu)\| > 0 \quad \forall \mu \in [0, s_i(t_k)], \forall j \in N_i \tag{47}$$

This completes the proof. □

This implies that robot  $i$  can continue to execute its last received motion command in an open-loop fashion and is guaranteed to not collide within the safe time horizon. Beyond this horizon, the robot stops moving and essentially behaves as a stationary obstacle for the other robots. At this point, Equation 44 ensures that no collisions take place. Thus, the original safety guarantee of the control algorithm is retained. This theorem also applies to the case when the communication is re-established before the safe time duration ends. In this case, the robot would simply resume normal operation.

## 4.5 Computing the Safe Time

In order to compute the safe time as defined in Definition 14, intersection tests must be performed between the trajectories of the robots, represented by  $Z_f(\mu, z_i(t_k))$  and the ellipses enclosing the reachable sets of its neighbors, denoted by  $\xi(\mathcal{R}(\mu; z_j(t_k), \phi_j(t_k)))$ . This computation must be performed in real-time. In this section we outline an optimization based strategy to efficiently compute the safe times.

### 4.5.1 Optimization Based Strategy

The concept of a pairwise safe time is crucial in the formulation of the optimization strategy.

**Definition 16.** *Let  $s_{ij}(t_k)$  denote the pairwise safe time computed for robot  $i$  with respect to robot  $j$  at time  $t_k$ . This denotes the time at which the trajectory  $Z_f(\mu, z_i(t_k))$  of robot  $i$  intersects the reachable set of its neighbor  $j$ . So, the safe time  $s_i(t_k)$  for robot  $i$  will simply be the smallest pairwise safe time over all the neighbors.*

$$s_i(t_k) = \min_{j \in N_i} s_{ij}(t_k) \quad (48)$$

We formulate a technique to compute the pairwise safe times  $s_{ij}(t_k)$ , then take the minimum value to obtain the safe time  $s_i(t_k)$ .

An optimization based technique is used to quickly compute the pairwise safe times. The optimization problem is solved using a gradient descent algorithm.

**Definition 17.** *The optimization problem to obtain pairwise safe times can be formulated as*

$$s_{ij}(t_k) = \arg \min_{\mu} J_{ij}(\mu, t_k) = Z_f(\mu, z_i(t_k))^T H_j(\mu, t_k) Z_f(\mu, z_i(t_k)) \quad (49)$$

$$s.t. \quad Z_f(\mu, z_i(t_k))^T H_j(\mu, t_k) Z_f(\mu, z_i(t_k)) > 1$$

where  $H_j(\mu, t_k)$  denotes the transformation matrix of the ellipse  $\xi(\mathcal{R}_j(\mu, z_j(t_k), \phi_j(t_k)))$  and as defined earlier,  $Z_f(\mu, z_i(t_k))$  represents the trajectory taken by robot  $i$  after experiencing communication failure at  $t_k$ .

The cost term here is the level set expression of an ellipse. If it is less than 1, it indicates that the point  $Z_f(\mu, z_i(t_k))$  is inside the ellipse  $\xi(\mathcal{R}_j(\mu, z_j(t_k), \phi_j(t_k)))$ . Intuitively, this optimization problem computes the time at which the trajectory of robot  $i$  gets the closest to the boundary of the ellipse. A simple gradient descent algorithm can be used to find a solution to this problem. By using the Armijo step size rule [39], fast convergence can be obtained. The algorithm for the gradient descent is elaborated below:

Initialize  $\mu_0$  s.t.  $J_{ij}(\mu_0, t_k) > 1$

Set  $r = 0$

**repeat**

    Compute the gradient  $\nabla_{\mu} J_{ij}(\mu_r, t_k)$

    Compute the Armijo step size  $\gamma_r$

    Move along the negative gradient:  $\mu_{r+1} = \mu_r - \gamma_r \nabla_{\mu} J(\mu_r, t_k)$

$r = r + 1$

**until**  $J(\mu_r, t_k) > 1$

$s_{ij}(t_k) = \mu_{r-1}$

#### 4.5.2 Computational Burden

Due to the geometric dependence of the algorithm, it is difficult to derive an upper bound on the complexity of the safe time computation algorithm. In the previous section, it was

seen that the safe time horizon for a robot can be given as the minimum of the pairwise safe times with all its neighbors.

$$s_i(t_k) = \min_{j \in N_i} s_{ij}(t_k)$$

Hence, the number of optimization problems that need to be solved by the host depends on the number of robots and the size of the neighborhood sets of the robots. Since the maximum safe time  $L$  directly affects the size of the neighborhood sets, it can be viewed as a design parameter which can be manipulated to obtain a trade-off between the maximum safe time and the computation times. If  $N_i^{max}$  represents the largest neighborhood set possible in the network, then the worst case complexity is proportional to  $N|N_i^{max}|$  where  $|\cdot|$  represents the cardinality of the set. Empirical testing shows that for a practical value of  $L$  and appropriately chosen velocity bounds, the neighborhood sets do not increase linearly with an increase in the number of robots.

To better demonstrate this point, we have used simulations to compare the network size  $N$  with the computation time associated with calculating the safe times. The number of robots  $N$  is varied from 1 to 86 in increments of 5 ( $N = 1, 6, 11, \dots, 86$ ). For each  $N$ , the  $x$  and  $y$  coordinates of the robots and their orientation  $\phi$  were generated randomly. The coordination control algorithm being executed was a multi-robot patrol algorithm. For each  $N$ , the total time required to compute the safe times at each iteration was averaged over the duration of the algorithm to obtain representative estimates. The simulation was performed on a test system equipped with a Intel i7 processor running at 2.30GHz and 6GB of DDR3 memory. We used Windows 7 as the operating system running Matlab 2012a as the simulator. The CPU time associated with the computation of the safe times was recorded (see Figure 13)

The plot suggests a quadratic relationship between network size and the computational time when the number of robots are relatively small. This tends towards a linear relation as the number of robots increases. This is expected because for small team sizes, an increase in the number of robots leads to a significant increase in the sizes of the neighborhood

sets. As the team size grows, the size of the neighborhood set saturates and we see a linear relationship between the network size and the computation time.

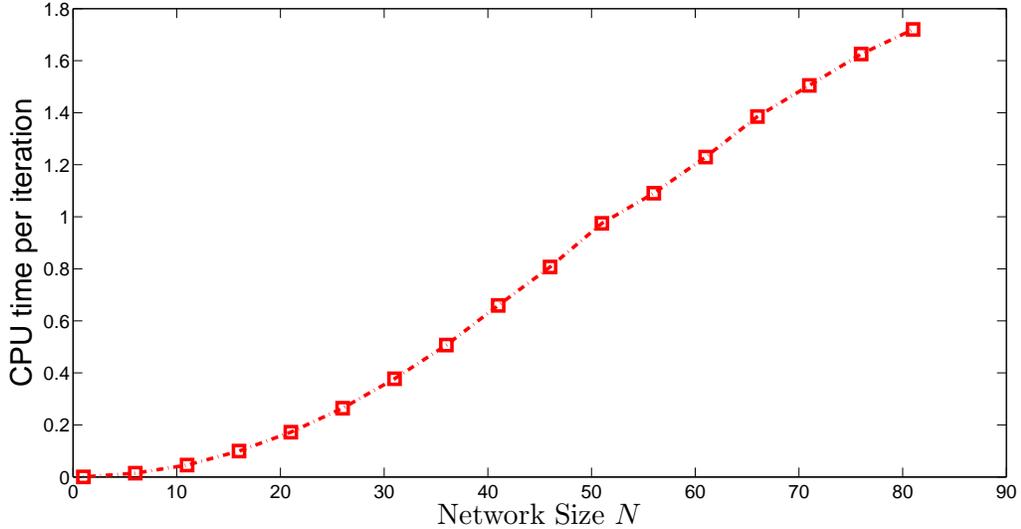


Figure 13: Computational Burden vs. Network Size

## 4.6 Simulations

To verify and supplement our theory, we implemented the proposed motion strategy in Matlab and simulated it using a network of unicycle robots. For this purpose, a virtual multi-robot testbed consisting of a central host and a team of robots, was created. The host has access to global position information of all the robots. The simulation scenario depicts a team of robots continuously patrolling a simulated corridor (Figure 14) by following a series of waypoints denoted by red squares. At each simulation time step, the centralized host computes the desired velocities and the safe times of the robots which are sent to the robots via a simulated communication channel. These velocities are applied as control inputs to the robots whose positions are updated using the dynamics given in Equation 3. Communication failures are injected into the system at random time intervals and for random time durations. Any number of robots can be experiencing failure at any given time. A red circle is used to indicate a robot experiencing communication failure and a

green circle is used to indicate a communicating robot. The maximum linear velocity of all the robots is chosen as  $0.254m/sec$  and the maximum angular velocity as  $2\pi rad/sec$ . These velocities are chosen from the actual specifications of the GRITSBot: a miniature unicycle robot being developed at Georgia Tech [40]. The maximum safe time  $L$  is chosen as 2 seconds. Figure 14 shows the simulation setup. The safe times are indicated beside the robots.

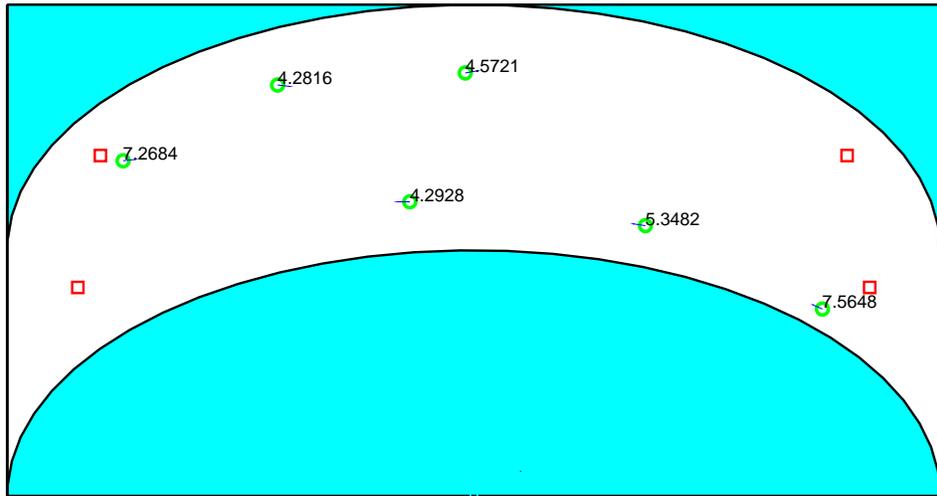
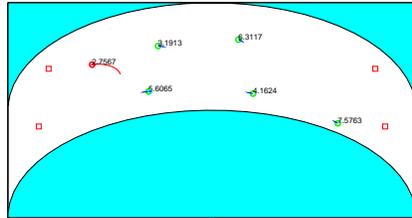


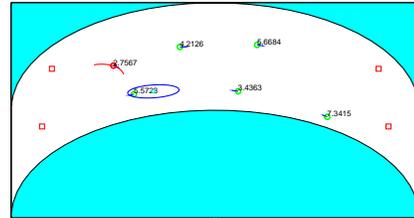
Figure 14: Multi-robot Patrol with Safe Times

#### 4.6.1 Single Robot Failure

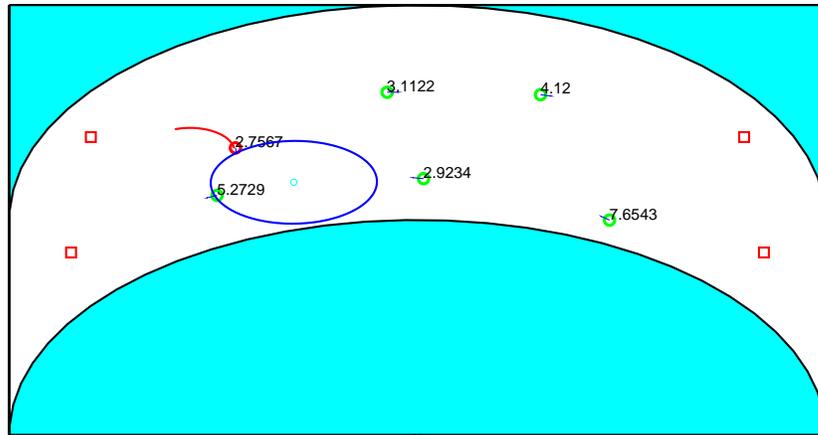
Figure 15 illustrates the situation where one of the robots experiences a communication failure. The red path indicates the trajectory that the robot will take as it executes the last received velocity command for the duration of the safe time. The blue ellipse represents the reachable set of the neighboring robot corresponding to the smallest pair wise safe time  $s_{ij}(t_k)$ . The robot can move until its path intersects the blue ellipse. After the safe time horizon, the robot stops moving.



(a)



(b)



(c)

Figure 15: Single robot communication failure. The red curve indicates the path taken by the robot and the blue ellipse is the reachable set of a neighboring robot.

### 4.6.2 Multiple Robot Failure

In the next scenario, two robots simultaneously experience communication failure on the testbed. As shown in Figure 16, both robots continue executing their last received velocity commands for their respective safe times and remain safe despite having no external information about other robots. From these simulations, we are able to verify that the developed

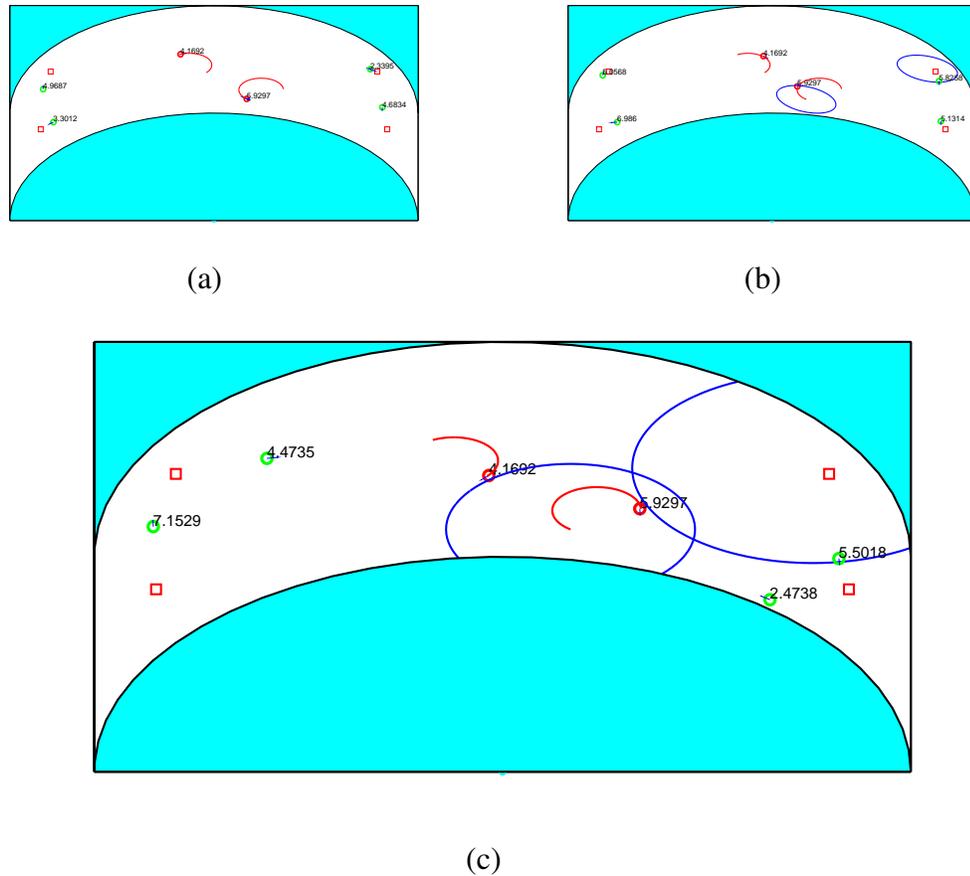


Figure 16: Double robot communication failure. Each robot executes its last received velocity for the respective safe time horizon.

algorithm generates robust and safe motion patterns for robots in the face of communication failures. In the next chapter, we implement this algorithm on a team of real robots.

# CHAPTER 5

## EXPERIMENTAL SETUP AND RESULTS

Since the safe time algorithm proved to be robust and stable in simulation, it was desirable to verify its performance on a physical multi-robot testbed. The experiment was performed on a testbed at the Georgia Robotics and Intelligent Systems (GRITS) Lab. The experiment consisted of using a team of 4 Khepera III robots in conjunction with a Opti-Track 3-D motion capture system which was used to obtain the global positions of the robots. A desktop computer connected to the motion capture system served as the host running the control algorithms. This chapter details the equipment used and the methodology of the experiment. Finally, a presentation and discussion of the results follow.

### 5.1 Experimental Setup

#### 5.1.1 Equipment

A multi-robot testbed consists of four fundamental components which are crucial for its operation: a global position tracking system, mobile robots with the ability to move in a plane, a communication network capable of transferring data packets to and from the robots, and a centralized host computer capable of managing software related tasks and running the control algorithm. We now briefly describe the specific components used for this experiment.

##### *5.1.1.1 Khepera III Robots*

The Khepera III robot is a small, modular, robotic platform specifically designed for multi-robot experiments. The robot is driven via a two wheel differential drive and therefore can be considered as a unicycle robot. The robot base includes an array of 9 Infrared sensors for obstacle detection as well as 5 Ultrasonic sensors for long range object detection [41]. The sensors were not used in this experiment. Additionally, an 802.11g compact flash wireless card is used to connect the robot to the local area network. Figure 17 illustrates the physical

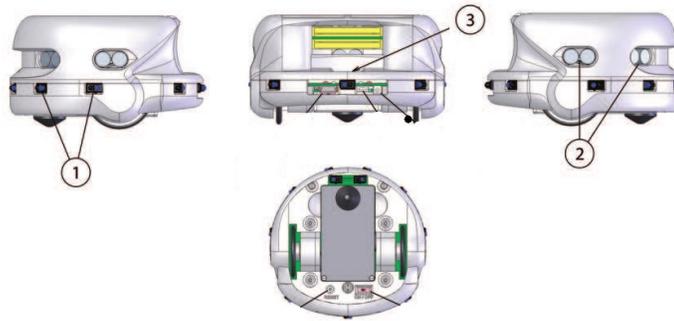


Figure 17: The design of a Khepera III Robot

design of the Khepera III robots. Figure 18 shows a picture of an actual Khepera robot. Working with one or more of the Kheperas is typically done by connecting them to the



Figure 18: The Khepera III Robot

same wireless local area network. Each robot is equipped with a unique IP address which is hard coded into the embedded processor on the robot.

### *5.1.1.2 OptiTrack Motion Capture System*

The Georgia Robotics and Intelligent Systems (GRITS) Lab contains a OptiTrack motion capture system capable of tracking the 3-dimensional pose of multiple bodies simultaneously. The system operates by tracking small spherical reflectors using a set of 12 cameras. These spherical reflectors are attached to the robots. Each camera is surrounded by LEDs that emit light of a particular wavelength. This light is then reflected by the markers directly back towards the camera. The set of 12 cameras are calibrated so their precise relative locations are known. This way, if a marker is seen by at least two cameras its precise location in 3-D space can be calculated. Markers can be localized with an accuracy of 5mm.

### *5.1.1.3 Communication Infrastructure*

A local area WiFi network is used to facilitate all the communication between the robots and the host. The commands generated by the host are converted to strings and sent over the network as a User Datagram Protocol(UDP) message. Messages can be sent to specific robots since they have unique IP addresses in the network.

## **5.2 Methodology**

We briefly discuss the methodology of the experiment. At the start of the experiment, 4 Khepera robots are lined up in view of the OptiTrack cameras. Each robot has a unique marker pattern, and within the OptiTrack software we define which markers belong to each robot. Once configured, the OptiTrack software will continuously report the 3D pose of every robot that has been entered into the system. A custom piece of software delivers the data generated by the OptiTrack system to the host computer. The host computer uses this data to close the position feedback loop in the control algorithm. Simultaneously, the host computer also uses the position data and velocities to compute the safe times for each robot. This information is then concatenated into a single data string and transmitted as a UDP message to all the robots. This process repeats at a rate of 5Hz. An overhead projector allows us to project useful information down onto the floor of the testbed such as

the simulated topology and connectivity status of the robots.

Communication failures in the network are simulated in a random fashion similar to what was described in Section 4.6. The Khepera robots used for this experiment were pre-configured to stop moving if they didn't receive messages over the network. Hence, it was not feasible to directly test the safe time algorithm, by not sending communication packets to the robots. To get around this problem, a simulated communication failure was implemented, wherein the packets being sent after the failure did not contain any new velocity or safe time information. When the robots stopped receiving new information, they were programmed to execute their last received velocity command for the duration of the safe time. As with the simulation, blue ellipses were drawn depicting the reachable set of the neighbor corresponding to the smallest pairwise safe time.

The experiment implements a multi-robot patrol algorithm on a team of four robots. As described in Section 4.6, the robots follow a series of waypoints to move along the corridor (seen in Figure 19 as red squares). The maximum velocity of the robots was limited to  $0.3m/sec$  and the maximum angular velocity was  $2\pi rad/sec$ . The parameter  $L$  for this experiment was chosen to be 3 seconds, which is the time required by a robot to traverse half the testbed at maximum velocity. A green dot is projected over robots which are receiving commands successfully over the network. When a robot loses connection and stops receiving messages, a red dot is projected over it. Figure 19 shows the experimental setup, with four Khepera robots in active patrol around the corridor.

In the next section, the results from the experiment are discussed and illustrated.

### **5.3 Results**

Using the equipment and methods described above, the safe time algorithm was successfully demonstrated on 4 Khepera robots. Three scenarios observed during the experiment are discussed below. For each scenario, the figures illustrate the algorithm in action.

1. It was verified that when long communication failures were experienced, the safe

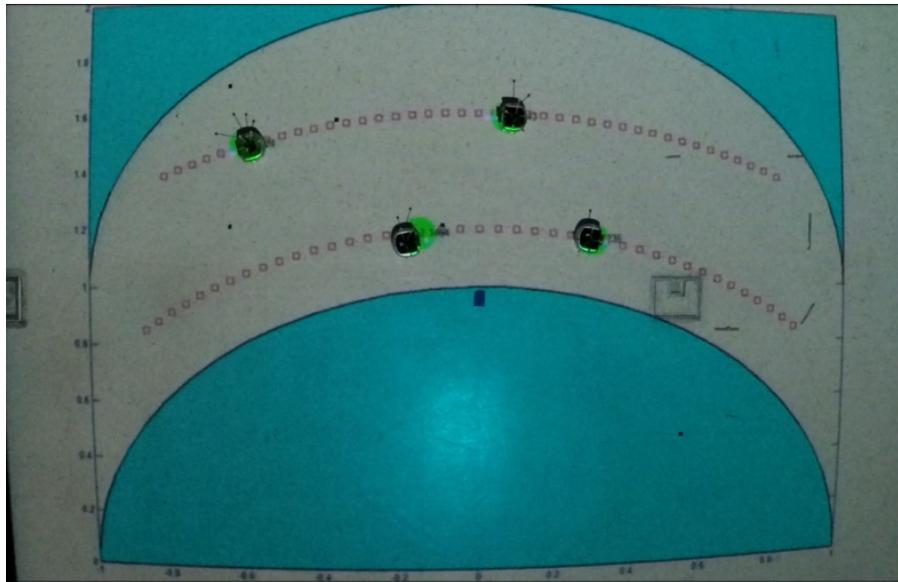


Figure 19: Four Khepera Robots patrolling a corridor

time algorithm allowed the robots to continue moving along the trajectory for a longer duration of time thereby reducing the disruption to the multi-robot patrol algorithm (Figure 20).

2. By simulating small duration communication failures, it was verified that the safe time algorithm successfully prevented jerky motion patterns and allowed the robots to continue moving smoothly.(Figure 21).
3. Multiple communication failures were tested under various circumstances and it was verified that there were no collisions caused due to the continued movement of the robots (Figure 22).

These three scenarios summarize the salient features of this algorithm.

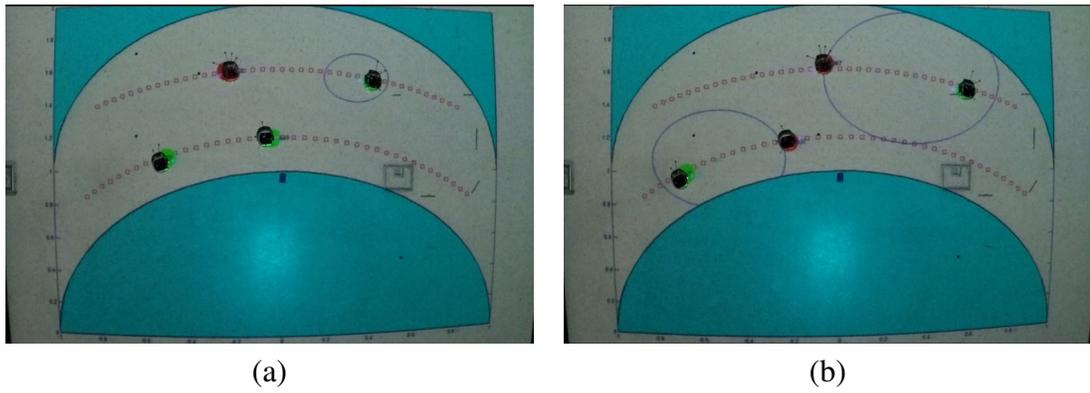


Figure 20: The safe time algorithm reduces the disruption in motion caused by communication failures.

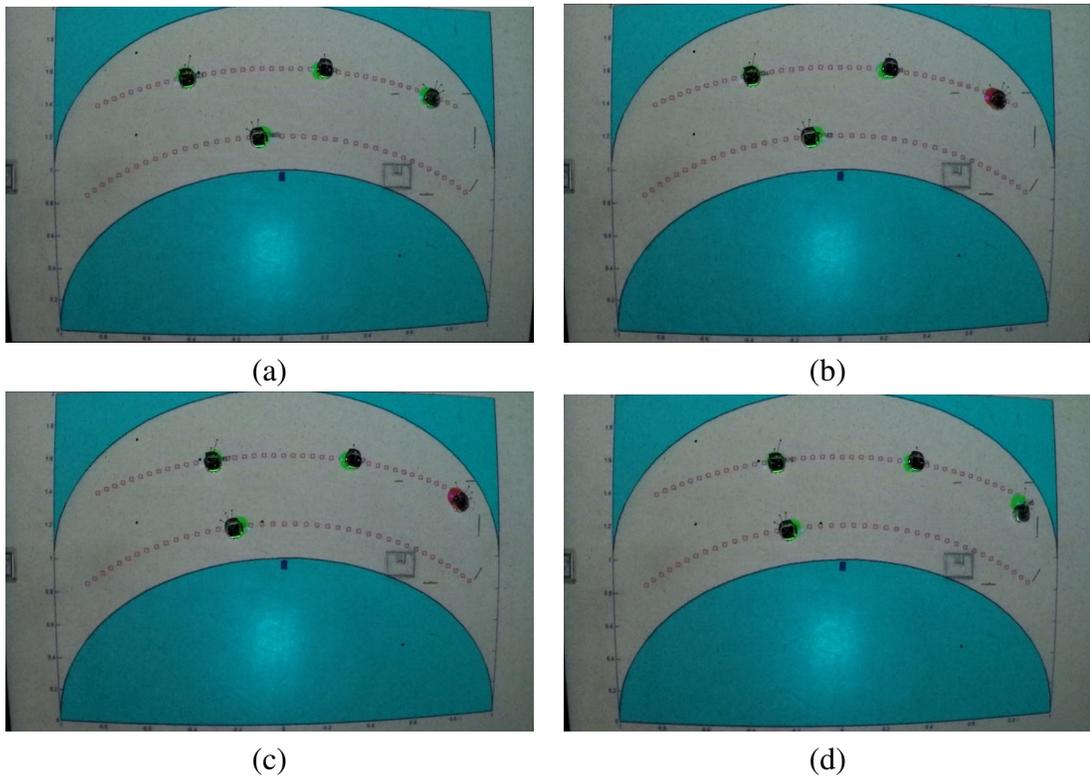


Figure 21: The robot on the right experiences a short duration communication failure. The safe time algorithm prevents jerky motion and allows the robot to continue moving.

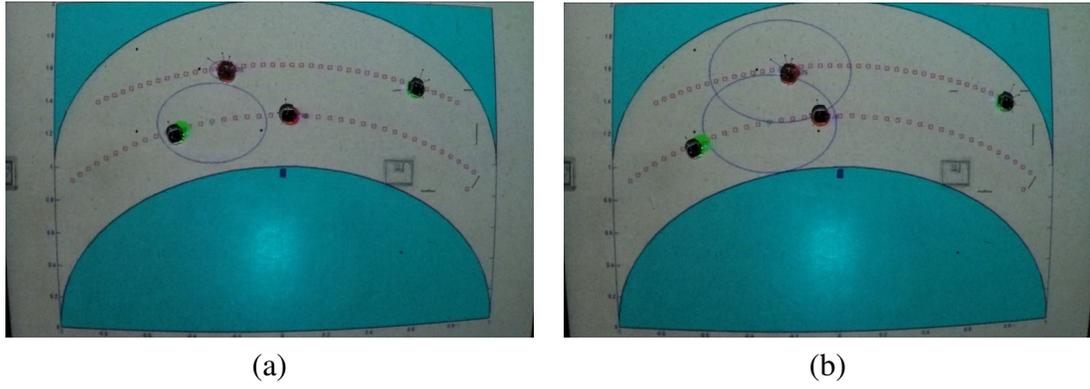


Figure 22: Multiple simultaneous communication failures are handled effectively by the safe time algorithm.

## CHAPTER 6

### CONCLUSIONS

In this thesis, we have developed an algorithm that allows robots in a multi-robot system to execute safe open-loop motion patterns while facing intermittent communications. To do this, we compute safe time horizons for each robot, which is determined by performing a reachability analysis of all the robots. This algorithm has been shown to prevent undesirable behaviors such as jerky and unpredictable movements in robots which are receiving intermittent motion commands. One major advantage of this algorithm is that it can be implemented in real-time. We have shown that for large teams of robots, the computational complexity tends to scale linearly with the number of robots. We also provide a design parameter which can be manipulated to make a trade-off between computational time and the maximum safe time durations. Furthermore, the safe time algorithm is agnostic to the control algorithm being executed by the robots and hence can be implemented on a wide range of multi-robot systems.

A notable contribution made in this thesis, is the derivation of continuous time analytical expressions for the minimum area ellipses enclosing the reachable sets of unicycle robots. The reachable sets of unicycle robots are non-convex and do not admit finite representation. Hence, our result has a major impact on the efficiency of the safe time algorithm and in-fact, can be used in any study/algorithm to obtain simple and efficient approximations of the reachable sets. To this end, it has been proved that the dissimilarity between the ellipsoidal approximation and the reachable set asymptotically converges to zero as time grows larger.

The safe time algorithm was implemented on a team of two-wheeled differential drive robots operating on a multi-robot testbed. It was shown that the algorithm prevented jerky motion behaviors, and resulted in safe trajectories for the robots.

## APPENDIX A

### STATE TRANSFORMATIONS: UNICYCLE ROBOTS

The unicycle dynamics represented in Equation 3 is repeated below.

$$\frac{dx}{dt} = v \cos(\phi)$$

$$\frac{dy}{dt} = v \sin(\phi)$$

$$\frac{d\phi}{dt} = \omega$$

$$|v| \leq v_m, |\omega| \leq \omega_m$$

To convert this to the dynamics shown in Equation 8, we transform the time-scale and the state space using the following equations:

$$x' = x \frac{\omega_m}{v_m}$$

$$y' = y \frac{\omega_m}{v_m}$$

$$\phi' = \phi$$

$$\hat{t} = \omega_m t$$

Then,

$$\frac{dx'}{d\hat{t}} = v' \cos(\phi')$$

$$\frac{dy'}{d\hat{t}} = v' \sin(\phi')$$

$$\frac{d\phi'}{d\hat{t}} = \omega'$$

$$|v'| \leq 1, |\omega'| \leq 1$$

Thus, we can obtain normalized representations for any unicycle robot.

## APPENDIX B

### CIRCULAR TRAJECTORY COMPUTATION

$Z_f(\mu, z_i(t_k))$  denotes the solution to the differential equations specified in Equation 3 for the constant velocities  $v_i(t_k)$  and  $\omega_i(t_k)$ . This can be written as,

$$\begin{aligned} x_i(t_k + \mu) &= x_i(t_k) + v_i(t_k) \int_0^\mu \cos(\phi_i(t)) dt \\ y_i(t_k + \mu) &= y_i(t_k) + v_i(t_k) \int_0^\mu \sin(\phi_i(t)) dt \\ \phi_i(t_k + \mu) &= \phi_i(t_k) + \mu\omega_i(t_k) \end{aligned}$$

Solving the integral we get the following two equations,

$$x_i(t_k + \mu) = x_i(t_k) + \frac{v_i(t_k)}{\omega_i(t_k)} [\sin(\omega_i(t_k)\mu + \phi_i(t_k)) - \sin(\phi_i(t_k))] \quad (50)$$

$$y_i(t_k + \mu) = y_i(t_k) - \frac{v_i(t_k)}{\omega_i(t_k)} [\cos(\omega_i(t_k)\mu + \phi_i(t_k)) - \cos(\phi_i(t_k))] \quad (51)$$

$$Z_f(\mu, z_i(t_k)) = \begin{cases} z_i(t_k) + \frac{v_i(t_k)}{\omega_i(t_k)} \begin{pmatrix} \sin(\omega_i(t_k)\mu + \phi_i(t_k)) - \sin(\phi_i(t_k)) \\ \cos(\phi_i(t_k)) - \cos(\omega_i(t_k)\mu + \phi_i(t_k)) \end{pmatrix}, & \text{if } \omega_i(t_k) \neq 0 \\ z_i(t_k) + v_i(t_k) \begin{pmatrix} \cos(\phi_i(t_k)) \\ \sin(\phi_i(t_k)) \end{pmatrix}, & \text{if } \omega_i(t_k) = 0 \end{cases} \quad (52)$$

## REFERENCES

- [1] A. Fedotov, V. Patsko, and V. Turova, *Reachable Sets for Simple Models of Car Motion*. INTECH Open Access Publisher, 2011.
- [2] E. Klavins, “Communication complexity of multi-robot systems,” in *Algorithmic Foundations of Robotics V*, pp. 275–291, Springer, 2004.
- [3] M. Schwager, D. Rus, and J.-J. Slotine, “Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment,” *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 371–383, 2011.
- [4] T. Balch and R. C. Arkin, “Behavior-based formation control for multirobot teams,” *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.
- [5] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.
- [6] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, “Cooperative air and ground surveillance,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 16–25, 2006.
- [7] E. King, Y. Kuwata, M. Alighanbari, L. Bertuccelli, and J. How, “Coordination and control experiments on a multi-vehicle testbed,” in *American Control Conference, 2004. Proceedings of the 2004*, vol. 6, pp. 5315–5320, IEEE, 2004.
- [8] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The grasp multiple micro-uav testbed,” *Robotics & Automation Magazine, IEEE*, vol. 17, no. 3, pp. 56–65, 2010.
- [9] Z. Jin, S. Waydo, E. B. Wildanger, M. Lammers, H. Scholze, P. Foley, D. Held, and R. M. Murray, “Mvwt-ii: The second generation caltech multi-vehicle wireless testbed,” in *American Control Conference, 2004. Proceedings of the 2004*, vol. 6, pp. 5321–5326, IEEE, 2004.
- [10] D. Pickem, L. Wang, P. Glotfelter, Y. Diaz-Mercado, M. Mote, A. Ames, E. Feron, and M. Egerstedt, “Safe, remote-access swarm robotics research on the robotarium,” *arXiv preprint arXiv:1604.00640*, 2016.
- [11] S. I. Roumeliotis and M. J. Mataric, ““small-world” networks of mobile robots,” in *AAAI/IAAI*, p. 1093, 2000.
- [12] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, “Control barrier certificates for safe swarm behavior,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.
- [13] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

- [14] Y. Lu, Y. Zhong, and B. Bhargava, “Packet loss in mobile ad hoc networks,” 2003.
- [15] V. Paxson, “End-to-end internet packet dynamics,” in *ACM SIGCOMM Computer Communication Review*, vol. 27, pp. 139–152, ACM, 1997.
- [16] G. H. Hagn and T. I. Dayharsh, “Land-mobile radio communication channel occupancy, waiting time, and spectrum saturation,” *Electromagnetic Compatibility, IEEE Transactions on*, no. 3, pp. 281–284, 1977.
- [17] A. De Luca, G. Oriolo, and M. Vendittelli, “Control of wheeled mobile robots: An experimental overview,” in *Ramsete*, pp. 181–226, Springer, 2001.
- [18] L. E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [19] J. Liberty, *Programming C#: Building .NET Applications with C#*. ” O’Reilly Media, Inc.”, 2005.
- [20] J.-P. Laumond, S. Sekhavat, and F. Lamiroux, *Guidelines in nonholonomic motion planning for mobile robots*. Springer, 1998.
- [21] E. Cockayne and G. Hall, “Plane motion of a particle subject to curvature constraints,” *SIAM Journal on Control*, vol. 13, no. 1, pp. 197–220, 1975.
- [22] J. Reeds and L. Shepp, “Optimal paths for a car that goes both forwards and backwards,” *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [23] H. J. Sussmann and G. Tang, “Shortest paths for the reeds-shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control,” *Rutgers Center for Systems and Control Technical Report*, vol. 10, pp. 1–71, 1991.
- [24] P. Souères, J.-Y. Fourquet, and J.-P. Laumond, “Set of reachable positions for a car,” *Automatic Control, IEEE Transactions on*, vol. 39, no. 8, pp. 1626–1630, 1994.
- [25] F. John, “Extremum problems with inequalities as subsidiary conditions,” in *Traces and Emergence of Nonlinear Programming*, pp. 197–215, Springer, 2014.
- [26] L. Danzer, D. Laugwitz, and H. Lenz, “U via the l ö wnersche ellipsoid and its analogue among a eik ”o body inscribed ellipsoids,” *Archives mathematics*, vol. 8, no. 3, pp. 214–219, 1957.
- [27] V. Zaguskin, “Circumscribed and inscribed ellipsoids of extremal volume,” *Uspekhi Matematicheskikh Nauk*, vol. 13, no. 6, pp. 89–93, 1958.
- [28] L. G. Khachiyan, “Polynomial algorithms in linear programming,” *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 1, pp. 53–72, 1980.
- [29] P. Sun and R. M. Freund, “Computation of minimum-volume covering ellipsoids,” *Operations Research*, vol. 52, no. 5, pp. 690–706, 2004.

- [30] S. Damla Ahipasaoglu, P. Sun, and M. J. Todd, “Linear convergence of a modified frank–wolfe algorithm for computing minimum-volume enclosing ellipsoids,” *Optimisation Methods and Software*, vol. 23, no. 1, pp. 5–19, 2008.
- [31] E. A. Yildirim, “On the minimum volume covering ellipsoid of ellipsoids,” *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 621–641, 2006.
- [32] O. Güler and F. Gürtuna, “Symmetry of convex sets and its applications to the extremal ellipsoids of convex bodies,” *Optimization Methods and Software*, vol. 27, no. 4-5, pp. 735–759, 2012.
- [33] O. Maler, “Computing reachable sets: An introduction,” tech. rep., Technical report, 2008.
- [34] J. T. Klosowski, M. Held, J. S. Mitchell, H. Sowizral, and K. Zikan, “Efficient collision detection using bounding volume hierarchies of k-dops,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 4, no. 1, pp. 21–36, 1998.
- [35] P. M. Hubbard, “Collision detection for interactive graphics applications,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 1, no. 3, pp. 218–230, 1995.
- [36] D. Eberly, “Dynamic collision detection using oriented bounding boxes,” *Geometric Tools, Inc*, 2002.
- [37] K. A. Andersen and C. Bay, “A survey of algorithms for construction of optimal heterogeneous bounding volume hierarchies,” *Saatavilla pdf-muodossa <http://image.diku.dk/projects/media/christian.bay.kasper.andersen.B>*, vol. 6, 2006.
- [38] P. Jaccard, *Distribution de la Flore Alpine: dans le Bassin des dranses et dans quelques régions voisines*. Rouge, 1901.
- [39] L. Armijo, “Minimization of functions having lipschitz continuous first partial derivatives,” *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.
- [40] D. Pickem, M. Lee, and M. Egerstedt, “The gritsbot in its natural habitat-a multi-robot testbed,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 4062–4067, IEEE, 2015.
- [41] “K Team Corporation Website.” <http://www.k-team.com/>, 2016.