Error Analysis in Multi-Agent Control Systems

Farzad Salehisadaghiani

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

March 2012

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Farzad Salehisadaghiani

Entitled: Error Analysis in Multi-Agent Control Systems

and submitted in partial fulfilment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____________________ Dr. R. Raut, Chair

____________________ Dr. Y. M. Zhang, External Examiner

____________________ Dr. L. Rodrigues, Examiner

____________________ Dr. A. G. Aghdam, Supervisor

Approved by: ____________________

Dr. W. E. Lynch, Chair
Department of Electrical and Computer Engineering

________ 20 ____________________

Dr. Robin A. L. Drew
Dean, Faculty of Engineering and Computer Science

ABSTRACT

Error Analysis in Multi-Agent Control Systems

Farzad Salehisadaghiani

Any cooperative control scheme relies on some measurements which are often assumed to be exact to simplify the analysis. However, it is known that in practice all measured quantities are subject to error, which can deteriorate the overall performance of the network significantly. This work proposes a new measurement error analysis in the control of multi-agent systems. In particular, the connectivity preservation of multi-agent systems with state-dependent error in distance measurements is considered. It is assumed that upper bounds on the measurement error and its rate of change are available. A general class of distributed control strategies is then proposed for the distance-dependent connectivity preservation of the agents in the network. It is shown that if two neighboring agents are initially located in the connectivity range, they are guaranteed to remain connected at all times. Furthermore, the formation control problem for a team of single-integrator agents subject to distance measurement error is investigated using navigation functions. Collision, obstacle and boundary avoidance are important features of the proposed strategy. Conditions on the magnitude of the measurement error and its rate of change are derived under which a new error-dependent formation can be achieved anywhere in the space. The effectiveness of the proposed control strategies in consensus and containment problems is demonstrated by simulation.

“ The more I learn, the more I learn how little I know. ” –Socrates

To my parents,

for their tireless support throughout my life

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Amir G. Aghdam for his sincere encouragement, guidance, and support during my Master's studies.

I would also like to thank Mr. Amir Ajorlou, who helped me all the time. His novel ideas and suggestions were essential for the results of this thesis; and it was a great honor for me to work with him.

I truly appreciate my family and those people who have enriched me as a professional and as a human being, so I sincerely thank them.

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), under the Discovery Grants Program, that is really appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

UAV	Unmanned Aerial Vehicle
NF	Navigation Function
DNF	Distributed Navigation Function
GPS	Global Positioning System
AHS	Automated Highway System
SR	Smart Road
ATC	Air Traffic Control
AFF	Autonomous Formation-Flying
RF	Radio Frequency

Chapter 1

Introduction

1.1 Motivation and Related Work

There has been increasing interest in multi-agent control systems in the past decade due to their wide range of applications. Some examples of real-world multi-agent systems include sensor networks, air traffic control, and automated highway systems, to name only a few [1–5]. In this type of problem, it is desired to achieve a global objective such as consensus, containment and formation by developing distributed control laws [6–11]. For instance, in the consensus problem, one or more state variables of all agents in the network are desired to come to a global agreement [12–15]. In the containment problem, it is desired that the followers converge to the convex hull of the leaders [16–19]. In the formation problem, on the other hand, the agents are to converge to a desired configuration in the workspace, which is defined in terms of the relative position of the agents [20, 21].

One of the common goals of any multi-agent control problem is the connectivity of the network. A pair of agents is said to be connected (via a communication link) if they are located in a sufficiently small distance from each other. The distributed connectivity preservation problem has been thoroughly investigated in prior literature [1, 22–31]. Unbounded potential functions are often used in these papers, where an unbounded potential field is generated between any two agents which tend to move away from the connectivity range. However, such approaches may not be as effective in practice because actuators cannot handle infinite control signal. To remedy this shortcoming, a general class of bounded distributed potential functions with the connectivity preserving property is proposed in [32]. The idea behind this technique is to design the potential functions in such a way that when an edge is about to lose connectivity, the gradient of the potential function lies in the direction of the edge, aiming to reduce its size. The work [32] proposes an effective alternative to conventional unbounded potential functions. However, this work and all of the papers cited above assume that the distance measurement (which is required in any connectivity control law) has no error. It is known that all measurements are subject to error in practice, and this can negatively affect the control performance. Later on in this thesis, we investigate the problem of connectivity preservation for a team of single-integrator agents helping the potential based functions subject to distance measurement error.

Other common goal of any multi-agent control problem which is tackled in this thesis is the formation control. To this end, potential functions are not always useful, so that many potential functions can be determined to be failed in reaching the goal. The most advantageous potential function are the ones including some suitable conditions under which, the desired states are reached with any initial state [33]. A potential function that satisfies these conditions is called a navigation function.

Navigation functions are known to be effective tools in the design of cooperative control schemes for multi-agent systems [34–36]. A class of triangulated graphs for algebraic representation of rigid formations is introduced in [37] to specify a mission cost for a group of vehicles. In [38–40] the formation behaviors are integrated with other navigational behaviors to enable a multi-agent network to reach the navigational goals. In [41], a novel distributed control scheme is designed and implemented to achieve dynamic formation control and collision avoidance for a group of non-holonomic agents. The corresponding approach is Lyapunov-based, and guarantees collision avoidance. In [42], formation control of unmanned aerial vehicles (UAVs) flying in an obstacle-laden environment is investigated. When static obstacles pop up during the operation, the UAVs are required to steer around them and also avoid collisions between each other. Three types of collision have been defined in [43] that should be avoided: 1) collision among the agents which are flying within communication range of each other; 2) collision of an agent with a fixed obstacle; 3) collision of an agent and the boundary of the workspace. The formation problem subject to collision and obstacle avoidance is studied in [44–49].

1.2 Thesis Objectives

Any cooperative objective such as the ones addressed in the previous section relies on some measurements (e.g. distance, speed, etc.) which are often assumed to be exact to simplify the analysis. However, it is known that in practice all measured quantities are subject to error, which can deteriorate the overall performance of the network significantly.

The first part of the thesis investigates the problem of connectivity preservation for a team of single-integrator agents subject to distance measurement error. It is assumed that upper bounds exist on the magnitude of the measurement error and its rate of change. These upper bounds are defined to ensure that the control signal will not be saturated. A general class of distributed control strategies is then proposed for the agents which has the connectivity preservation property. It is shown that if two neighboring agents are located within a more conservative distance from each other (compared to the case of the error-free measurements), they will remain in the connectivity range at all times. While the results of this work are presented for a static information flow graph, they can be easily extended to the more general case of dynamic edge addition.

The second part of the thesis tackles the formation control problem in the presence of measurement error. The objective is to design a controller under which the agents converge to any desired configuration (or a sufficiently small neighborhood of it) in the presence of measurement error, while they maintain a minimum distance from each other and from any fixed obstacle in the workspace (including the boundaries). To this end, a navigation-based distributed control law is designed which uses the information about the magnitude of the measurement error and its rate of change. It is shown that under some conditions the formation configuration in the presence of measurement error can be determined.

1.3 Thesis Outline and Publications

The remainder of this thesis is organized as follows. Chapter 2 provides the basic theoretical background, where some preliminaries on multi-agent control systems are presented. The problem of connectivity preservation of the multi-agent systems is described in chapter 3. In this chapter, some notations and definitions are given which will be used in the development of the main results. The connectivity preserving controller design is then elaborated in Chapter 3, which concludes with simulation results to demonstrate the effectiveness of the proposed control strategy. Distributed navigation functions for the formation control of a team of single-integrator agents with collision and obstacle avoidance in the presence of measurement error are introduced in Chapter 4. This chapter is also concluded with simulations which confirm the efficacy of the proposed navigation function-based control scheme in formation control of agents. Finally, conclusions and future research directions are presented in Chapter 5.

The results of this research appear in the following papers:

[1] **Farzad Salehisadaghiani**, Amir Ajorlou and Amir G. Aghdam, "Distributed Connectivity Preservation of a Team of Single-Integrator Agents Subject to Measurement Error," in *Proceedings of 2011 IEEE Multiconference on Systems & Control*, Denver, USA, pp. 626–631, Sep. 2011.

[2] **Farzad Salehisadaghiani**, Mehdi Asadi and Amir G. Aghdam, "Formation Control of a Team of Single-Integrator Agents with Collision and Obstacle Avoidance Schemes and Measurement Error," submitted for publication.

Chapter 2

Background

This chapter provides some background on multi-agent control systems. Some basic concepts are presented first, and examples of multi-agent systems are given. Then, the connectivity preservation and also formation control problems in multi-agent systems are formally defined, and some important results from the literature are presented.

2.1 Networked Systems

A great number of systems in our daily life can be modeled as a network of interconnected systems. Traffic networks, stock market, social networks and the Internet are examples of such systems. This type of system consists of a number of subsystems, each of which operates according to its dynamics, and also its interaction with other subsystems.

Multi-agent networks are a special class of interconnected systems, where each agent is a subsystem which is desired to be controlled with limited information from other agents. In the control literature, this is often referred to as *distributed control* structure. In the distributed control of multi-agent systems, it is desired to achieve an objective concerning the entire networked system by using a local controller for each agent. Every local controller uses the measure information of the agent associated with it as well as the measured information of a subset of agents, called neighboring agents, defined according to their sensing capabilities [50, 51] to construct the control signal for that agent.

In the sequel, some emerging applications of multi-agent control systems are presented.

2.1.1 A Network of Mobile Robots

A mobile robot network is often modeled as a group of autonomous agents capable of moving in a prespecified region and communicating with some other robots. This type of robot is sometimes used in mobile sensor networks where it is desired to increase sensing coverage in the region, or monitor a moving target [52–55]. Mobile sensor networks have applications in intrusion detection, surveillance, and environmental monitoring. Each mobile robot is equipped with an energy source (typically a battery) as well as sensing and communicating devices. It is important to employ an effective deployment strategy to improve the performance of the network (e.g. coverage and tracking), and an efficient resource management algorithm to minimize the power consumption of the batteries, and hence increase the lifetime of the network [56]. Some other practical problems in a network of mobile robots include collision avoidance and obstacle avoidance [57, 58].

2.1.2 Formation Flight of Unmanned Aerial Vehicles (UAV)

A UAV is an autonomous flying vehicle which is used in various civilian and military missions to reduce the risk of pilot casualties in a hostile environment. They have been recently used in combat operations, reconnaissance and border patrolling. In many of these applications, it is more desirable to perform the operation using a group of UAVs in a cooperative fashion. One of the important issues in this type of mission is that the UAVs are properly positioned and aligned with respect to each other, such that they form a certain configuration while they are flying. This is called formation flight, and has been extensively studied in the recent literature [59, 60]. In the problem of formation flying of UAVs, one should model the dynamics of the UAVs with a sufficient accuracy (single integrator, double integrator, unicycle, etc.) [61–63]. A controller is then designed for each vehicle which uses the measured distance of neighboring UAVs to generate the control signal for that vehicle. The type of sensor used to measure the distance or velocity in a formation flying control system depends on the specific application. For example, while autonomous formation-flying (AFF) sensors are more popular in satellite formation, radio frequency (RF) sensors are more desirable in terrain mapping [64–66]. The performance of the overall system highly depends on the accuracy of the measurements, communication topology, and control strategy.

2.1.3 Automated Highway Systems

The objective of an automated highway system (AHS) or smart road (SR) is to increase the capacity of the roads and highways. This is achieved by grouping the vehicles in different platoons [67, 68]. The speeds of all vehicle in each platoon is reduced or increased simultaneously. This in turn helps the platoon move synchronously in such a way that the waiting time of the vehicles at the traffic lights is reduced or eliminated completely. The cars in the platoon can move sufficiently close to each other using a proper control strategy, which can also help reduce traffic congestion.

In order to implement an AHS, a magnetic marker sensing system is deployed in the roadway to measure speed, and the radar-based forward collision warning technology is often used [69, 70]. The desired speed of the vehicles is computed at the decision making unit, and is communicated from the road to the vehicles [71, 72].

2.1.4 Air Traffic Control

Air traffic control (ATC) is a service provided to the aircraft (on the ground and in the air) to keep them away from the obstacles and hence prevent collision. The required information is typically provided by a ground-based control unit to properly coordinate the flow of traffic [73]. While the commands are issued based on a prescribed priority scheme, the pilot in command has the ultimate responsibility to avoid obstacles and other air traffic for the safety of the flight.

2.2 Cooperative Control

In the cooperative control problem, it is desired to perform the control action using the measurements obtained from the interacting agents based on an information exchange structure which can vary with time [74]. The objective is to coordinate the agents using a proper distributed control strategy with partial information from other agents [75]. Some important objectives in cooperative control of multi-agent systems include consensus, containment, formation, connectivity preservation, etc. Some of these objectives are described below:

- Consensus: All agents in the network come to a global agreement (in terms of a specific state variable such as position).
- Assignments: The assignment of tasks among multiple agents is decided according to the capability of the agents and their positions.
- Coverage: The agents are desired to be positioned properly in the region such that the coverage holes in the network are minimized.
- Flocking/Swarming: This is an essential behavior for a group of mobile agents, where they do not converge to a point but do not break up into smaller groups either. This complex group behavior requires coordinated decision making for agents, and can be observed in schools of fish or flock of birds.

2.2.1 Local Interaction Protocol

Graph-theoretic approaches are known to be very effective in the development of a cooperative control strategy [76]. Each agent in the network can be denoted by a vertex in the graph, and the communication between the agents can be represented by edges. In practice, the communication links in a multi-agent system are subject to noise and packet loss. However, in the analysis and design of this type of control structure, it is often assumed that the communication between neighboring agents is precise to simplify the stability analysis and controller design. In order to design the local controllers, it is important to have a sufficiently accurate model for the dynamics of each agent.

2.2.2 Connectivity Preservation

Network connectivity is an important issues in the cooperative control of multi-agent systems. Two agents are said to be in the connectivity range if the distance between them is less than a certain value, so that they can communicate with each other. Connectivity preservation, on the other hand, means that if two agents are initially at the connectivity range, they will never lose connectivity [22, 77, 78]. The distance between the agents needs to be measured, and a proper control action is needed to ensure that this distance remains smaller than the threshold value. In practice, the measurements are erroneous and the controller should preserve connectivity in the worst case scenario (maximum error magnitude).

Two of the well-known approaches to connectivity preservation are provided in the sequel.

2.2.2.1 Graph Laplacian Approach

One of the earliest approaches to connectivity preservation of multi-agent systems was based on graph Laplacian [24]. It is known that the connectivity of the network is closely related to the eigenvalues of graph Laplacian. In particular, it is well-known that maximizing the second smallest eigenvalue of Laplacian graph is very effective in connectivity preservation of the network [23]. Several techniques are proposed in the literature for maximizing the second smallest eigenvalue of graph Laplacian [79, 80].

2.2.2.2 Potential Functions

Potential functions are used to design control laws for multi-agent systems in order to achieve some objectives [81]. The negative gradient of a potential function is attractive to the desired destination and repulsive otherwise. Therefore, the negative gradient of the potential function is used as a feedback control signal to navigate the agents toward the desired destination while avoiding the obstacles. One of such functions is *nearest neighbor potential function*, where each agent updates its heading based on the average of its own heading plus the heading of its neighbors. Under this strategy, all neighbors update their heading in such a way that they remain optimally close to their local neighbors. Given the initial connectivity of the network, this implies that the network remains connected at all times.

The main drawback of this approach is that potential functions tend to infinity when the corresponding agents are about to lose connectivity [25–27]. However, in practice actuators cannot handle a signal whose magnitude is above a certain threshold. As a remedy to this problem, a class of bounded connectivity preserving control laws is proposed in [32].

Remark 2.1 *One of the commonly used structures in multi-agent control systems is the leader-follower architecture [82]. The leaders are those agents which receive external commands; the followers follow the leaders. To preserve connectivity in this type of structure, it is important to have a certain leader-to-follower ratio in the network so that by properly driving the leaders, the followers can move to any arbitrary positions [83].*

2.3 Measurement Error

Any feedback control system relies on output measurement which is used in the feedback loop as a representative of the output. In physical systems measurements are always subject to error. The accuracy of a sensor depends on the signal being measured, and the quality of the sensor. In design of a reliable control system, it is important to take measurement error into consideration in such a way that its effect in the closed-loop performance is minimized. Three main types of measurement error are described below.

2.3.1 Bias Error

When the output signal is not zero when the measured quantity is zero, the sensor is said to have *bias error* (also called offset error). The expected value of a measurement made by a sensor subject to bias error can differ significantly from the actual mean value of the measured quantity. This type of error is mainly caused by imperfect calibration of the sensor.

2.3.2 Drift Error

Sometimes the error in the measured value gradually changes even when the quantity is fixed. This type of error is called *drift error*. The effect of drift is more visible when the same fixed quantity is measured repeatedly. If each time the magnitude of the quantity is greater than the previously measured value, it is likely due to the drift error. Usually the drift error is caused by gradual changes in the environment such as temperature.

2.3.3 Random Error

Sometimes the measured output consists of rapidly changing signals with small magnitude. This is resulted by sensor noise, whose effect on the system output can be significantly attenuated by using a proper filter (such as Kalman filter). Statistical techniques can be used to analyze the effect of noise in the system.

Chapter 3

Connectivity Preservation of a Multi-agent System Subject to Measurement Error

In this chapter, the connectivity preservation of multi-agent systems with state-dependent error in distance measurement is investigated. A general class of distributed control strategies is proposed for the distance-dependent connectivity preservation of the agents in the network. The effectiveness of the proposed control strategies in consensus and containment problems is demonstrated by simulation.

3.1 System Model

The following definitions are borrowed from [32], and will prove convenient in presenting the main results.

Definition 3.1 *[32] For a real or vector valued function $f(t)$, the index of $f(t)$ at time t, denoted by $\rho(f(t))$, is defined to be the smallest natural number n for which $f^{(n)}(t) \neq 0$, where $f^{(n)}(t)$ is the n-th derivative of $f(t)$ with respect to time.*

Definition 3.2 *The function f is said to be of class C^k if the derivatives $f^{(1)}, \dots, f^{(k)}$ exist and are continuous. A function of class C^∞ is referred to as a smooth function.*

Definition 3.3 *[32] Multinomial coefficients are defined as:*

$$\begin{pmatrix} k \\ r_1, r_2, \dots, r_\mu \end{pmatrix} := \frac{k!}{r_1! r_2! \dots r_\mu!}$$

where $r_1, r_2, \dots, r_\mu$ are nonnegative integers, and $k = r_1 + r_2 + \dots + r_\mu$. In the special case when $\mu = 2$, these cofficients are called bionomial coefficients, and are given by $\begin{pmatrix} k \\ r_1, r_2 \end{pmatrix} = \begin{pmatrix} k \\ r_1 \end{pmatrix} = \begin{pmatrix} k \\ r_2 \end{pmatrix}$.

Notation 1 *For any given function $h(x,y)$, the derivative $\frac{\partial h}{\partial y} h(x,y)_{|y=0}$ is represented by $\frac{\partial h}{\partial y} h(x,0)$ (and similarly, $\frac{\partial h}{\partial x} h(0,y) = \frac{\partial h}{\partial x} h(x,y)_{|x=0}$).*

Notice that while this may be considered standard notation, it is emphasized here for the sake of clarity, and to avoid possible confusion.

Consider a set of n single-integrator agents in a plane with the control law of the form:

$$\dot{q}_i(t) = u_i(t) = -\frac{\partial h_i}{\partial q_i} \tag{3.1}$$

where $q_i(t)$ denotes the position of agent i in the plane at time t, and h_i's are distributed potential functions. Denote with $G = (V, E)$ the information flow graph, with $V = \{1, \ldots, n\}$ the vertices, and with $E \subset V \times V$ the edges. It is assumed that the information flow graph G is connected and undirected.

Definition 3.4 *The set of neighbors of agent i, denoted by $N_i(G)$, is a set consisting of any vertex in G which is connected to vertex i by an edge, i.e., $N_i := \{j | (i, j) \in E\}$. Moreover, the degree of the set of neighbors N_i is denoted by $d_i(G)$.*

Assume that each agent can only use the relative position of its neighbors in its local control law. Let the error function for distance measurement between any two agents i and j be denoted by $\varepsilon_{ij}(\|q_i - q_j\|)$, which is a smooth positive scalar function of distance ($\|\cdot\|$ denotes the Euclidean norm) and it occurs when agent i is sensing the position of agent j. The error function is assumed to be bounded with a known bound m, as follows:

$$\mid \varepsilon_{ij}(\|q_i - q_j\|) \mid \leq m$$

Furthermore, the rate of change of the error function is assumed to be bounded by 1, i.e.:

$$\left|\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|}\right| < 1$$

The main reason for the above assumption is to ensure that the variation of error is sufficiently slow.

Assume that L is a real distance between the two agents i and j, but the distance $L - \varepsilon_{ij}(\|q_i - q_j\|)$ is measured instead. In the error-free connectivity preservation problem (i.e., perfect measurements) two agents i and j are said to be in the connectivity range if $\|q_i - q_j\| \leq d$, where d is a pre-specified positive real number referred to as the *critical distance* [32]. However, in a practical setting, a more conservative condition needs to be adopted in order to ensure that connectivity is preserved in the presence of measurement error. More precisely, if the distance d is measured by a sensor, normally implying that the corresponding agent is in the connectivity range, the real distance can be as great as $d + m$, resulting in loss of connectivity. So, the critical distance in this case is adjusted to $d - m + \varepsilon_{ij}$, where $m < d$ (note that the distance $d - m + \varepsilon_{ij}$ is always less than or equal to d). The objective is to design a class of distributed potential function which preserve connectivity in this case. More precisely, it is desired to derive conditions under which if $\|q_i(0) - q_j(0)\| \leq d - m$ for all $(i, j) \in E$, then $\|q_i(t) - q_j(t)\| \leq d$ for all $(i, j) \in E$ and all $t \geq 0$.

3.2 Connectivity Preserving Controller Design

For every agent *i*, define:

$$\sigma_i(t) \;:=\; \frac{1}{2}\sum_{j\in N_i(G)} (\|q_i(t)-q_j(t)\|-\varepsilon_{ij})^2 \tag{3.2}$$

$$\pi_i(t) \;:=\; \frac{1}{2}\prod_{j\in N_i(G)} ((d-m)^2-(\|q_i(t)-q_j(t)\|-\varepsilon_{ij})^2) \tag{3.3}$$

$$\pi_{ij}(t) \;:=\; \prod_{\substack{k\in N_i(G)\\ k\neq j}} ((d-m)^2-(\|q_i(t)-q_k(t)\|-\varepsilon_{ik})^2) \tag{3.4}$$

Consider a set of distributed smooth potential functions of the form $h_i(\sigma_i,\pi_i)$ with the following properties:

$$\frac{\partial h_i}{\partial \sigma_i}(\sigma_i,0)=0,\ \frac{\partial h_i}{\partial \pi_i}(\sigma_i,0)<0, \qquad \text{for } \sigma_i\in\mathbb{R}^+ \tag{3.5}$$

These are the same potential function used in [32] for designing connectivity preserving control laws in the ideal (error-free) case. The aim of this section is to show that using this type of potential function and under some conditions, the control law (3.1) is connectivity preserving. Using the formula $\frac{\partial h_i}{\partial q_i}=\frac{\partial h_i}{\partial \sigma_i}\frac{\partial \sigma_i}{\partial q_i}+\frac{\partial h_i}{\partial \pi_i}\frac{\partial \pi_i}{\partial q_i}$, one can rewrite the control law (3.1) as:

$$\begin{aligned}\dot{q}_i \;=\; &-\sum_{j\in N_i(G)} (((q_i-q_j)+\frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2})-\frac{\partial}{\partial q_i}(\varepsilon_{ij}\|q_i-q_j\|))\\ &(\frac{\partial h_i}{\partial \sigma_i}-\frac{\partial h_i}{\partial \pi_i}\pi_{ij}))\end{aligned} \tag{3.6}$$

Let $t_0 = \inf\{t | \exists (i,j) \in E : \|q_i - q_j\| > d - m + \varepsilon_{ij}\}$. Clearly, $\|q_i(t) - \ q_j(t)\| \leq d - m + \varepsilon_{ij}$, for all $(i,j) \in E$ and $t \leq t_0$. Construct a graph $G_d = (V_d, E_d)$ as the union of those edges $(i,j) \in E$ for which $\|q_i(t_0) - q_j(t_0)\| = d - m + \varepsilon_{ij}$. Define $s_{ij}(t) = (\|q_i(t) - q_j(t)\| - \varepsilon_{ij})^2$, for all $(i,j) \in E_d$. To prove the above claim, it suffices to show that there is a neighborhood of t_0 in which every s_{ij} is either decreasing or fixed.

The following lemmas will be used in the proof of the main results.

Lemma 3.1 *Consider a real or vector valued function f of $x(t)$ in C^k. If $x^{(r)}(t_0) = 0$ for all $r \in \{1, 2, \ldots, k\}$, then:*

$$f^{(r)}(x(t_0)) = 0, \forall r \in \{1, 2, \ldots, k\}$$

Proof. Since $f(x) \in C^k$, thus $\frac{\partial^r}{\partial x^r} f(x(t_0))$ exists for all $r \in \{1, 2, \ldots, k\}$. On the other hand, $\dot{f}(x(t)) = \frac{\partial f(x)}{\partial x} \dot{x}(t)$. The proof follows directly from the fact that:

$$(xy)^{(k)} = \sum_{r=0}^{k} x^{(r)} y^{(k-r)} \begin{pmatrix} k \\ r \end{pmatrix} \tag{3.7}$$

which implies $f^{(r)}(x(t)) = \sum_{n=0}^{r-1} (\frac{\partial f(x)}{\partial x})^{(n)} x^{(r-n)}(t) \begin{pmatrix} r-1 \\ m \end{pmatrix}$. Note that since $x^{(r)}(t_0) = 0$ for all $r \in \{1, 2, \ldots, k\}$, thus $f^{(r)}(x(t_0)) = 0$ for all $r \in \{1, 2, \ldots, k\}$. ■

Lemma 3.2 *Suppose that $q_i^{(r)}(t) = q_j^{(r)}(t) = 0$, for all $r \in \{1,\dots,k-1\}$ and some t. Then:*

$$
\begin{aligned}
&(\varepsilon_{ij}^2 - 2\varepsilon_{ij}\|q_i - q_j\|)^{(k)} = (2\frac{\varepsilon_{ij}}{\|q_i - q_j\|}\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - \\
&2\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - 2\frac{\varepsilon_{ij}}{\|q_i - q_j\|})(q_i(t) - q_j(t))^T \\
&(q_i^{(k)}(t) - q_j^{(k)}(t))
\end{aligned}
\tag{3.8}
$$

Proof. It is required to take the derivative of the error terms iteratively, in order to find the k-th derivative. This can be carried out as follows:

$$
\begin{aligned}
&\frac{d}{dt}(\varepsilon_{ij}^2 - 2\varepsilon_{ij}\|q_i - q_j\|) = \dot{\varepsilon}_{ij}(\varepsilon_{ij} - 2\|q_i - q_j\|) + \varepsilon_{ij} \\
&\frac{d}{dt}(\varepsilon_{ij} - 2\|q_i - q_j\|) = (2\frac{\varepsilon_{ij}}{\|q_i - q_j\|}\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - 2 \\
&\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - 2\frac{\varepsilon_{ij}}{\|q_i - q_j\|})(q_i(t) - q_j(t))^T(\dot{q}_i(t) - \dot{q}_j(t))
\end{aligned}
$$

Since $q_i^{(r)}(t) = q_j^{(r)}(t) = 0$ for all $r \in \{1,\dots,k-1\}$, hence:

$$
\begin{aligned}
&(\varepsilon_{ij}^2 - 2\varepsilon_{ij}\|q_i - q_j\|)^{(k)} = (2\frac{\varepsilon_{ij}}{\|q_i - q_j\|}\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} \\
&-2\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - 2\frac{\varepsilon_{ij}}{\|q_i - q_j\|})(q_i(t) - q_j(t))^T \\
&(q_i^{(k)}(t) - q_j^{(k)}(t))
\end{aligned}
$$

■

Lemma 3.3 *Consider agents i and j in G_d, where j is the neighbor for which $\|q_i - q_j\| = d - m + \varepsilon_{ij}$. Suppose that $q_i^{(r)}(t) = q_j^{(r)}(t) = 0$ for all $r \in \{1, \ldots, k-1\}$ and some t. Then:*

$$s_{ij}^{(k)}(t) = A(q_i(t) - q_j(t))^T (q_i^{(k)}(t) - q_j^{(k)}(t)) \tag{3.9}$$

where $A > 0$.

Proof. According to Lemma 3.2 and by using (3.7), one can write:

$$\begin{aligned} s_{ij}^{(k)}(t) \quad = \quad & (2 + 2\frac{\varepsilon_{ij}}{\|q_i - q_j\|} \frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - 2\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - \\ & 2\frac{\varepsilon_{ij}}{\|q_i - q_j\|})(q_i(t) - q_j(t))^T (q_i^{(k)}(t) - q_j^{(k)}(t)) \end{aligned}$$

To prove this lemma, it suffices to show that:

$$\begin{aligned} & 2 + \frac{2\varepsilon_{ij}}{\|q_i - q_j\|} \frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - \frac{2\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - \frac{2\varepsilon_{ij}}{\|q_i - q_j\|} > 0 \\ & 2(\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} - 1)(\frac{\varepsilon_{ij}}{\|q_i - q_j\|} - 1) > 0 \end{aligned}$$

The above inequality follows directly from $\left| \frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|} \right| < 1$ and $\mid \varepsilon_{ij} \mid < m < d$. This completes the proof. ■

Lemma 3.4 *Consider agents i and j in G_d, where j is the neighbor for which $\|q_i - q_j\| =$*

$d - m + \varepsilon_{ij}$. *Then:*

$$\|\frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \frac{\partial}{\partial q_i}(\varepsilon_{ij}\|q_i - q_j\|)\| < \|q_i - q_j\| \tag{3.10}$$

Proof. The left side of 3.10 can be expressed as:

$$\begin{aligned} &\|\varepsilon_{ij}\frac{\partial \varepsilon_{ij}}{\partial q_i} - \frac{\partial \varepsilon_{ij}}{\partial q_i}\|q_i - q_j\| - \varepsilon_{ij}\frac{\partial}{\partial q_i}\|q_i - q_j\|\| = \\ &\|\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|}\frac{(q_i - q_j)^T}{\|q_i - q_j\|}(\varepsilon_{ij} - \|q_i - q_j\|) - \\ &\varepsilon_{ij}\frac{(q_i - q_j)^T}{\|q_i - q_j\|}\| = \|\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|}(m - d) - \varepsilon_{ij}\| \end{aligned} \tag{3.11}$$

Now, using the inequality $m < d$, one arrives at:

$$\begin{aligned} &\|\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|}(m - d) - \varepsilon_{ij}\| < \\ &\|\frac{\partial \varepsilon_{ij}}{\partial \|q_i - q_j\|}\|(d - m) + \varepsilon_{ij} < d - m + \varepsilon_{ij} \end{aligned}$$

This completes the proof. ■

Lemma 3.5 *Consider agents i and j in G_d, where j is the neighbor for which* $\|q_i - q_j\| =$

$d-m+\varepsilon_{ij}$. Then:

$$(q_i-q_j)^T((q_i-q_j)+\frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2})-\frac{\partial}{\partial q_i}(\varepsilon_{ij}\|q_i-q_j\|))>0 \tag{3.12}$$

Proof. The left side of 3.12 can be rewritten as:

$$\|q_i-q_j\|^2+(q_i-q_j)^T(\frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2})-\frac{\partial}{\partial q_i}(\varepsilon_{ij}\|q_i-q_j\|))$$
$$\geq \|q_i-q_j\|^2-\|q_i-q_j\|.\|\frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2})-\frac{\partial}{\partial q_i}(\varepsilon_{ij}$$
$$\|q_i-q_j\|)\|=\|q_i-q_j\|(\|q_i-q_j\|-\|\frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2})-$$
$$\frac{\partial}{\partial q_i}(\varepsilon_{ij}\|q_i-q_j\|))\|$$

The rest of the proof follows directly from Lemma 3.4. ■

Lemma 3.6 *Consider an agent i in G_d and assume that $\eta=\min_{j\in N_i(G)}\{\rho(\pi_{ij})\}$. Assume also that $d_i(G_d)\geq 2$; then the following statements hold:*

i) $\pi_{ij}^{(r)}=0$, *for* $0\leq r\leq \eta-1$, *and* $j\in N_i(G)$.

ii) $\pi_i^{(r)}=0$, *for* $0\leq r\leq \eta-1$.

iii) $(\frac{\partial h_i}{\partial \sigma_i})^{(r)}=0$, *for* $0\leq r\leq \eta-1$.

iv) $\rho(q_i)\geq \eta+1$.

Proof. The proof is similar to that of Lemma 3 in [32]. ■

Remark 3.1 *Since the error function satisfies the conditions of Lemma 3.1, therefore* $\rho(\varepsilon_{ij}) \geq \rho(q_i)$ *and* $\rho(\varepsilon_{ij}) \geq \rho(q_j)$.

Remark 3.2 *In the case when* $d_i(G_d) = 1$*, it is straightforward to show that* $\dot{q}_i = ((q_i - q_j) + \frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \frac{\partial}{\partial q_i}(\varepsilon_{ij}.\|q_i - q_j\|))\frac{\partial h_i}{\partial \pi_i}\pi_{ij}$*, where agent* j *is the neighbor for which* $\|q_i - q_j\| = d - m + \varepsilon_{ij}$.

Remark 3.3 *If* $\rho(\pi_{ij})$ *is not the same for all* $j \in N_i(G_d)$*, then part (ii) of Lemma 3.6 also holds for* $r = \eta$*. Consequently, part (iii) also holds for* $r = \eta$.

Lemma 3.7 *For any agent* i *in* G_d*, let* ν *be one of the (possibly multiple) neighbors of* i *in* G_d *for which* $\rho(q_\nu) = \max_{j \in N_i(G_d)}\{\rho(q_j)\}$*. Then:*

$$\rho(q_i) \geq 1 + \sum_{\substack{j \in N_i(G_d) \\ j \neq \nu}} \rho(q_j) \tag{3.13}$$

Proof. For the case when $d_i(G_d) = 1$, it is implied from the definition of the index of a function that $\rho(q_i) \geq 1 + \sum_{\substack{j \in N_i(G_d) \\ j \neq \nu}} \rho(q_j)$. Now, consider the case where $d_i(G_d) \geq 2$; for any $j \in N_i(G)$, by differentiating (3.4) k times, one arrives at:

$$\begin{aligned} \pi_{ij}^{(k)} &= \sum_{\substack{r_1 + \ldots + r_\mu = k \\ r_1, \ldots, r_\mu \geq 0}} \begin{pmatrix} k \\ r_1, \ldots, r_\mu \end{pmatrix} \prod_{s=1}^{\mu} ((d-m)^2 - \\ &\quad (\|q_i - q_{i_s}\| - \varepsilon_{ij})^2)^{(r_s)} \end{aligned} \tag{3.14}$$

where $\{i_1,\dots,i_\mu\} = N_i(G) - \{j\}$. Let $k \le \eta$; using Remark 3.1 and on noting (from Lemma 3.6) that $\rho(q_i) > k$, one can easily verify that the term corresponding to $(r_1,...,r_\mu)$ in the above summation is nonzero only if $r_s \ge \rho(q_{i_s})$ for every $i_s \in N_i(G_d) - \{j\}$. On the other hand:

$$k = \sum_{s=1}^{\mu} r_s \ge \sum_{\substack{i_s \in N_i(G_d)\\ i_s \neq j}} r_s \tag{3.15}$$

Therefore, a necessary condition for $\pi_{ij}^{(k)}$ to be nonzero can be obtained as

$$k \ge \sum_{\substack{i_s \in N_i(G_d)\\ i_s \neq j}} \rho(q_{i_s}) \tag{3.16}$$

Now, choose $k = \eta$. Since $\eta = \min_{j \in N_i(G)}\{\rho(\pi_{ij})\}$, thus $\pi_{ij}^{(\eta)} \neq 0$ for at least one $j \in N_i(G)$. This implies that (3.16) should hold for $k = \eta$ and at least one $j \in N_i(G)$. Clearly, the right side of (3.16) is minimized for $j = v$ because this is when $\rho(q_j)$ is maximized. This fact along with part (iv) of Lemma 3.6 results in (3.13). ■

Lemma 3.8 *Let $\rho_l(q_i)$ be the lower bound for $\rho(q_i)$ given in Lemma 3.7, i.e.:*

$$\rho_l(q_i) = 1 + \sum_{\substack{j \in N_i(G_d)\\ j \neq v}} \rho(q_j) \tag{3.17}$$

where $\rho(q_v) = \max_{j \in N_i(G_d)}\{\rho(q_j)\}$. If v is unique, then:

i) $\pi_{iv}^{(\rho_l(q_i)-1)} = \tilde{\pi}_{iv} \prod_{\substack{j \in N_i(G_d)\\ j \neq v}} (q_i - q_j)^T q_j^{(\rho(q_j))}$, *where* $\tilde{\pi}_{iv} > 0$.

ii) $q_i^{(\rho_l(q_i))} = \frac{\partial h_i}{\partial \pi_i}\tilde{\pi}_{i\nu}((q_i - q_\nu) + \frac{\partial}{\partial q_i}(\frac{\varepsilon_{i\nu}^2}{2}) - \frac{\partial}{\partial q_i}(\varepsilon_{i\nu}\|q_i - q_\nu\|))\prod_{\substack{j\in N_i(G_d)\\ j\neq\nu}} (q_i - q_j)^T q_j^{(\rho(q_j))}$.

Proof. i) Let (3.14) be revisited for $k = \rho_l(q_i) - 1$. Hence, $\pi_{ij}^{(k)} = 0$ for $j \neq \nu$. Lemma 3.7 yields $k = \sum_{\substack{j\in N_i(G)\\ j\neq\nu}} r_s \geq \sum_{\substack{j\in N_i(G_d)\\ j\neq\nu}} r_s \geq \sum_{\substack{j\in N_i(G_d)\\ j\neq\nu}} \rho(q_j)$. On the other hand, $k = \sum_{\substack{j\in N_i(G_d)\\ j\neq\nu}} \rho(q_j)$ which leads to:

$$r_s = \begin{cases} \rho(q_j) & j \in N_i(G_d) - \{\nu\} \\ 0 & j \notin N_i(G_d) - \{\nu\} \end{cases} \tag{3.18}$$

The function $\pi_{i\nu}^{(k)}$ has only one possibly nonzero term as follows:

$$\begin{aligned} \pi_{i\nu}^{(k)} &= \binom{k}{r_1,\dots,r_\mu} \prod_{\substack{j\in N_i(G_d)\\ j\neq\nu}} ((d-m)^2 - (\|q_i - q_j\| - \\ &\quad \varepsilon_{ij})^2)^{(\rho q_j)} \times \prod_{\substack{j\in N_i(G)\\ j\notin N_i(G_d)}} ((d-m)^2 - (\|q_i - q_j\| \\ &\quad -\varepsilon_{ij})^2) \end{aligned} \tag{3.19}$$

Note that $\rho(q_i) > k$ and $k \geq \rho(q_j)$, which yields $q_i^{(\rho(q_j))} = 0$. Hence:

$$\begin{aligned} \pi_{i\nu}^{(k)} &= \binom{k}{r_1,\dots,r_\mu} \prod_{\substack{j\in N_i(G_d)\\ j\neq\nu}} -A(q_i - q_j)^T (q_i^{\rho(q_j)} \\ &\quad -q_j^{(\rho(q_j))}) \times \prod_{\substack{j\in N_i(G)\\ j\notin N_i(G_d)}} ((d-m)^2 - (\|q_i - q_j\| \\ &\quad -\varepsilon_{ij})^2) = \tilde{\pi}_{i\nu} \prod_{\substack{j\in N_i(G_d)\\ j\neq\nu}} (q_i - q_j)^T q_j^{(\rho(q_j))} \end{aligned} \tag{3.20}$$

Note also that $\prod_{\substack{j\in N_i(G)\\ j\notin N_i(G_d)}} ((d-m)^2-(\|q_i-q_j\|-\varepsilon_{ij})^2)>0$, which implies that $\tilde{\pi}_{iv}>0$.

ii) By differentiating both sides of (3.6) k times, one arrives at:

$$\begin{aligned} q_i^{(k+1)} = -\sum_{j\in N_i(G)}\sum_{r=0}^{k}((q_i-q_j)+\frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2})-\frac{\partial}{\partial q_i} \\ (\varepsilon_{ij}\|q_i-q_j\|))^{(r)}(\frac{\partial h_i}{\partial \sigma_i}-\frac{\partial h_i}{\partial \pi_i}\pi_{ij})^{(k-r)}\begin{pmatrix} k \\ r \end{pmatrix} \end{aligned} \tag{3.21}$$

Consider (3.21) for $k=\rho_l(q_i)-1$. Using the equality $\pi_{ij}^{(k)}=0$ (for $j\neq v$) along with Lemma 3.6 and Remark 3.3, one can conclude that $q_i^{(\rho_l(q_i))}$ has only one possibly nonzero term as:

$$\begin{aligned} q_i^{(\rho_l(q_i))} &= \frac{\partial h_i}{\partial \pi_i}\pi_{iv}^{((\rho_l(q_i))-1)}[(q_i-q_v)+\frac{\partial}{\partial q_i}(\frac{\varepsilon_{iv}^2}{2})- \\ &\quad \frac{\partial}{\partial q_i}(\varepsilon_{iv}\|q_i-q_v\|)] \\ &= \frac{\partial h_i}{\partial \pi_i}\tilde{\pi}_{iv}[(q_i-q_v)+\frac{\partial}{\partial q_i}(\frac{\varepsilon_{iv}^2}{2})-\frac{\partial}{\partial q_i} \\ &\quad (\varepsilon_{iv}\|q_i-q_v\|)]\prod_{\substack{j\in N_i(G_d)\\ j\neq v}}(q_i-q_j)^T q_j^{(\rho(q_j))} \end{aligned} \tag{3.22}$$

This completes the proof. ■

Lemma 3.9 *Consider a real or vector valued function f for which $f^{(\rho(f(t)))}(t)<0$, for some t; then f is monotonically decreasing in the interval $[t,t+\varepsilon]$, for some $\varepsilon>0$.*

Proof. The proof is similar to that of Lemma 1 in [32]. ■

Lemma 3.10 *Define the subgraph $\tilde{G}_d$ of G_d as the union of those edges $e = (i, j) \in E_d$ for which $\min(\rho(q_i), \rho(q_j)) < \infty$, and denote its set of edges with $\tilde{E}_d$. Then, for any $(i, j) \in \tilde{E}_d$, $\rho(s_{ij}) = \min\{\rho(q_i), \rho(q_j)\}$ and $s_{ij}^{(\rho(s_{ij}))} < 0$.*

Proof. One can prove this lemma by induction on $\min(\rho(q_i), \rho(q_j))$. Start with $\min(\rho(q_i), \rho(q_j)) = 1$, and without loss of generality assume that $\rho(q_i) = 1$. If $\rho(q_j) > 1$, then $\dot{q}_j = 0$, and hence from Lemma 3.3 and Remark 3.2:

$$\begin{aligned}
\dot{s}_{ij} &= A(q_i - q_j)^T(\dot{q}_i - \dot{q}_j) \\
&= A(q_i - q_j)^T \frac{\partial h_i}{\partial \pi_i} \pi_{ij}((q_i - q_j) + \frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \\
&\quad \frac{\partial}{\partial q_i}(\varepsilon_{ij}\|q_i - q_j\|))
\end{aligned} \tag{3.23}$$

According to Lemma 3.5 and on noting that $A > 0$, one can conclude that $\dot{s}_{ij} < 0$. Also, if $\rho(q_j) = \rho(q_i) = 1$, then:

$$\begin{aligned}
\dot{s}_{ij} &= A(q_i - q_j)^T(((q_i - q_j) + \frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \frac{\partial}{\partial q_i} \\
&\quad (\varepsilon_{ij}\|q_i - q_j\|))\frac{\partial h_i}{\partial \pi_i}\pi_{ij} - ((q_j - q_i) + \frac{\partial}{\partial q_j}(\frac{\varepsilon_{ji}^2}{2}) - \\
&\quad \frac{\partial}{\partial q_j}(\varepsilon_{ji}\|q_j - q_i\|))\frac{\partial h_j}{\partial \pi_j}\pi_{ji}) \\
&= A(q_i - q_j)^T \frac{\partial h_i}{\partial \pi_i}\pi_{ij}((q_i - q_j) + \frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \frac{\partial}{\partial q_i} \\
&\quad (\varepsilon_{ij}\|q_i - q_j\|)) + A(q_j - q_i)^T \frac{\partial h_j}{\partial \pi_j}\pi_{ji}((q_j - q_i) + \\
&\quad \frac{\partial}{\partial q_j}(\frac{\varepsilon_{ji}^2}{2}) - \frac{\partial}{\partial q_j}(\varepsilon_{ji}\|q_j - q_i\|))
\end{aligned} \tag{3.24}$$

which yields $\dot{s}_{ij} < 0$. Now, suppose that the lemma holds for $\min(\rho(q_i), \rho(q_j)) < k$. To prove the lemma for $\min(\rho(q_i), \rho(q_j)) = k$, assume without loss of generality that $\rho(q_i) = k$. Since $\rho(q_i) \leq \rho(q_j)$, using Lemma 3.7 one can easily show that $\max_{\omega \in N_i(G_d)}\{\rho(q_\omega)\}$ is unique, and in fact equals to q_j. As another consequence of Lemma 3.7, $\rho(q_\omega) < \rho(q_i)$ for $\omega \in N_i(G_d)$, $\omega \neq j$. Therefore, $\min(\rho(q_i), \rho(q_\omega)) = \rho(q_\omega) < k$, and hence $\rho(s_{i\omega}) = \rho(q_\omega)$ and $s_{i\omega}^{(\rho(s_{i\omega}))} < 0$. This along with Lemmas 3.3 and 3.8 results in:

$$\begin{aligned} q_i^{(\rho_l(q_i))} &= \frac{\partial h_i}{\partial \pi_i}\tilde{\pi}_{ij}((q_i - q_j) + \frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \frac{\partial}{\partial q_i} \\ &\quad (\varepsilon_{ij}\|q_i - q_j\|)) \prod_{\substack{\omega \in N_i(G_d) \\ \omega \neq j}} (q_i - q_\omega)^T q_\omega^{(\rho(q_\omega))} \\ &= \frac{\partial h_i}{\partial \pi_i}\tilde{\pi}_{ij}((q_i - q_j) + \frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \frac{\partial}{\partial q_i} \\ &\quad (\varepsilon_{ij}\|q_i - q_j\|)) \prod_{\substack{\omega \in N_i(G_d) \\ \omega \neq j}} -\frac{1}{A} s_{i\omega}^{(\rho(s_{i\omega}))} \end{aligned} \tag{3.25}$$

Thus:

$$\begin{aligned} (q_i - q_j)^T q_i^{(\rho_l(q_i))} &= (q_i - q_j)^T \frac{\partial h_i}{\partial \pi_i}\tilde{\pi}_{ij}((q_i - q_j) + \\ &\quad \frac{\partial}{\partial q_i}(\frac{\varepsilon_{ij}^2}{2}) - \frac{\partial}{\partial q_i}(\varepsilon_{ij}\|q_i - q_j\|)) \\ &\quad \prod_{\substack{\omega \in N_i(G_d) \\ \omega \neq j}} -\frac{1}{A} s_{i\omega}^{(\rho(s_{i\omega}))} < 0 \end{aligned} \tag{3.26}$$

from which one can conclude that $\rho(q_i) = \rho_l(q_i)$. On the other hand:

$$\begin{aligned} s_{ij}^{(\rho(q_i))} &= A(q_i - q_j)^T(q_i^{(\rho(q_i))} - q_j^{(\rho(q_i))}) \\ &= A(q_i - q_j)^T q_i^{(\rho(q_i))} + A(q_j - q_i)^T q_j^{(\rho(q_i))} \end{aligned} \tag{3.27}$$

If $\rho(q_j) > \rho(q_i)$, then the second term in the right side of 3.27 vanishes, and it follows from (3.26) that $s_{ij}^{(\rho(q_i))} < 0$. If $\rho(q_j) = \rho(q_i)$, the same inequality as (3.26) holds for $\rho(q_j)$. Therefore, both terms in (3.27) are less than zero, and hence $s_{ij}^{(\rho(q_i))} < 0$. ■

Lemma 3.11 *Consider the partition $E_d = E_\infty \cup \tilde{E}_d$. Then, for every $i \in V_\infty$,*

i) $d_i(G_\infty) \geq 2$.

ii) $\dot{q}_i(t) = 0$, *for* $t \geq t_0$.

Proof. The proof is similar to that of Lemma 7 in [32]. ■

Theorem 3.1 *Under the conditions given by (3.5), the control law (3.1) is connectivity preserving.*

Proof. To prove the theorem, it suffices to show that there is a neighborhood of t_0 in which for every $(i, j) \in E_d$, s_{ij} is either decreasing or fixed. It follows from Lemmas 3.9 and 3.10 that s_{ij} is decreasing in a neighborhood of t_0 for any $(i, j) \in \tilde{E}_d$. Also, from Lemma 3.11, s_{ij} is fixed for any $t \geq t_0$ and $(i, j) \in E_\infty$. The proof is completed on noting that $E_d = E_\infty \cup \tilde{E}_d$.■

Corollary 1 *For the case where the information flow graph G is a tree, the connectivity preservation is strict, meaning that if $\|q_i(0) - q_j(0)\| \leq d - m$ for all $(i, j) \in E$, then $\|q_i(t) - q_j(t)\| < d$, for all $(i, j) \in E$ and all $t > 0$.*

Proof. The proof is similar to that of Corollary 1 in [32]. ■

3.3 Simulation Results

Example 3.1 *Consider 4 single-integrator agents moving in a two-dimensional space with the control law given by (3.1) and the information flow graph G_1 depicted in Fig. 3.1. Let the potential function h_i ($i = 1, 2, \ldots, 4$) be chosen as:*

$$h_i(\sigma_i, \pi_i) = \frac{\sigma_i}{\sigma_i + \pi_i + {\pi_i}^2} \tag{3.28}$$

It can be shown that the above function satisfies the conditions in (3.5), and hence the control law obtained by using this function is connectivity-preserving. Assume also that the error function has the following form:

$$\varepsilon_{ij} = m(1 - e^{-\|q_i - q_j\|}) \tag{3.29}$$

One can verify that the maximum value of the above function is m, and that function has the maximum rate of change of this function with respect to the relative distance is less than 1.

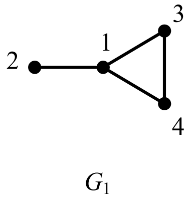

Figure 3.1: The information flow graph for the multi-agent system of Example 3.1.

Hence, this function satisfies all of the required conditions discussed earlier.

The control law (3.6) along with (3.5) implies that the velocity of each agent is directed towards a point inside the convex hull of its neighbors. This results in the contraction of the convex hull of the entire team, which in turn leads to the convergence of the agents to a single point. Let d and m be equal to $1m$ and $0.1m$, respectively. The planar motion of the agents for the initial positions marked by the indices of the agents is shown in Fig. 3.2. Denote the relative distance between agent i and its neighbor j with d_{ij}, i.e., $d_{ij} = \|q_i - q_j\|$. The relative distances d_{12}, d_{13}, and d_{34} are depicted in Fig. 3.3, which confirms the convergence to consensus. Furthermore, the norm of the control inputs u_1, u_2 and u_3 are drawn in Fig. 3.4, which shows they are bounded at all times, as expected.

Example 3.2 *Consider now a team of 3 static leaders and 3 followers with the information flow graph G_2 depicted in Fig. 3.5. The followers are desired to converge to the triangle of the leaders while preserving the connectivity of the information flow graph. Consider the following potential function:*

$$h_i(\sigma_i, \pi_i) = -\pi_i \tag{3.30}$$

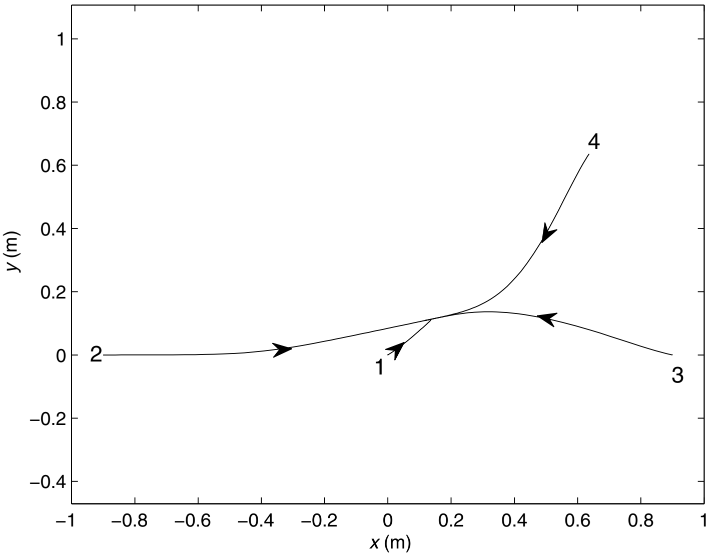

Figure 3.2: The agents' planar motion in Example 3.1.

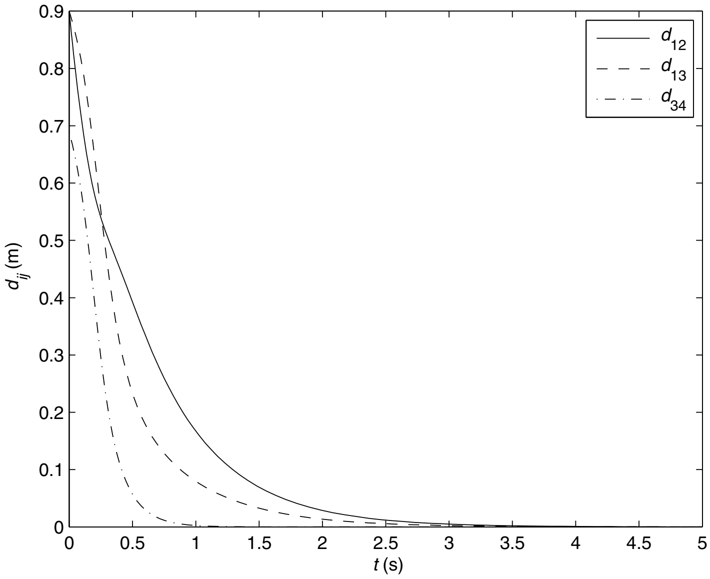

Figure 3.3: The relative distances d_{12}, d_{13} and d_{34} in Example 3.1.

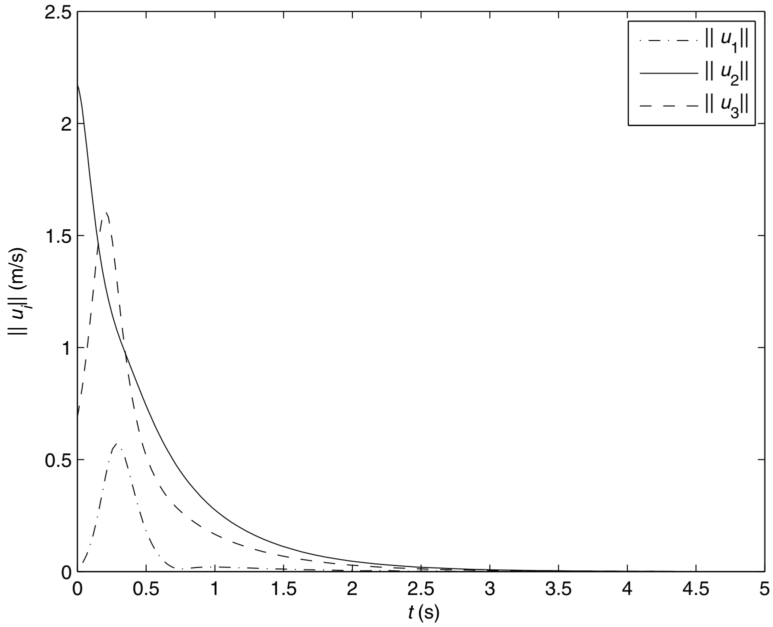

Figure 3.4: The norm of the control inputs u_1, u_2 and u_3 in Example 3.1.

It can be easily verified that the function given above satisfies the conditions in (3.5), which means that the corresponding control law is connectivity preserving. Again assume that the error function has the same form as Example 3.1.

Assume that the error function has the same form as in Example 1. Let also d and m be equal to $1m$ *and* $0.1m$*, respectively, and the initial position of each agent be marked by its label, as shown in Fig. 3.6. This figure shows the motion of the agents in the two dimensional plane. The resultant relative distances are sketched in Fig. 3.7, which confirm that the connectivity is preserved in the presence of the measurement error given above. The corresponding control inputs* $\|u_4\|$, $\|u_5\|$ *and* $\|u_6\|$ *are depicted in Fig. 3.8. This figure shows that the control inputs*

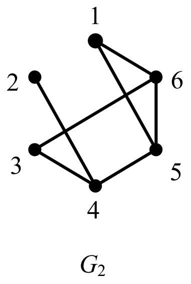

Figure 3.5: The information flow graph for the multi-agent system of Example 3.2.

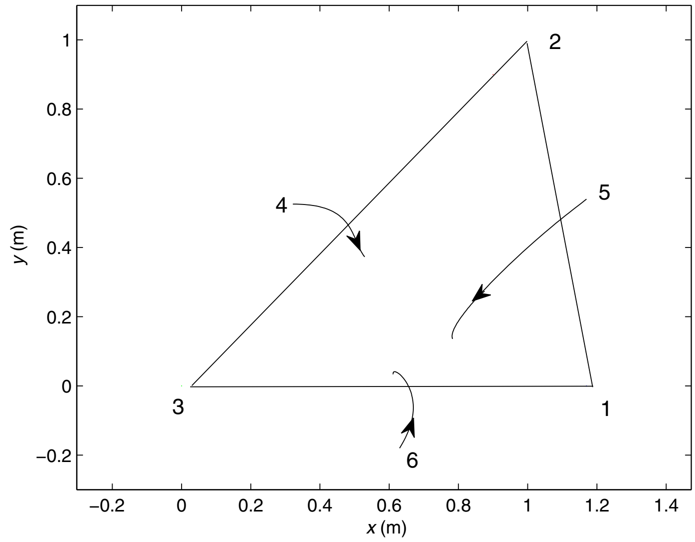

Figure 3.6: The agents' planar motion in Example 3.2.

are bounded, although some of the agents are initially about to lose connectivity.

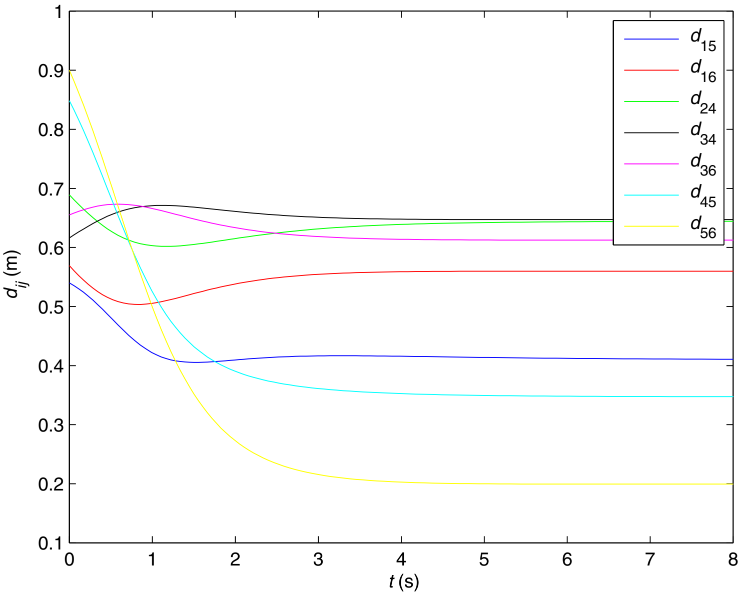

Figure 3.7: The relative distances d_{15}, d_{16}, d_{24}, d_{34}, d_{36}, d_{45} and d_{56} in Example 3.2.

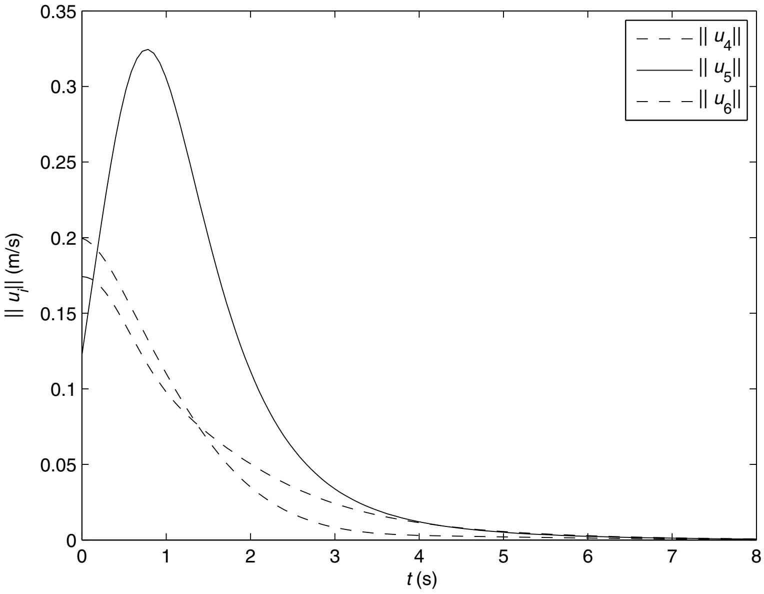

Figure 3.8: The norm of the control inputs u_4, u_5 and u_6 in Example 3.2.

Chapter 4

Formation Control of a Multi-agent System with Collision Avoidance Scheme and Measurement Error

In this chapter, the formation control problem for a team of single-integrator agents subject to distance measurement error is discussed. A decentralized navigation function is proposed to move the agents toward a desired final configuration which is defined based on the pairwise distances of the connected agents and the characteristics of the distance measurement error. The efficacy of the proposed control strategy is demonstrated by simulation.

4.1 Problem Formulation

Notation 2 *A symmetric positive semidefinite matrix A is represented as $A \succeq 0$. Similarly, $A \succ 0$, $A \preceq 0$ and $A \prec 0$ denote a matrix A that is positive definite, negative semidefinite and negative definite, respectively.*

Consider a set of N single-integrator agents represented by:

$$\dot{q}_i(t) = u_i(t) \tag{4.1}$$

where $q_i(t)$ and $u_i(t)$ denote the position and velocity of agent i at time t, respectively. Denote with $G = (V,E)$ the information flow graph, with $V = \{1,\ldots,N\}$ the nodes (agents), and with $E \subset V \times V$ the edges (communication links between agents). It is assumed that the information flow graph G is connected and undirected. Define a bounded workspace F with radius R_F, and assume the agents are point masses inside this workspace. Assume also that there are S fixed point obstacles $p_1,\ldots,p_S$ in the workspace.

The set of neighbors of agent i, denoted by $N_i(G)$, is a set consisting of any vertex in G which is connected to vertex i by an edge, i.e. $N_i := \{j \neq i | (i,j) \in E\}$. Moreover, the degree of the set of neighbors N_i is denoted by $d_i(G)$, i.e. $d_i(G) := |N_i|$.

Assume that each agent can only communicate with its neighbors (which implies limited communication among the agents). Like Chapter 3, let the error function for distance measurement between any two agents i and j be denoted by $\varepsilon_{ij}(\|q_i - q_j\|)$. Here ε_{ij} is assumed to be a C^2 positive scalar function of distance [84]. The distance measurement error is assumed to be always less than or equal to the actual distance between the two agents, i.e.:

$$\varepsilon_{ij}(\|q_i - q_j\|) \leq \|q_i - q_j\| \tag{4.2}$$

Furthermore, the derivative of the error function is assumed to be bounded by 1, as follows:

$$\varepsilon_{ij}^{'} = \frac{d\varepsilon_{ij}(\|q_i - q_j\|)}{d\|q_i - q_j\|} < 1 \tag{4.3}$$

For simplicity of notation, the argument of the function $\varepsilon_{ij}(\cdot)$ is omitted when no confusion can arise.

Definition 4.1 *The communication region of agent i is a circle of radius R_i (communication radius) centered at the agent, and any agent inside this region is considered as a neighbor of agent i. The set of all neighbors of agent i is given by:*

$$N_i := \{j \neq i \mid \|q_i - q_j\| - \varepsilon_{ij} \leq R_i\} \tag{4.4}$$

Definition 4.2 *The* watch zone *of agent i is defined as a circle of radius χ_i centered at the agent, where $\chi_i < R_i$. Any agent, fixed obstacle or workspace boundary inside the watch zone is considered as an obstacle.*

The set of obstacles of agent i, where the obstacles are considered as the neighboring agents, is defined as:

$$O_i := \{j \neq i \,|\, \|q_i - q_j\| - \varepsilon_{ij} \leq \chi_i\} \tag{4.5}$$

For simplicity, all the agents are supposed to have the same communication radius, $R_i = R$, and the same watch zone radius $\chi_i = \chi$, for all $i \in \{1, \ldots, N\}$. Assume that L is the exact distance between the two agents i and j, and that due to the sensor error the measured distance is $L - \varepsilon_{ij}(\|q_i - q_j\|)$. Thus, if the distance $L - \varepsilon_{ij} < R_i$ is measured by the sensor of agent i, normally implying that the agent j is in the communication region of the agent i, but probably the real distance can be as great as $L \geq R_i$, resulting in losing the desired configuration.

4.2 Distributed Navigation Functions

Navigation functions (NF) are real-valued maps realized through cost functions $\varphi(q)$, whose negated gradient field is attractive towards the desired pairwise distances and repulsive with respect to the obstacles (fixed point obstacles, potential agents inside the obstacle region, and also the boundary of the workspace) [44]. The main objective of a multi-agent control system is to reach a desired configuration in the workspace in terms of the relative position of the connected agents in the presence of distance measurement error. The control action for agent i is:

$$u_i = -\alpha \nabla \varphi(q_i) \tag{4.6}$$

where α is a positive scaling factor. The function $\varphi(q_i)$ is defined as:

$$\varphi(q_i) = \frac{\gamma(q_i)}{(\gamma(q_i)^k + \beta(q_i))^{1/k}} \tag{4.7}$$

where $\gamma(q_i)$ is the goal function and $\beta(q_i)$ is the obstacle function, which will be introduced later, and k is a tuning parameter ($k \geq 1$). It will be shown later that (4.7) meets all the requirements of a navigation function.

4.3 Construction of the Goal Function

The goal function $\gamma(q_i) : F \rightarrow \mathbb{R}_+$ has a unique minimum, which occurs when agent i is at the desired position w.r.t. its neighbors. It is defined as the summation of pairwise goal functions γ_{ij}, for all agents distinct from i:

$$\gamma(q_i) = \sum_{j=1, j\neq i}^{N} \gamma_{ij}(q_i, q_j). \tag{4.8}$$

The special structure of the goal function given above guarantees that the objective is satisfied if and only if the distance of agent i from all other agents approach the desired final distances. Note that the function $\gamma_{ij}(q_i, q_j)$ depends on the measured distances and the desired final distances w_{ij}'s for the error-free case as follows:

$$\gamma_{ij} = \begin{cases} \frac{1}{w_{ij}^2}(\|q_i - q_j\| - \varepsilon_{ij} - w_{ij})^2 & \|q_i - q_j\| - \varepsilon_{ij} \leq w_{ij} \\ \frac{1}{1+e^{a(\|q_i-q_j\|-\varepsilon_{ij}-(\frac{R+w_{ij}}{2}))}} & \|q_i - q_j\| - \varepsilon_{ij} > w_{ij} \end{cases} \tag{4.9}$$

where w_{ij} is assumed to be greater than χ for all $i, j \in \{1, \ldots, N\}$. Note also that γ_{ij} is chosen as a convex function at a sufficiently small vicinity of the point $\|q_i - q_j\| = \varepsilon_{ij} + w_{ij}$, because $\varphi(q_i)$ is to be minimum if agent i is positioned desirably w.r.t. its neighbors. To this end, a quadratic function is chosen for the first interval $\|q_i - q_j\| - \varepsilon_{ij} \leq w_{ij}$. Moreover, γ_{ij} takes its maximum value 1 at $\|q_i - q_j\| - \varepsilon_{ij} = 0$, i.e., when collision occurs. For the second interval, a sigmoid function is considered with the tuning parameter a. This coefficient is chosen in such a way that γ_{ij} equals zero as $\|q_i - q_j\| - \varepsilon_{ij}$ approaches w_{ij}, and equals one for $\|q_i - q_j\| - \varepsilon_{ij} > R$. It is important to note that γ_{ij} needs to be twice differentiable, so that the smooth functions are used in the construction of $\gamma(q_i)$. Note also that w_{ij} is the desired mutual distance between agents i and j in the absence of measurement error. However, in spite of the distance measurement error, agents will not arrive at the previous desired distances and in fact the desired configuration is being changed. To find the desired formation, the following equation needs to be solved for $\|q_i - q_j\|$:

$$\|q_i - q_j\| - \varepsilon_{ij}(\|q_i - q_j\|) - w_{ij} = 0 \tag{4.10}$$

If it is assumed that ε_{ij} is bounded by a known constant $\bar{\varepsilon}$, i.e. $\varepsilon_{ij} < \bar{\varepsilon}$. Thus, the new desired distance $\tilde{w}_{ij}$ between agents i and j in the presence of measurement error would be bounded by $w_{ij} + \bar{\varepsilon}$, i.e. $\|q_i - q_j\| = \tilde{w}_{ij}$, $\tilde{w}_{ij} < w_{ij} + \bar{\varepsilon}$, for $i, j \in \{1, \ldots, N\}$.

4.4 Construction of the Watch Zone Function

As noted earlier, the main objective of this work is to design a formation control scheme with the collision avoidance feature. Three types of collision are considered:

- Collision between two agents, represented by *collision avoidance function* $\beta_1(q_i, q_j)$ for agents i and j.
- Collision between an agent and a fixed obstacle, represented by *obstacle avoidance function* $\beta_2(q_i, p_k)$ for agent i and obstacle k.
- Collision of an agent and the boundary of the workspace, represented by *boundary avoidance function* $\beta_3(q_i)$ for agent i.

The watch zone of agent i is chosen as the product of the functions defined above, for all agents $j \in O_i$ and obstacles $k \in \{1, \ldots, S\}$, i.e.:

$$\beta(q_i) = \beta_3(q_i) \prod_{j \in O_i} \beta_1(q_i, q_j) \prod_{k \in \{1, \ldots, S\}} \beta_2(q_i, p_k) \tag{4.11}$$

With this choice of watch zone, it is guaranteed that if any type of collision (introduced above) occurs, the objective is not satisfied. The procedure of designing the obstacle functions is described in the sequel.

Ideally the obstacle function $\beta_1(q_i, q_j)$ associated with agents i and j should be equal to zero if agent i collides with agent j, and should be equal to one if the relative distance between the two agents is greater than χ:

$$\beta_1(q_i, q_j) = \frac{1}{1 + e^{b(\|q_i - q_j\| - \varepsilon_{ij} - \frac{\chi}{2})}} \tag{4.12}$$

where b is a tuning factor which is chosen in such a way that the β_1 approaches two specific values as the agents get close to far from each other. More precisely, β_1 reaches its unique minimum when the two agents collide, and its maximum when the two agents are outside each other's watch zone. It is important to note that β_1 belongs to C^2, and hence one can take its gradient.

Assume there are S fixed obstacles $\{p_1, \dots, p_S\}$ in the workspace. For any agent q_i and fixed obstacle p_k define the following function:

$$\beta_2(q_i, p_k) = \frac{1}{1 + e^{c(\|q_i - p_k\| - \varepsilon_{ik} - \frac{\chi}{2})}} \tag{4.13}$$

If agent i is sufficiently far from the obstacles, then β_2 approaches one, which implies that it will not play any role in the product of the obstacle functions. If, on the other hand, a collision occurs, then the corresponding obstacle function becomes zero, which in turn makes the product of the obstacle functions zero.

As for the boundary of the workspace, the controller which is proposed later in the paper treats it as infinitely many obstacles at radius R_F, which is the radius of the workspace. Now, similar to the obstacle function defined for the agent, consider a circle of radius $R_F - \chi > w_{ij}$ for the boundary. The region between this circle and the workspace boundary is defined as the *boundary avoidance margin*.

$$\beta_3(q_i) = \frac{1}{1 + e^{u(\|q_i\| - \varepsilon_i - (R_F - \frac{\chi}{2}))}} \tag{4.14}$$

The function $\beta_3(\cdot)$ has the property that it is equal to 1 as long as the corresponding agent is outside the boundary avoidance margin, and converges to 0 as the agent approaches the above region.

4.5 Navigation Function Analysis

In this section, it is desired to show that $\varphi(q_i)$ is a navigation function. Navigation functions are used as control tools to direct the agents to their desired locations in the formation, where they are positioned properly w.r.t. their neighbors, in the presence of distance measurement error. However, connectivity preservation of an agent and its neighbors is not guaranteed under such a navigation-function-based control strategy. As a remedy to this problem, it is assumed that a certain level of connection will always be held in the global system (this is in fact, a realistic assumption in practice, and simulations also verify that). The objective now is to investigate how the navigation function drives each agent to its desired location under the above assumption. To this end, let the navigation function be formally defined first.

Definition 4.3 *Let $F \subset R^{2N}$ be a compact connected analytic manifold and denote its boundary with ϑF. A map $\varphi : F \to [0,1]$ is a navigation function if [44]:*

1. *It is analytic on F.*
2. *It has a unique minimum at $q_d \in int(F)$ (i.e. it is* Polar *on F).*
3. *Its Hessian at all critical points (zero gradient vector field) is full-rank (i.e., it is* Morse *on F).*
4. *$lim_{q \to \vartheta F} \varphi(q) = 1$ (i.e., it is admissible on F).*

Given $\xi > 0$, define $\beta_{i,l}^{nc}(\xi) = \{q_i : 0 < \beta_l(q_i, \cdot) < \xi, l \in \{1,2,3\}\}$. Partition the workspace to four regions of interest as follows [44]:

1. The desired destination $F_d(q_i) = \{q_i : \|q_i - q_j\| - \varepsilon_{ij} = w_{ij}, \forall j \in N_i\}$;
2. the workspace boundary ϑF;

3. the set representing near collision regions $F_0(\xi) = \bigcup_{l\in\{1,2,3\}} \beta_{i,l}^{nc}(\xi) - F_d(q_i)$, and

4. the set representing the region sufficiently far from the watch zone $F_1(\xi) = F - (F_d(q_i) \cup \vartheta F \cup F_0(\xi))$.

To verify that $\varphi(\cdot)$ is a navigation function, it suffices to show that when the agents are located at the desired distances w.r.t. their neighbors, it constitutes an equilibrium point which is a non-degenerate local minimum. Furthermore, $\varphi(q)$ has no other critical points of the above form in the other subsets.

Lemma 4.1 *The function $\varphi(\cdot)$ has a non-degenerate minimum at the desired formation.*

Proof. To find the critical points of $\varphi(q_i)$, it is required to derive its gradient first. One can write:

$$\nabla\varphi(q_i) = \frac{k\beta\nabla\gamma - \gamma\nabla\beta}{k(\gamma^k+\beta)^{\frac{1}{k}+1}} \tag{4.15}$$

where

$$\nabla\gamma(q_i) = \sum_{j=1,j\neq i}^{N} \nabla\gamma_{ij}(q_i,q_j) =$$

$$\begin{cases} \sum\limits_{j=1,j\neq i}^{N} \frac{2}{w_{ij}^2}(1-\frac{\varepsilon_{ij}+w_{ij}}{\|q_i-q_j\|}). & \\ (1-\varepsilon_{ij}^{'})(q_i-q_j)^T & \|q_i-q_j\|-\varepsilon_{ij} \leq w_{ij} \\ \sum\limits_{j=1,j\neq i}^{N} \frac{-ae^{a(\|q_i-q_j\|-\varepsilon_{ij}-(\frac{R+w_{ij}}{2}))}}{(1+e^{a(\|q_i-q_j\|-\varepsilon_{ij}-(\frac{R+w_{ij}}{2}))})^2}. & \\ (1-\varepsilon_{ij}^{'})\frac{(q_i-q_j)^T}{\|q_i-q_j\|} & \|q_i-q_j\|-\varepsilon_{ij} > w_{ij} \end{cases}$$

and

$$\nabla\beta(q_i) = \sum_{j\in O_i} \nabla\beta_1(q_i,q_j)\frac{\beta(q_i)}{\beta_1(q_i,q_j)} + \sum_{k\in\{1,\ldots,S\}} \nabla\beta_2(q_i,p_k)\frac{\beta(q_i)}{\beta_2(q_i,p_k)} + \nabla\beta_3(q_i)\frac{\beta(q_i)}{\beta_3(q_i)} \tag{4.16}$$

The critical points of $\varphi(q_i)$ are derived from the relation below:

$$\nabla\varphi(q_i) = 0 \Leftrightarrow k\beta(q_i)\nabla\gamma(q_i) = \gamma(q_i)\nabla\beta(q_i) \tag{4.17}$$

To show that this equilibrium is non-degenerate, it is required to prove that the Hessian is positive definite. One can write:

$$\nabla^2\varphi = \frac{1}{k(\gamma^k+\beta)^{\frac{1}{k}+2}}\{(\gamma^k+\beta)[k\nabla\beta\nabla\gamma^T - \nabla\gamma\nabla\beta^T + k\beta\nabla^2\gamma - \gamma\nabla^2\beta] - (\frac{1}{k}+1)[k\beta\nabla\gamma - \gamma\nabla\beta][k\gamma^{k-1}\nabla\gamma + \nabla\beta]^T\} \tag{4.18}$$

At the critical points the relation $k\beta\nabla\gamma = \gamma\nabla\beta$ holds. This can be used to simplify the above matrix as:

$$\begin{aligned}\nabla^2\varphi \;=\; & \frac{\gamma^k+\beta}{k(\gamma^k+\beta)^{\frac{1}{k}+2}}[k\nabla\beta\nabla\gamma^T - \nabla\gamma\nabla\beta^T + k\beta\nabla^2\gamma \\ & -\gamma\nabla^2\beta] = \frac{1}{k(\gamma^k+\beta)^{\frac{1}{k}+1}}[(1-\frac{1}{k})[\frac{\gamma}{\beta}\nabla\beta\nabla\beta^T] \\ & +k\beta\nabla^2\gamma - \gamma\nabla^2\beta]\end{aligned} \tag{4.19}$$

Since $k \geq 1$, $\frac{\gamma}{\beta} > 0$ and $\nabla\beta\nabla\beta^T \succeq 0$, hence $(1-\frac{1}{k})[\frac{\gamma}{\beta}\nabla\beta\nabla\beta^T] \succeq 0$. It is now required to show that the only remaining term, i.e. $k\beta\nabla^2\gamma - \gamma\nabla^2\beta$, is positive definite at the desired formation. To this end, it is important to note that $\nabla^2\beta = 0$ at the desired equilibrium. This results from the fact that, $\beta = 1$ as long as $w_{ij} > \chi$ and $w_{ij} < R_F - \chi$. Thus, the gradient and the Hessian are both equal to zero. Therefore, it suffices to show that $\nabla^2\gamma \succ 0$.

$$\begin{aligned}\nabla^2\gamma = {} & [\frac{2}{w_{ij}^2}(1-\frac{\varepsilon_{ij}+w_{ij}}{\|q_i-q_j\|})(1-\varepsilon^{'}_{ij})I] - [\frac{2}{w_{ij}^2}(1- \\ & \frac{\varepsilon_{ij}+w_{ij}}{\|q_i-q_j\|})(q_i-q_j)(q_i-q_j)^T\frac{\varepsilon^{''}_{ij}}{\|q_i-q_j\|}] - [\frac{2}{w_{ij}^2} \\ & (1-\varepsilon^{'}_{ij})(q_i-q_j)(q_i-q_j)^T(\frac{\varepsilon^{'}_{ij}-\frac{\varepsilon_{ij}+w_{ij}}{\|q_i-q_j\|}}{\|q_i-q_j\|^2})]\end{aligned} \tag{4.20}$$

Using the characteristics of desired formation, one arrives at:

$$\nabla^2\gamma = \frac{2}{w_{ij}^2}(1-\varepsilon^{'}_{ij})^2\frac{(q_i-q_j)(q_i-q_j)^T}{\|q_i-q_j\|^2} \tag{4.21}$$

Now, since $\varepsilon^{'}_{ij} < 1$, $\|q_i - q_j\| - \varepsilon_{ij} = w_{ij}$ and $q_i \neq q_j$, hence $\nabla^2 \gamma_i \succ 0$. ■

Lemma 4.2 *All the Critical points of $\varphi(\cdot)$ are in the interior of the workspace.*

Proof. Assume that agent i (which is one of the critical points of $\varphi(q_i)$) is located on the boundary and that collision occurs between agents i, j. Then $\beta(q_i) = 0$, and thus from (4.15):

$$\nabla\varphi(q_i) = \frac{1}{k(\gamma^k + \beta)^{\frac{1}{k}+1}}(k\beta\nabla\gamma - \gamma\nabla\beta) = -\frac{\nabla\beta}{k\gamma^k} \tag{4.22}$$

Since agent i is on the boundary and at least one collision has occurred, it is concluded that:

$$\nabla\beta(q_i) \neq 0 \Rightarrow \nabla\varphi(q_i) \neq 0 \tag{4.23}$$

which contradicts the initial assumption that this is a critical point. This completes the proof. ■

Lemma 4.3 *For every $\xi > 0$, there exists a positive integer $N(\xi)$ such that if $k > N(\xi)$, then none of the critical points of $\varphi(q_i)$ in are $F_1(\xi)$.*

Proof. Let $q \in F_1(\xi)$ be a critical point. Then:

$$k\beta\nabla\gamma = \gamma\nabla\beta \tag{4.24}$$

which yields:

$$k\beta\|\nabla\gamma\| = \gamma\|\nabla\beta\| \tag{4.25}$$

Hence, a sufficient condition for q not to be a critical point is:

$$\frac{\gamma\|\nabla\beta\|}{\beta\|\nabla\gamma\|} < k, \text{ for all } q \in F_1(\xi) \tag{4.26}$$

Note that we are analyzing the critical points which are away of the obstacles, $\beta_l(q_i,\cdot) > \xi$ for all $l \in \{1,2,3\}$. Then, an upper bound for the left side is given by:

$$\begin{aligned}\frac{\gamma\|\nabla\beta\|}{\beta\|\nabla\gamma\|} <& \frac{\gamma_i}{\|\nabla\gamma_i\|}(\frac{\|\sum_{j\in O_i}\nabla\beta_1(q_i,q_j)\frac{\beta}{\beta_1(q_i,q_j)}\|}{\beta}\\ &+\frac{\|\sum_{k\in\{1,\ldots,S\}}\nabla\beta_2(q_i,p_k)\frac{\beta}{\beta_2(q_i,p_k)}\|}{\beta}\\ &+\frac{\|\nabla\beta_3(q_i)\frac{\beta}{\beta_3(q_i)}\|}{\beta})\\ <&\frac{1}{\xi}\frac{\max(\gamma)}{\min(\|\nabla\gamma\|)}(\sum_{j\in O_i}\max\left(\|\nabla\beta_1(q_i,q_j)\|\right)\\ &+\sum_{k\in\{1,\ldots,S\}}\max\left(\|\nabla\beta_2(q_i,p_k)\|\right)\\ &+\max\left(\|\nabla\beta_3(q_i)\|\right)\end{aligned}$$

The proof follows immediately by choosing:

$$\begin{aligned}N(\xi) = &\frac{1}{\xi}\frac{\max(\gamma)}{\min(\|\nabla\gamma\|)}(\sum_{j\in O_i}\max\left(\|\nabla\beta_1(q_i,q_j)\|\right)\\ &+\sum_{k\in\{1,\ldots,S\}}\max\left(\|\nabla\beta_2(q_i,p_k)\|\right)+\max\left(\|\nabla\beta_3(q_i)\|\right).\end{aligned}$$

■

Lemma 4.4 *There exists $\xi_0 > 0$ such that $\varphi(q_i)$ has no local minimum in $F_0(\xi)$, as long as $\xi < \xi_0$.*

Proof. If $q \in F_0(\xi)$ is a critical point of φ_i, then $q \in \beta_{i,l}^{nc}(\xi)$ for some i. This implies that q is very close to some obstacles. It is desired now to show that $\nabla^2\varphi(q_i)$ has at least one negative eigenvalue. At a critical point the following relation holds:

$$\nabla\varphi(q_i) = 0 \Rightarrow k\beta\nabla\gamma = \gamma\nabla\beta \tag{4.27}$$

The Hessian at a critical point is given by:

$$\begin{aligned}\nabla^2\varphi(q_i) = \frac{1}{k(\gamma^k+\beta)^{\frac{1}{k}+1}}&[(1-\frac{1}{k})[\frac{\gamma}{\beta}\nabla\beta\nabla\beta^T]\\ &+k\beta\nabla^2\gamma - \gamma\nabla^2\beta]\end{aligned} \tag{4.28}$$

Define $\beta = \beta_\lambda\bar{\beta}_\lambda$ where β_λ is one of the collision functions appears in $\beta(q_i)$ and $\bar{\beta}_\lambda$ is product of all the collision functions except β_λ. One can choose a proper sigmoid function for β_λ. For instance, by using a function of the form $\beta_1(q_i, q_j)$ (the collision avoidance function), one can write:

$$\begin{aligned}\nabla^2\varphi(q) = \frac{1}{k(\gamma^k+\beta)^{\frac{1}{k}+1}}&(k\beta\nabla^2\gamma+(1-\frac{1}{k})\frac{\gamma}{\beta}\\ &[\beta_\lambda^2\nabla\bar{\beta}_\lambda\nabla\bar{\beta}_\lambda^T+2\beta_\lambda\bar{\beta}_\lambda(\nabla\bar{\beta}_\lambda\nabla\beta_\lambda^T)+\\ &\bar{\beta}_\lambda^2\nabla\beta_\lambda\nabla\beta_\lambda^T]-\gamma[\beta_\lambda\nabla^2\bar{\beta}_\lambda+2\\ &(\nabla\bar{\beta}_\lambda^T\nabla\beta_\lambda)+\bar{\beta}_\lambda\nabla^2\beta_\lambda])\end{aligned} \tag{4.29}$$

Choose a test vector of unit magnitude [44] which is orthogonal to $\nabla\beta_\lambda$ at a critical point q_c, i.e.:

$$\hat{v} = \frac{\nabla\beta_\lambda(q_c)^\perp}{\|\nabla\beta_\lambda(q_c)^\perp\|}$$

It is straightforward to verify that the following quadratic equation holds:

$$\begin{aligned}&k(\gamma^k+\beta)^{1+\frac{1}{k}}\hat{v}^T\nabla\varphi^2\hat{v} = k\beta\hat{v}^T\nabla^2\gamma\hat{v}-\gamma\bar{\beta}_\lambda\hat{v}^T\nabla\beta_\lambda^2\hat{v}+\\ &(1-\frac{1}{k})\frac{\gamma}{\beta}\beta_\lambda^2\hat{v}^T\nabla\bar{\beta}_\lambda\nabla\bar{\beta}_\lambda^T\hat{v}-\gamma\beta_\lambda\hat{v}^T\nabla^2\bar{\beta}_\lambda\hat{v}\end{aligned} \tag{4.30}$$

where

$$\nabla^2\gamma = G_1 I + G_2(q_i-q_j)(q_i-q_j)^T \tag{4.31}$$

$$\nabla^2\beta_\lambda = B_1 I + B_2(q_i-q_j)(q_i-q_j)^T \tag{4.32}$$

where G_1, G_2, B_1 and B_2 are the functions of $\|q_i - q_j\|$. Now, take the inner-product of $\nabla\gamma$ and both sides of the equation $k\beta\nabla\gamma = \gamma\nabla\beta$ to obtain:

$$\begin{aligned} 4k\beta\frac{(1-\varepsilon'_{ij})^2}{w_{ij}^2} &= \nabla\beta.\nabla\gamma \\ &= \bar{\beta}_\lambda\nabla\beta_\lambda.\nabla\gamma + \beta_\lambda\nabla\bar{\beta}_\lambda.\nabla\gamma \end{aligned} \tag{4.33}$$

Using the above relation in Eq. 4.30 and grouping the terms which are proportional to β_λ yields:

$$\begin{aligned} & k(\gamma^k+\beta)^{1+\frac{1}{k}}\hat{v}^T\nabla\varphi^2\hat{v} = \beta_\lambda[\frac{w_{ij}^2}{4(1-\varepsilon'_{ij})^2}(G_1+G_2\hat{v}^T(q_i-q_j) \\ & (q_i-q_j)^T\hat{v})\nabla\bar{\beta}_\lambda.\nabla\gamma+\gamma\hat{v}^T((1-\frac{1}{k})\frac{1}{\bar{\beta}_\lambda}\nabla\bar{\beta}_\lambda\nabla\bar{\beta}_\lambda^T-\nabla^2\bar{\beta}_\lambda)\hat{v}] \\ & +\bar{\beta}_\lambda[\frac{w_{ij}^2}{4(1-\varepsilon'_{ij})^2}(G_1+G_2\hat{v}^T(q_i-q_j)(q_i-q_j)^T\hat{v})\nabla\beta_\lambda.\nabla\gamma- \\ & \gamma(B_1+B_2\hat{v}^T(q_i-q_j)(q_i-q_j)^T\hat{v})]. \end{aligned} \tag{4.34}$$

There are two terms in the right side of the above equation. The first term is proportional to β_λ and can be made arbitrarily small by a proper choice of ξ. However, since this term can be positive, the second term which is proportional to $\bar{\beta}_\lambda$ should be strictly negative. Let the latter term be denoted by $Pr_{\bar{\beta}_\lambda}$, and rewritten as:

$$\begin{aligned}
Pr_{\bar{\beta}_\lambda} =& \frac{w_{ij}^2}{4(1-\varepsilon_{ij}^{'})^2}(G_1 + G_2\hat{v}^T(q_i - q_j)(q_i - q_j)^T\hat{v})\nabla\beta_\lambda.\nabla\gamma \\
& - \gamma(B_1 + B_2\hat{v}^T(q_i - q_j)(q_i - q_j)^T\hat{v}) \\
=& G_3\nabla\beta_\lambda.\nabla\gamma - B_3\gamma
\end{aligned} \tag{4.35}$$

where $G_3 := \frac{w_{ij}^2}{4(1-\varepsilon_{ij}^{'})^2}(G_1+G_2\hat{v}^T(q_i-q_j)(q_i-q_j)^T\hat{v}), B_3 := (B_1+B_2\hat{v}^T(q_i-q_j)(q_i-q_j)^T\hat{v})$. Note that G_3, B_3 and γ are strictly positive for any $q \in F_0(\xi)$ (which implies no collision has occurred). Thus, to complete the proof, it suffices to show that $\nabla\beta_\lambda.\nabla\gamma < 0$. To this end, one can write:

$$\begin{aligned}
\nabla\beta_\lambda.\nabla\gamma =& \frac{-be^{b(\|q_i-q_j\|-\varepsilon_{ij}-w_{ij})}}{(1+e^{b(\|q_i-q_j\|-\varepsilon_{ij}-w_{ij})})^2}(1-\varepsilon_{ij}^{'})\frac{(q_i-q_j)^T}{\|q_i-q_j\|}.\frac{2}{w_{ij}^2} \\
& (\|q_i-q_j\| - \varepsilon_{ij} - w_{ij})(1-\varepsilon_{ij}^{'})\frac{(q_i-q_j)^T}{\|q_i-q_j\|} \\
=& A(\|q_i-q_j\| - \varepsilon_{ij} - w_{ij})
\end{aligned} \tag{4.36}$$

where A is strictly positive. Now, for any $q \in F_0(\xi)$:

$$\begin{gathered}
0 < \beta_\lambda < \xi \Rightarrow \frac{1}{1+e^{b(\|q_i-q_j\|-\varepsilon_{ij}-\frac{\chi}{2})}} < \xi \Rightarrow \\
\|q_i - q_j\| - \varepsilon_{ij} < \frac{\chi}{2} + \frac{1}{b}ln(\frac{1}{\xi} - 1)
\end{gathered}$$

On the other hand, the maximum value of $\nabla\beta_\lambda.\nabla\gamma$ is given by:

$$\max(\nabla\beta_\lambda.\nabla\gamma) = A(\frac{\chi}{2} + \frac{1}{b}ln(\frac{1}{\xi} - 1)) \tag{4.37}$$

Therefore, the maximum value of the term proportional to $\bar{\beta}_\lambda$ in (4.34) is:

$$\max(Pr_{\bar{\beta}_\lambda}) = G_3A(\frac{\chi}{2} + \frac{1}{b}ln(\frac{1}{\xi} - 1)) - B_3\gamma \tag{4.38}$$

This means that $\max(Pr_{\bar{\beta}_\lambda})$ is strictly negative provided the following inequality holds:

$$\xi < \frac{1}{1 + e^{b(\frac{B_3}{G_3}\frac{\gamma}{A} - \frac{\chi}{2})}} \tag{4.39}$$

The proof follows now by choosing $\xi_0 = \frac{1}{1+e^{b(\frac{B_3}{G_3}\frac{\gamma}{A} - \frac{\chi}{2})}}$. ■

Proposition 4.1 *For each fixed t, the function $\varphi(q_i)$ is a navigation function if the parameter k has a value greater than a finite lower bound.*

Proof. The proof follows from Lemmas 1-4 so that the $\varphi(q_i)$ satisfies all the conditions stated in Definition 4.3. ■

4.6 Simulation Results

Example 4.1 *Consider 3 single-integrator agents moving in a two-dimensional plane under the control law (4.6) and a point obstacle fixed in the origin of the workspace. The radius* R_F *of the workspace is assumed to be* $60m$*. Let the watch zone and communication region be circles of radius* $\chi = 5m$ *and* $R = 40m$*, respectively. Let also the control action parameters be* $\alpha = 1$ *and* $k = 1$*. Assume that the error-free desired configuration is an equilateral triangle with edges of length* $w_{ij} = 15m$*. Assume also that the measurement error function has the following form:*

$$\varepsilon_{ij} = \mu(1 - e^{-\frac{\|q_i - q_j\|}{h}}) \tag{4.40}$$

It is straightforward to verify that the above function is always less than or equal to $\|q_i - q_j\|$*, and that its maximum rate of change is less than 1, for any* $\mu < h$*. Hence, it satisfies all of the required conditions for the distance measurement error described earlier. Let* h *and* μ *be equal to* $30m$ *and* $2m$*, respectively. Due to the distance measurement error, the agents will converge to a configuration where their pairwise distance is derived from the following nonlinear equation:*

$$\|q_i - q_j\| - \varepsilon_{ij}(\|q_i - q_j\|) - w_{ij} = 0 \tag{4.41}$$

For the error function (4.40) and the given values for the parameters, (4.41) can be expressed as:

$$\|q_i - q_j\| - 2(1 - e^{-\frac{\|q_i - q_j\|}{30}}) - 15 = 0 \tag{4.42}$$

whose solution is $\|q_i - q_j\| = 15.82m$. Let the initial positions of the agents be marked by asterisks and the final positions by diamonds. Let also the obstacle avoidance region and the boundary avoidance margin be represented by dashed lines. Fig. 4.1 depicts the planar motion of the agents in this case. The agents are initially connected, with the pairwise distances chosen close to R (in order to test the goal function near the boundary of the communication

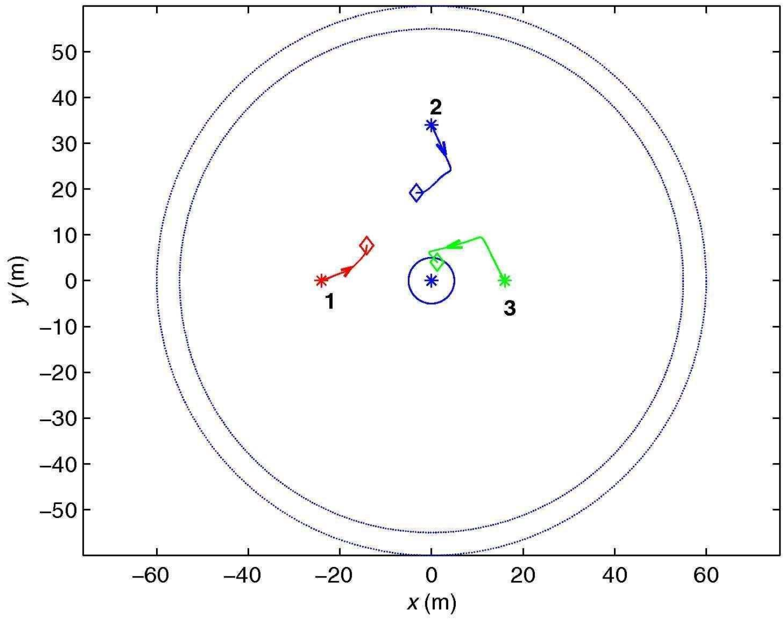

Figure 4.1: Planar motion of the agents for the case when the pairwise distances are close to R.

region). As shown in Fig. 4.2, the pairwise distances approach the desired values (i.e. $15.82m$*) as time increases.*

Fig. 4.3 shows the planar motion of the agents for the case when agent 2 starts from inside

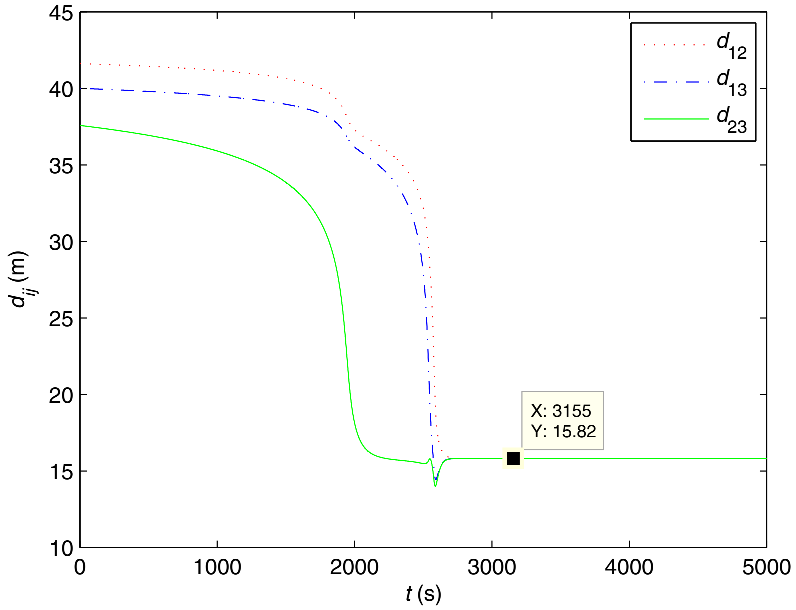

Figure 4.2: Mutual distances among the agents for the case when the pairwise distances are close to R.

the obstacle avoidance region (but no collision initially). As shown in Fig. 4.3, there is no collision in the trajectory of the agents (and in particular agent 2) as they move to their desired configuration. Fig. 4.4 demonstrates the final pairwise distances which are in accordance with the desired configuration.

Fig. 4.5 depicts an initial configuration near the boundary avoidance margin. Agent 2 moves out of this region, and finally the agents reach the desired configuration. Fig. 4.6 confirms the desired pairwise final distances between the agents.

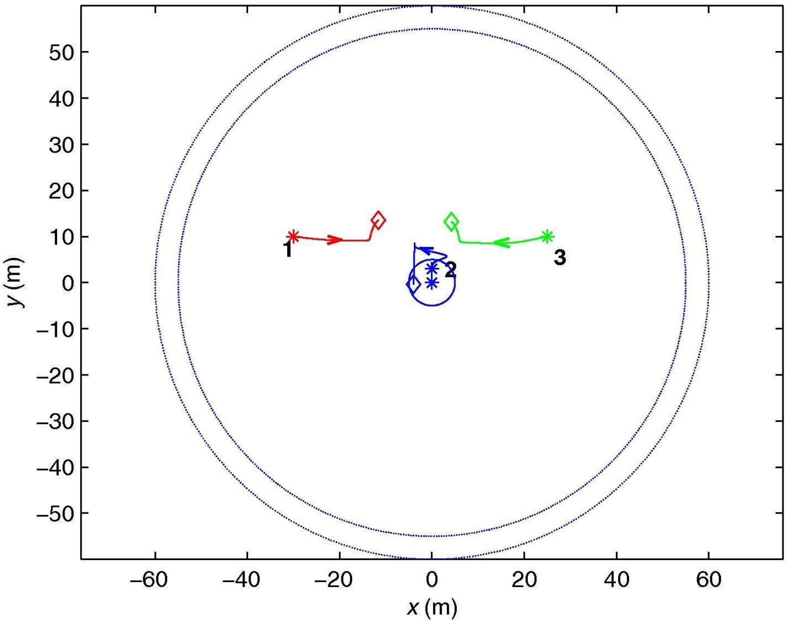

Figure 4.3: Planar motion of the agents for the case when agent 2 starts from inside the obstacle avoidance region.

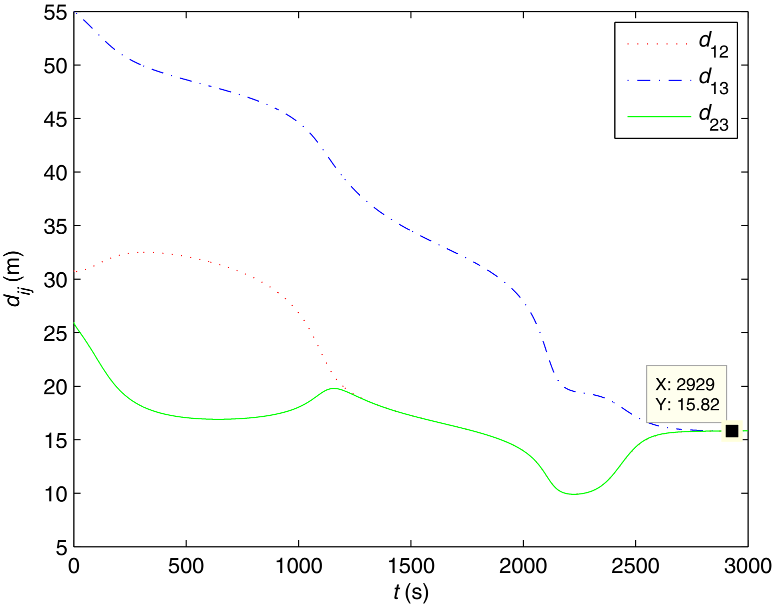

Figure 4.4: Pairwise distances of the agents for the case when agent 2 starts from inside the obstacle avoidance region.

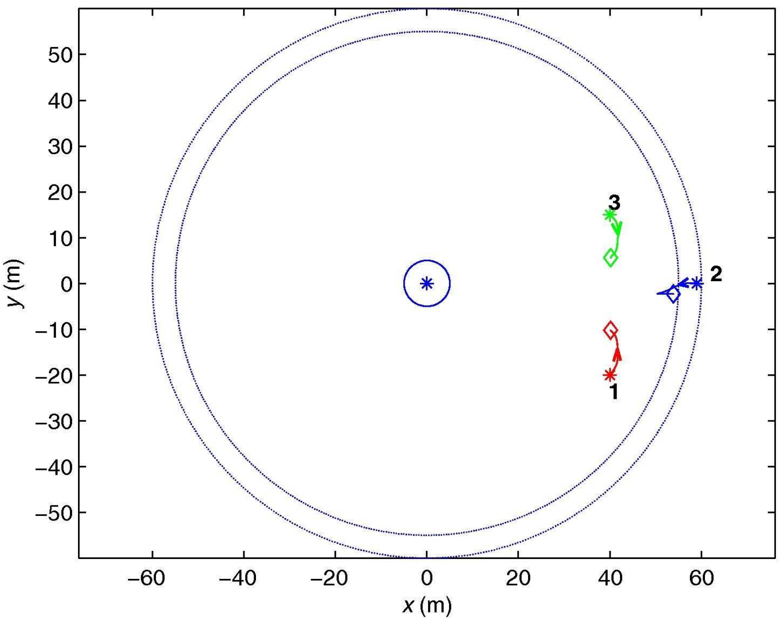

Figure 4.5: Planar motion of the agents for the case when the initial configuration is near the boundary avoidance margin.

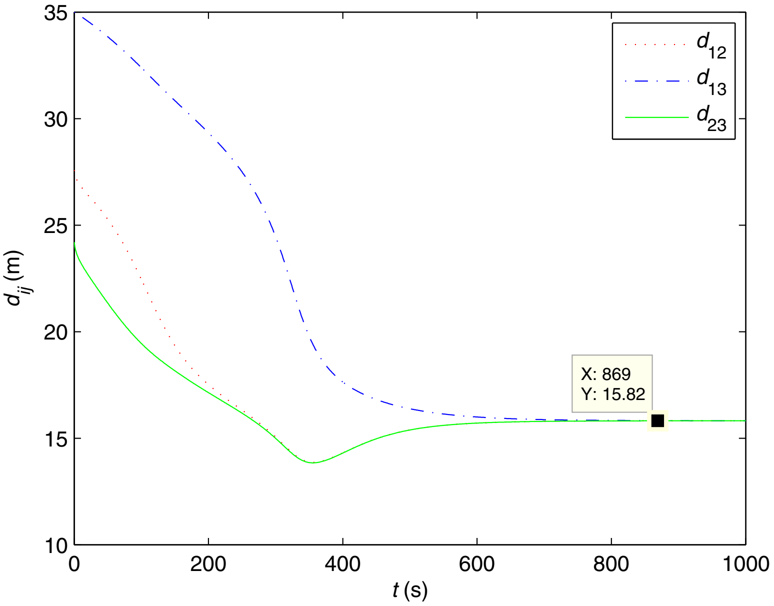

Figure 4.6: Pairwise distances of the agents for the case when the initial configuration is near the boundary avoidance margin.

Chapter 5

Conclusions and Future Work

5.1 Summary of Contributions

An error analysis concerning the control of multi-agent systems is presented in this thesis. Any cooperative control scheme relies on measurement of quantities such as distance, speed, etc. These measurements are often assumed to be exact to simplify the analysis but in practice all measured quantities are subject to error, which can deteriorate the overall performance of the network significantly.

Chapter 3 extends the connectivity preserving bounded control design technique for single-integrator agents to the case when the distance measurements are subject to error. Sufficient conditions are presented for a class of distributed potential functions, which guarantee the connectivity preservation of the resultant control laws. It is assumed that the measurement error and its rate of change are bounded, with known bounds. Unlike existing methods, the potential function given in this work can be designed in such a way that the resultant bounded control inputs are connectivity preserving, even in the presence of the measurement error. The efficacy of the proposed control strategy is demonstrated by simulation for two different choices of connectivity preserving potential functions. The selected potential function used in the first example of chapter 3 leads to consensus, while preserving the connectivity of the corresponding information flow graph during convergence. Simulation results show that the measurement error has negligible effect on the consensus problem. The second example is a containment problem, where the followers are desired to converge to the convex hull of the static leaders. Simulations confirm that the containment problem is quite sensitive to the measurement error.

Furthermore, in chapter 4 a distributed navigation function-based controller is proposed to drive a group of single-integrator agents to a desired configuration. It is assumed that the distance measurements are subject to error, and that the agents should avoid collision and obstacles in the workspace (as well as the boundaries of the workspace). The final formation is expressed in terms of the desired distances between the connected agents and the distance measurement error. The formation can be reached anywhere in the space and with any orientation provided some sufficient conditions on the magnitude of the distance measurement error and its rate of its change hold. The navigation functions used to design the controller ensure collision avoidance between the agents as well as obstacle and workspace-boundary avoidance. The efficacy of the proposed control strategy is demonstrated by simulation for three different cases in terms of the initial position of agents.

5.2 Suggestions for Future Work

Distance measurement error analysis for multi-agent control systems is a topic motivated by practical considerations, which to the best of the author's knowledge is investigated in this work for the first time. Typically, every agent in a multi-agent network is equipped with sensors to measure distance, it is important to take into account the distance measurement error caused by any sensor deviations in practice.

There are a number of future research directions related to this topic, and in this section two of them are presented for interested researchers. First of all, one can consider different type of agent dynamics such as double-integrator agents or nonholonomic agents. Note that the single-integrator dynamics is often used for this type of system to simplify the corresponding analysis. It is known, however, that acceleration control in double-integrator agents is more feasible than velocity control in the single-integrator case.

On the other hand, one can consider a more precise model for error, compared to the one considered in this work. Two possible approaches are as follows:

1. Representing error as a random variable and using a probabilistic approach for the error

analysis.

2. Considering different values for the measurement error of different agents (as opposed to the identical models considered in this thesis). For this purpose, the error can be formulated as a vector, where different elements of the vector can be correlated with each other (for example, this type of model is desirable for GPS-based measurements).

Bibliography

[1] D. V. Dimarogonas and K. J. Kyriakopoulos, “Connectedness preserving distributed swarm aggregation for multiple kinematic robots,” *IEEE Transactions on Robotics and Automation*, vol. 24, no. 5, pp. 1213–1223, 2008.

[2] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. K. Hedrick, “An overview of emerging results in cooperative UAV control,” in *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp. 602–607, 2004.

[3] R. Horowitz and P. Varaiya, “Control design of an automated highway system,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 913–925, 2000.

[4] C. Tomlin, G. J. Pappas, and S. Sastry, “Conflict resolution for air traffic management: A study in multiagent hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998.

[5] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, “Collective motion, sensor networks, and ocean sampling,” *Proceedings of the IEEE*,

vol. 95, no. 1, pp. 48–74, 2007.

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

[7] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[8] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.

[9] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[10] J. Cortes, S. Martinez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.

[11] G. Lafferriere, A. Williams, J. Caughman, and J. J. P. Veerman, "Decentralized control of vehicle formations," *Systems and Control Letters*, vol. 54, no. 9, pp. 899–910, 2005.

[12] A. Ajorlou, A. Momeni, and A. G. Aghdam, "Convergence analysis for a class of nonlinear consensus algorithms," in *Proceedings of the American Control Conference*, pp. 6318–6323, 2010.

[13] P. Hovareshti, J. S. Baras, and V. Gupta, "Average consensus over small world networks: A probabilistic framework," in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 375–380, 2008.

[14] R. Olfati-Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proceedings of the American Control Conference*, pp. 951–956, 2003.

[15] A. Ajorlou, A. Momeni, and A. G. Aghdam, "Sufficient conditions for the convergence of a class of nonlinear distributed consensus algorithm," *Automatica*, vol. 47, no. 3, pp. 625–629, 2011.

[16] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1972–1975, 2008.

[17] Y. Cao and W. Ren, "Containment control with multiple stationary or dynamic leaders under a directed interaction graph," in *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference*, pp. 3014–3019, 2009.

[18] D. V. Dimarogonas, M. Egerstedt, and K. J. Kyriakopoulos, “A leader-based containment control strategy for multiple unicycles,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 5968–5973, 2006.

[19] Y. Cao, D. Stuart, W. Ren, and Z. Meng, “Distributed containment control for multiple autonomous vehicles with double-integrator dynamics: Algorithms and experiments,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 929–938, 2011.

[20] T. H. Summers, C. Yu, and B. D. Anderson, “Robustness to agent loss in vehicle formations & sensor networks,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 1193–1199, 2008.

[21] P. Yang, R. A. Freeman, and K. M. Lynch, “Multi-agent coordination by decentralized estimation and control,” *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2480–2496, 2008.

[22] D. P. Spanos and R. M. Murray, “Robust connectivity of networked vehicles,” in *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp. 2893–2898, 2004.

[23] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian,” in *Proceedings of the American Control Conference*, pp. 99–103, 2005.

[24] M. C. D. Gennaro and A. Jadbabaie, “Decentralized control of connectivity for multi-agent systems,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3628–3633, 2006.

[25] M. M. Zavlanos and G. J. Pappas, “Distributed connectivity control of mobile networks,” in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 3591–3596, 2007.

[26] D. V. Dimarogonas and K. J. Kyriakopoulos, “On the rendezvous problem for multiple nonholonomic agents,” *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 916–922, 2007.

[27] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, “Flocking while preserving network connectivity,” in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 2919–2924, 2007.

[28] M. Ji and M. Egerstedt, “Distributed coordination control of multiagent systems while preserving connectedness,” *IEEE Transactions on Robotics and Automation*, vol. 23, no. 4, pp. 693–703, 2007.

[29] M. Ji and M. Egerstedt, “Distributed formation control while preserving connectedness,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 5962–5967, 2006.

[30] M. Ji and M. Egerstedt, “Connectedness preserving distributed coordination control over dynamic graphs,” in *Proceedings of the American Control Conference*, pp. 93–98, 2005.

[31] D. V. Dimarogonas and K. H. Johansson, “Decentralized connectivity maintenance in mobile networks with bounded inputs,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1507–1512, 2008.

[32] A. Ajorlou, A. Momeni, and A. G. Aghdam, “A class of bounded distributed control strategies for connectivity preservation in multi-agent systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2828–2833, 2010.

[33] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[34] V. Gupta, B. Hassibi, and R. Murray, “Stability analysis of stochastically varying formations of dynamic agents,” in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 504–509, 2003.

[35] P. Ogren, M. Egerstedt, and X. Hu, “A control Lyapunov function approach to multiagent coordination,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 847–851, 2002.

[36] H. Yamaguchi and J. W. Burdick, “Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3573–3580, 1998.

[37] R. Olfati-Saber, W. B. Dunbar, and R. M. Murray, “Cooperative control of multi-vehicle systems using cost graphs and optimization,” in *Proceedings of the American Control Conference*, pp. 2217–2222, 2003.

[38] T. Balch and R. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1999.

[39] T. Hayakawa, T. Matsuzawa, and S. Hara, “Formation control of multi-agent systems with sampled information-relationship between information exchange structure and control performance,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 4333–4338, 2006.

[40] F. Xiao, L. Wang, J. Chen, and Y. Gao, “Finite-time formation control for multi-agent systems,” *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009.

[41] S. Mastellone, D. M. Stipanovic, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, “Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments,” *International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[42] X. Wang, V. Yadav, and S. N. Balakrishnan, “Cooperative UAV formation flying with obstacle/collision avoidance,” *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 672–679, 2007.

[43] M. C. D. Gennaro and A. Jadbabaie, “Formation control for a cooperative multi-agent system using decentralized navigation functions,” in *Proceedings of the American Control Conference*, pp. 1346–1351, 2006.

[44] D. E. Koditschek and E. Rimon, “Robot navigation functions on manifolds with boundary,” *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990.

[45] L. Krick, M. E. Broucke, and B. A. Francis, “Stabilization of infinitesimally rigid formations of multi-robot networks,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 477–482, 2008.

[46] B. S. Smith, J. Wang, and M. B. Egerstedt, “Persistent formation control of multi-robot networks,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 471–476, 2008.

[47] A. Ajorlou, K. Moezzi, A. G. Aghdam, and S. G. Nersesov, “Two-stage time-optimal formation reconfiguration strategy under acceleration and velocity constraints,” in *Proceedings of the 49th IEEE Conference on Decision and Control*, pp. 7455–7460, 2010.

[48] P. Tabuada, G. J. Pappas, and P. Lima, “Feasible formations of multi-agent systems,” in *Proceedings of the American Control Conference*, pp. 56–61, 2001.

[49] Y. Hong, G. Chen, and L. Bushnell, “Distributed observers design for leader-following control of multi-agent networks,” *Automatica*, vol. 44, no. 3, pp. 846–850, 2008.

[50] P. Ogren, M. Egerstedt, and X. Hu, "A control lyapunov function approach to multi-agent coordination," in *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 1150–1155, 2001.

[51] M. Ji, A. Muhammad, and M. Egerstedt, "Leader-based multi-agent coordination: Controllability and optimal control," in *Proceedings of the American Control Conference*, pp. 1358–1363, 2006.

[52] I. R. N. Roland Siegwart, *Intelligent Mobile Robot Navigation.* Massachusetts, USA: MIT Press, 1st ed., 2004.

[53] Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177–1182, 2006.

[54] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[55] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 384–389, 1990.

[56] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.

[57] A. O. Federico Cuesta, *Introduction to Autonomous Mobile Robots*. Springer, 1st ed., 2005.

[58] R. C. Arkin, "Motor schema-based mobile robot navigation," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 92–112, 1989.

[59] F. Giulietti, L. Pollini, and M. Innocenti, "Autonomous formation flight," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 34–44, 2000.

[60] P. R. Chandler, M. Pachter, and S. Rasmussen, "UAV cooperative control," in *Proceedings of the American Control Conference*, pp. 50–55, 2001.

[61] W. R. Randal, W. B. Ella, and M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the American Control Conference*, pp. 1859–1864, 2005.

[62] K. Savla, F. Bullo, and E. Frazzoli, "Traveling salesperson problems for a double integrator," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 788–793, 2009.

[63] P. Panyakeow and M. Mesbahi, "Decentralized deconfliction algorithms for unicycle UAVs," in *Proceedings of the American Control Conference*, pp. 794–799, 2010.

[64] A. Krishnamurthy and R. Preis, "Satellite formation, a mobile sensor network in space," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, 2005.

[65] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley and Sons, 2010.

[66] K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. John Wiley and Sons, 2007.

[67] R. E. Fenton and R. J. Mayhan, "Automated highway studies at the ohio state university-an overview," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 100–113, 1991.

[68] D. N. Godbole, V. Hagenmeyer, R. Sengupta, and D. Swaroop, "Design of emergency manoeuvres for automated highway system: obstacle avoidance problem," in *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 4774–4779, 1997.

[69] R. Horowitz and P. Varaiya, "Control design of an automated highway system," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 913–925, 2000.

[70] J. K. Hedrick, M. Tomizuka, and P. Varaiya, "Control issues in automated highway systems," *IEEE Control Systems Magazine*, vol. 14, no. 6, pp. 21–32, 1994.

[71] H.-S. Tan, R. Rajamani, and W.-B. Zhang, "Demonstration of an automated highway platoon system," in *Proceedings of the American Control Conference*, pp. 1823–1827, 1998.

[72] D. N. Godbole, J. Lygeros, E. Singh, A. Deshpande, and A. E. Lindsey, “Communication protocols for a fault-tolerant automated highway system,” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 5, pp. 787–800, 2000.

[73] G. A. Gilbert, *Air Traffic Control*. Chicago, New York: Ziff-Davis Publishing Company, 1945.

[74] J. Shamma, *Cooperative Control of Distributed Multi-Agent Systems*. John Wiley and Sons, 2008.

[75] H. Bai and M. Arcak, “Instability mechanisms in cooperative control,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 357–362, 2008.

[76] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multi-Agent Networks*. Princeton University Press, 2010.

[77] M. Zhong and C. G. Cassandras, “Asynchronous distributed optimization with minimal communication,” in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 363–368, 2008.

[78] M. Arcak, “Passivity as a design tool for group coordination,” *IEEE Transactions on Automatic Control*, vol. 350, no. 5, pp. 1380–1390, 2006.

[79] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics and Automation*, vol. 23, no. 4, pp. 693–703, 2007.

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

[81] H. Su, X. Wang, and G. Chen, "Rendezvous of multiple mobile agents with preserved network connectivity," *Automatica*, vol. 59, no. 5, pp. 313–322, 2010.

[82] T. Gustavi, D. V. Dimarogonas, M. Egerstedt, and X. Hu, "Sufficient conditions for connectivity maintenance and rendezvous in leader-follower networks," *Automatica*, vol. 46, no. 1, pp. 133–139, 2010.

[83] D. V. Dimarogonas, T. Gustavi, M. Egerstedt, and X. Hu, "On the number of leaders needed to ensure network connectivity," in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 1797–1802, 2008.

[84] F. Salehisadaghiani, A. Ajorlou, and A. G. Aghdam, "Distributed connectivity preservation of a team of single integrator agents subject to measurement error," in *Proceedings of the IEEE Multi-Conference on Systems and Control*, pp. 626–631, 2011.

Appendix A

Appendix

A.1 MATLAB Simulation Codes

The simulations in this thesis are obtained using MATLAB version 7.2.0.232 $(R2006a)$ Service Pack 2 from Mathworks Inc. The MATLAB codes which are used in Chapters 3 and 4 are provided in Sections A.1.1 and A.1.2, respectively.

A.1.1 Connectivity Preservation of a Multi-agent System subject to Measurement Error

```
function f1=f1(x,y)

f1=(y+y^2)/(x+y+y^2)^2;

%%
```

```
function f2=f2(x,y)

f2=(-(2*y+1)*(x))/(x+y+y^2)^2;

%%

function pcomp=pcomp(A,d,q,i,j);

sA=size(A);

n=sA(1);

%

p=1;

for k=1:n

    if A(i,k)~=0 & k~=j

        p=p*((d-0.1)^2-(norm(q(i,:)-q(k,:))-0.1*(1-exp(-norm

         (q(i,:)-q(k,:))))))^2);

    end

end

pcomp=p;

%%

function ppcomp=ppcomp(A,d,q,i);

sA=size(A);

n=sA(1);

%

p=1;

for k=1:n
```

```
    if A(i,k)~=0
        p=p*((d-0.1)^2-(norm(q(i,:)-q(k,:))-0.1*(1-exp(-norm
        (q(i,:)-q(k,:)))))^2);
    end
end
ppcomp=p/2;
%%
function sigma=sigmacomp(A,q,i);
sA=size(A);
n=sA(1);
%
s=0;
for k=1:n
    if A(i,k)~=0
        s=s+(norm(q(i,:)-q(k,:))-0.1*(1-exp(-norm(q(i,:)-
        q(k,:)))))^2;
    end
end
sigma=s/2;
%%
function cr2=cr2(t,y);
%%
```

```
n=y(1,:);

d=y(2,:);

%%

for k=1:n

    q(k,:)=(y(2*k+1:2*k+2,:))';

end

for k=1:n

    A(k,:)=(y(n*(k-1)+2*n+3:n*k+2*n+2,:))';

end

%%

for i=1:n

    sq=0;

    sigma(i)=sigmacomp(A,q,i);

    pp(i)=ppcomp(A,d,q,i);

    for j=1:n

    if A(i,j)~=0

        i;

        j;

        p(i,j)=pcomp(A,d,q,i,j);

        b(i,j)=f1(sigma(i),pp(i))-f2(sigma(i),pp(i))*p(i,j);

        st=b(i,j)*((q(i,:)-q(j,:))'+(0.1^2)*(q(i,:)-q(j,:))'*

        (exp(-norm(q(i,:)-q(j,:)))-exp(-2*norm(q(i,:)-q(j,:)))))/
        (norm(q(i,:)-q(j,:)))-0.1*(q(i,:)-q(j,:))’);
        sq=sq+st;
    end
    end
    q;
    cr2(i*2+1:i*2+2,:)=-sq;

end
    b;
%%
cr2(1,:)=0;
cr2(2,:)=0;
cr2(2*n+3:n*n+2*n+2,:)=zeros(n*n,1);
%%
%
close all
clear all
clc
hold on
%%
A=[0 1 1 1;
   1 0 0 0;
   1 0 0 1;
   1 0 1 0];
%%
theta=pi/4;
q0(1,:)=[0;0]';
q0(2,:)=[-0.9;0]';
q0(3,:)=[0.9;0]';
q0(4,:)=0.9*[cos(theta);sin(theta)]';

%%
d=1;
%%
Tfinal=10;
%%
sA=size(A);
n=sA(1);
%%
y0(1,:)=n;
y0(2,:)=d;
%%
for k=1:n
    y0(2*k+1:2*k+2,:)=(q0(k,:))';
```

```
end
for k=1:n
    y0(n*(k-1)+2*n+3:n*k+2*n+2,:)=(A(k,:))’;
end
[tout,yout]=ode45(@cr2,[0 Tfinal],y0);
s=[’b’ ’r’ ’g’ ’k’ ’m’ ’c’ ’y’];
s2=[’1’ ’2’ ’3’ ’4’ ’5’ ’6’ ’7’];
for i=1:n
    text(yout(1,2*i+1),yout(1,2*i+2),s2(i))
    plot(yout(:,2*i+1),yout(:,2*i+2),s(i))
end
axis equal
hold off
% grid on
%
figure
% grid on
hold on
counter=1;
for agenti=1:n
    for agentj=agenti:n
        if A(agenti,agentj)~=0
```

```
            dij1=yout(:,2*agenti+1)-yout(:,2*agentj+1);
            dij2=yout(:,2*agenti+2)-yout(:,2*agentj+2);
            dij=sqrt(dij1.^2+dij2.^2)
            plot(tout,dij,s(counter))

            counter=counter+1;
        end
    end
end
hold off
% grid on
%
kt=length(tout);
for k=1:kt
    vout(k,:)=cr2(tout(k),yout(k,:)’);
end

figure
% grid on
hold on
for indexu=1:n
    voutx=vout(:,indexu*2+1);
    vouty=vout(:,indexu*2+2);

    voutt=sqrt(voutx.^2+vouty.^2);

    plot(tout,voutt,s(indexu))

end

hold off

% grid on
```

A.1.2 Formation Control of a Multi-agent System with Collision Avoidance Scheme and subject to Measurement Error

```
%% Time Evolution Constants

delt=1;

tf=5000;

V=1;

Y=zeros(6,tf/delt);

Y(:,1)=[-24;0;0;34;16;0];

%%

while V<=(tf/delt)

    q=Y(:,V);


%% Betal
```

```
delta=5;

z=2;

m=30;

ep12=z*(1-exp(-(1/m)*(sqrt((q(1)-q(3))^2+(q(2)-q(4))^2))));

ep13=z*(1-exp(-(1/m)*(sqrt((q(1)-q(5))^2+(q(2)-q(6))^2))));

ep23=z*(1-exp(-(1/m)*(sqrt((q(3)-q(5))^2+(q(4)-q(6))^2))));

R12=sqrt((q(1)-q(3))^2+(q(2)-q(4))^2)-ep12;

R13=sqrt((q(1)-q(5))^2+(q(2)-q(6))^2)-ep13;

R23=sqrt((q(3)-q(5))^2+(q(4)-q(6))^2)-ep23;

ep10=z*(1-exp(-(1/m)*(sqrt((q(1))^2+(q(2))^2))));

ep20=z*(1-exp(-(1/m)*(sqrt((q(3))^2+(q(4))^2))));

ep30=z*(1-exp(-(1/m)*(sqrt((q(5))^2+(q(6))^2))));

R10=sqrt((q(1))^2+(q(2))^2)-ep10;

R20=sqrt((q(3))^2+(q(4))^2)-ep20;

R30=sqrt((q(5))^2+(q(6))^2)-ep30;

Betal(1)=1/(1+exp(-3*(R12-2.5)));

Betal(2)=1/(1+exp(-3*(R13-2.5)));

Betal(3)=1/(1+exp(-3*(R23-2.5)));

Betal(4)=1/(1+exp(-3*(R10-2.5)));

Betal(5)=1/(1+exp(-3*(R20-2.5)));

Betal(6)=1/(1+exp(-3*(R30-2.5)));

%% Betaw
```

```
Rw=60;
ep1=z*(1-exp(-(1/m)*(sqrt((q(1))^2+(q(2))^2))));
ep2=z*(1-exp(-(1/m)*(sqrt((q(3))^2+(q(4))^2))));
ep3=z*(1-exp(-(1/m)*(sqrt((q(5))^2+(q(6))^2))));
R1=sqrt((q(1))^2+(q(2))^2)-ep1;
R2=sqrt((q(3))^2+(q(4))^2)-ep2;
R3=sqrt((q(5))^2+(q(6))^2)-ep3;
Betaw(1)=1/(1+exp(3*(R1-57.5)));
Betaw(2)=1/(1+exp(3*(R2-57.5)));
Betaw(3)=1/(1+exp(3*(R3-57.5)));
%% Betapi
Betapi(1)=Betal(1)*Betal(2)*Betal(4)*Betaw(1);
Betapi(2)=Betal(1)*Betal(3)*Betal(5)*Betaw(2);
Betapi(3)=Betal(2)*Betal(3)*Betal(6)*Betaw(3);
%% GBetal
R12el=sqrt((q(1)-q(3))^2+(q(2)-q(4))^2);
R13el=sqrt((q(1)-q(5))^2+(q(2)-q(6))^2);
R23el=sqrt((q(3)-q(5))^2+(q(4)-q(6))^2);
R10el=sqrt((q(1))^2+(q(2))^2);
R20el=sqrt((q(3))^2+(q(4))^2);
R30el=sqrt((q(5))^2+(q(6))^2);
GBetal(1)=((3*exp(-3*(R12-2.5)))/(1+exp(-3*(R12-2.5)))^2)*
```

```
(1-(z/m)*exp(-R12el/m))*(q(1)-q(3))/R12el;

GBetal(2)=((3*exp(-3*(R12-2.5)))/(1+exp(-3*(R12-2.5)))^2)*

(1-(z/m)*exp(-R12el/m))*(q(2)-q(4))/R12el;

GBetal(3)=((3*exp(-3*(R13-2.5)))/(1+exp(-3*(R13-2.5)))^2)*

(1-(z/m)*exp(-R13el/m))*(q(1)-q(5))/R13el;

GBetal(4)=((3*exp(-3*(R13-2.5)))/(1+exp(-3*(R13-2.5)))^2)*

(1-(z/m)*exp(-R13el/m))*(q(2)-q(6))/R13el;

GBetal(5)=((3*exp(-3*(R23-2.5)))/(1+exp(-3*(R23-2.5)))^2)*

(1-(z/m)*exp(-R23el/m))*(q(3)-q(5))/R23el;

GBetal(6)=((3*exp(-3*(R23-2.5)))/(1+exp(-3*(R23-2.5)))^2)*

(1-(z/m)*exp(-R23el/m))*(q(4)-q(6))/R23el;

GBetal(7)=((3*exp(-3*(R10-2.5)))/(1+exp(-3*(R10-2.5)))^2)*

(1-(z/m)*exp(-R10el/m))*q(1)/R10el;

GBetal(8)=((3*exp(-3*(R10-2.5)))/(1+exp(-3*(R10-2.5)))^2)*

(1-(z/m)*exp(-R10el/m))*q(2)/R10el;

GBetal(9)=((3*exp(-3*(R20-2.5)))/(1+exp(-3*(R20-2.5)))^2)*

(1-(z/m)*exp(-R20el/m))*q(3)/R20el;

GBetal(10)=((3*exp(-3*(R20-2.5)))/(1+exp(-3*(R20-2.5)))^2)*

(1-(z/m)*exp(-R20el/m))*q(4)/R20el;

GBetal(11)=((3*exp(-3*(R30-2.5)))/(1+exp(-3*(R30-2.5)))^2)*

(1-(z/m)*exp(-R30el/m))*q(5)/R30el;

GBetal(12)=((3*exp(-3*(R30-2.5)))/(1+exp(-3*(R30-2.5)))^2)*
```

```
(1-(z/m)*exp(-R30el/m))*q(6)/R30el;

%% GBetaw

R1el=sqrt((q(1))^2+(q(2))^2);

R2el=sqrt((q(3))^2+(q(4))^2);

R3el=sqrt((q(5))^2+(q(6))^2);

GBetaw(1)=((-3*exp(3*(R1-57.5)))/(1+exp(3*(R1-57.5)))^2)*(1-(z/m)*

exp(-R1el/m))*q(1)/R1el;

GBetaw(2)=((-3*exp(3*(R1-57.5)))/(1+exp(3*(R1-57.5)))^2)*(1-(z/m)*

exp(-R1el/m))*q(2)/R1el;

GBetaw(3)=((-3*exp(3*(R2-57.5)))/(1+exp(3*(R2-57.5)))^2)*(1-(z/m)*

exp(-R2el/m))*q(3)/R2el;

GBetaw(4)=((-3*exp(3*(R2-57.5)))/(1+exp(3*(R2-57.5)))^2)*(1-(z/m)*

exp(-R2el/m))*q(4)/R2el;

GBetaw(5)=((-3*exp(3*(R3-57.5)))/(1+exp(3*(R3-57.5)))^2)*(1-(z/m)*

exp(-R3el/m))*q(5)/R3el;

GBetaw(6)=((-3*exp(3*(R3-57.5)))/(1+exp(3*(R3-57.5)))^2)*(1-(z/m)*

exp(-R3el/m))*q(6)/R3el;

%% GBetapi

GBetapi(1)=GBetal(1)*Betal(2)*Betal(4)*Betaw(1)+Betal(1)*

GBetal(3)*Betal(4)*Betaw(1)+Betal(1)*Betal(2)*GBetal(7)*Betaw(1)+

Betal(1)*Betal(2)*Betal(4)*GBetaw(1);

GBetapi(2)=GBetal(2)*Betal(2)*Betal(4)*Betaw(1)+Betal(1)*
```

```
GBetal(4)*Betal(4)*Betaw(1)+Betal(1)*Betal(2)*GBetal(8)*Betaw(1)+
Betal(1)*Betal(2)*Betal(4)*GBetaw(2);
GBetapi(3)=-GBetal(1)*Betal(3)*Betal(5)*Betaw(2)+Betal(1)*
GBetal(5)*Betal(5)*Betaw(2)+Betal(1)*Betal(3)*GBetal(9)*Betaw(2)+
Betal(1)*Betal(3)*Betal(5)*GBetaw(3);
GBetapi(4)=-GBetal(2)*Betal(3)*Betal(5)*Betaw(2)+Betal(1)*
GBetal(6)*Betal(5)*Betaw(2)+Betal(1)*Betal(3)*GBetal(10)*Betaw(2)+
Betal(1)*Betal(3)*Betal(5)*GBetaw(4);
GBetapi(5)=-GBetal(3)*Betal(3)*Betal(6)*Betaw(3)-Betal(2)*
GBetal(5)*Betal(6)*Betaw(3)+Betal(2)*Betal(3)*GBetal(11)*Betaw(3)+
Betal(2)*Betal(3)*Betal(6)*GBetaw(5);
GBetapi(6)=-GBetal(4)*Betal(3)*Betal(6)*Betaw(3)-Betal(2)*
GBetal(6)*Betal(6)*Betaw(3)+Betal(2)*Betal(3)*GBetal(12)*Betaw(3)+
Betal(2)*Betal(3)*Betal(6)*GBetaw(6);
%% Gamma
R=40;
c=15;
a=0.00444444444444444;
b=-6.39999997815682e-05;
c1=0.00924444443352286;
d=-0.119999999863480;
e=1;
```

```
if R12<=c
    Gamma(1)=(R12/c-1)^2;
elseif c<R12<=R
    Gamma(1)=a*(R12-c)^2/(b*(R12-c)^3+c1*(R12-c)^2+d*(R12-c)+e);
elseif R12>R
    Gamma(1)=a*(R-c)^2/(b*(R-c)^3+c1*(R-c)^2+d*(R-c)+e);
end;

if R13<=c
    Gamma(2)=(R13/c-1)^2;
elseif c<R13<=R
    Gamma(2)=a*(R13-c)^2/(b*(R13-c)^3+c1*(R13-c)^2+d*(R13-c)+e);
elseif R13>R
    Gamma(2)=a*(R-c)^2/(b*(R-c)^3+c1*(R-c)^2+d*(R-c)+e);
end;

if R23<=c
    Gamma(3)=(R23/c-1)^2;
elseif c<R23<=R
    Gamma(3)=a*(R23-c)^2/(b*(R23-c)^3+c1*(R23-c)^2+d*(R23-c)+e);
elseif R23>R
    Gamma(3)=a*(R-c)^2/(b*(R-c)^3+c1*(R-c)^2+d*(R-c)+e);
```

```
end;
%% GGamma


if R12<=c
    GGamma(1)=(2/c^2)*(R12-c)*(1-(z/m)*exp(-R12el/m))*
    (q(1)-q(3))/R12el;
elseif c<R12<=R
    GGamma(1)=((a*(R12-c)*(-b*(R12-c)^3+d*(R12-c)+2*e))/
    (e + d*(R12 - c)+b*(R12 - c)^3 + c1*(R12 - c)^2)^2)*
    (1-(z/m)*exp(-R12el/m))*(q(1)-q(3))/R12el;
elseif R12>R
    GGamma(1)=0;
end;


if R12<=c
    GGamma(2)=(2/c^2)*(R12-c)*(1-(z/m)*exp(-R12el/m))*
    (q(2)-q(4))/R12el;
elseif c<R12<=R
    GGamma(2)=((a*(R12-c)*(-b*(R12-c)^3+d*(R12-c)+2*e))/
    (e + d*(R12 - c)+b*(R12 - c)^3 + c1*(R12 - c)^2)^2)*
    (1-(z/m)*exp(-R12el/m))*(q(2)-q(4))/R12el;
elseif R12>R
```

```
    GGamma(2)=0;
end;

if R13<=c
    GGamma(3)=(2/c^2)*(R13-c)*(1-(z/m)*exp(-R13el/m))*
    (q(1)-q(5))/R13el;
elseif c<R13<=R
    GGamma(3)=((a*(R13-c)*(-b*(R13-c)^3+d*(R13-c)+2*e))/
    (e + d*(R13 - c)+b*(R13 - c)^3 + c1*(R13 - c)^2)^2)*
    (1-(z/m)*exp(-R13el/m))*(q(1)-q(5))/R13el;
elseif R13>R
    GGamma(3)=0;
end;

if R13<=c
    GGamma(4)=(2/c^2)*(R13-c)*(1-(z/m)*exp(-R13el/m))*
    (q(2)-q(6))/R13el;
elseif c<R13<=R
    GGamma(4)=((a*(R13-c)*(-b*(R13-c)^3+d*(R13-c)+2*e))/
    (e + d*(R13 - c)+b*(R13 - c)^3 + c1*(R13 - c)^2)^2)*
    (1-(z/m)*exp(-R13el/m))*(q(2)-q(6))/R13el;
elseif R13>R
```

```
    GGamma(4)=0;
end;

if R23<=c
    GGamma(5)=(2/c^2)*(R23-c)*(1-(z/m)*exp(-R23el/m))*
    (q(3)-q(5))/R23el;
elseif c<R23<=R
    GGamma(5)=((a*(R23-c)*(-b*(R23-c)^3+d*(R23-c)+2*e))/
    (e + d*(R23 - c)+b*(R23 - c)^3 + c1*(R23 - c)^2)^2)*
    (1-(z/m)*exp(-R23el/m))*(q(3)-q(5))/R23el;
elseif R23>R
    GGamma(5)=0;
end;

if R23<=c
    GGamma(6)=(2/c^2)*(R23-c)*(1-(z/m)*exp(-R23el/m))*
    (q(4)-q(6))/R23el;
elseif c<R23<=R
    GGamma(6)=((a*(R23-c)*(-b*(R23-c)^3+d*(R23-c)+2*e))/
    (e + d*(R23 - c)+b*(R23 - c)^3 + c1*(R23 - c)^2)^2)*
    (1-(z/m)*exp(-R23el/m))*(q(4)-q(6))/R23el;
elseif R23>R
```

```
    GGamma(6)=0;
end;

%% Gammasum
Gammasum(1)=Gamma(1)+Gamma(2);
Gammasum(2)=Gamma(1)+Gamma(3);
Gammasum(3)=Gamma(2)+Gamma(3);

%% GGammasum
GGammasum(1)=GGamma(1)+GGamma(3);
GGammasum(2)=GGamma(2)+GGamma(4);
GGammasum(3)=-GGamma(1)+GGamma(5);
GGammasum(4)=-GGamma(2)+GGamma(6);
GGammasum(5)=-GGamma(3)-GGamma(5);
GGammasum(6)=-GGamma(4)-GGamma(6);

%% odefunc

k=1;
odefunc(1)=-(1/(k*(Gammasum(1)^k+Betapi(1)))^(1/k+1))*
(k*Betapi(1)*GGammasum(1)-Gammasum(1)*GBetapi(1));
odefunc(2)=-(1/(k*(Gammasum(1)^k+Betapi(1)))^(1/k+1))*
```

```
(k*Betapi(1)*GGammasum(2)-Gammasum(1)*GBetapi(2));
odefunc(3)=-(1/(k*(Gammasum(2)^k+Betapi(2)))^(1/k+1))*
(k*Betapi(2)*GGammasum(3)-Gammasum(2)*GBetapi(3));
odefunc(4)=-(1/(k*(Gammasum(2)^k+Betapi(2)))^(1/k+1))*
(k*Betapi(2)*GGammasum(4)-Gammasum(2)*GBetapi(4));
odefunc(5)=-(1/(k*(Gammasum(3)^k+Betapi(3)))^(1/k+1))*
(k*Betapi(3)*GGammasum(5)-Gammasum(3)*GBetapi(5));
odefunc(6)=-(1/(k*(Gammasum(3)^k+Betapi(3)))^(1/k+1))*
(k*Betapi(3)*GGammasum(6)-Gammasum(3)*GBetapi(6));
odefunc=[odefunc(1);odefunc(2);odefunc(3);odefunc(4);
    odefunc(5);odefunc(6)];
%%
   Y(:,V+1)=Y(:,V)+delt*3*odefunc;
   V=V+1;
end;
%% PLOT
figure(1);
hold on
axis equal
s=[’1’ ’2’ ’3’];
cl=[’r’ ’b’ ’g’];
for i=1:3
```

```
    text(Y(2*i-1,1)-0.15,Y(2*i,1)-0.15,s(i),

    'fontweight','b','Fontsize',10)

    n=length(Y(2*i-1,:));

    for j=1:n

        plot(Y(2*i-1,j),Y(2*i,j),cl(i));

    end;

end;

for w=0:0.01:2*pi

    plot(Rw*cos(w),Rw*sin(w),'b');

end;

for w=0:0.01:2*pi

    plot((Rw-5)*cos(w),(Rw-5)*sin(w),'b');

end;

for w=0:0.01:2*pi

    plot(5*cos(w),5*sin(w),'b');

end;

plot(0,0,'*')


plot(Y(1,1),Y(2,1),'r*')

plot(Y(3,1),Y(4,1),'b*')

plot(Y(5,1),Y(6,1),'g*')

plot(Y(1,tf/delt),Y(2,tf/delt),'rdiamond')
plot(Y(3,tf/delt),Y(4,tf/delt),'bdiamond')
plot(Y(5,tf/delt),Y(6,tf/delt),'gdiamond')
hold off;

figure(2);
hold on
for i=1:2
    for j=i+1:3
        plot((0:delt:delt*(n-1)),sqrt((Y(2*i-1,:)-
        Y(2*j-1,:)).^2+(Y(2*i,:)-Y(2*j,:)).^2),cl(i+j-2));

    end;
end;
hold off;
%%
```