LEADER-FOLLOWING CONSENSUS OF MULTI-AGENT SYSTEM WITH COMMUNICATION CONSTRAINTS USING LYAPUNOV BASED CONTROL

by

Ajinkya Pawar

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
December 2016

Just as a house needs a foundation in order to stand firm, so does a person. Thank you mom, dad and my sister Juhi for your firm support.

Table of Contents

List of Tables

List of Figures

Abstract

Wireless networked control systems possess many challenges to control engineers. These challenges include time delays, packet data dropouts, switching topology, noise, quantization errors. Time delays are almost ubiquitous in almost all the networked control systems and they affect the system performance in a negative way.

The objective of the thesis is to develop a novel control algorithm for achieving leader-following consensus of multi-agent systems (MAS). The topology of the wireless networked MAS is modeled by an algebraic graph theory and defined as a discrete time-invariant system with second order dynamics. The communication link failure is governed by Bernoulli's distribution principle and the time-delays are incorporated in to the system dynamics. Lyapunov-based methodologies and Linear Matrix Inequality (LMI) techniques are then applied to find an appropriate value of the control gain by sufficient conditions of error dynamics.Finally simulation result and experimental result studies are carried out by using two Pioneer P3-DX robots as real follower robots and a virtual leader robot. The results are verified at the end.

List of Abbreviations Used

AFF Autonomous Formation Flight

ARIA Advanced Robot Interface for Applications

LMI Linear Matrix Inequality

LTI Linear time-invariant

MAS Multi-agent system

MSE Mean Square Error

NASA National Aeronautics and Space Administration

TCP/IP Transmission Control Protocol/Internet Protocol

UAV Unmanned Aerial Vehicles

UDP User Datagram Protocol

Acknowledgements

I would primarily like to thank my supervisor, Dr Ya-Jun Pan, for giving me an opportunity to work with her in the research field of Advanced Control and Mechatronics. She has been a great guidance from the start for helping me choose my research interest. Her timely advice and track of my progress has finally helped me cover this mammoth task of thesis work. Its been a long learning curve with ups and down. Without her help and guidance, this all seems to be impossible now. I would also like to thank Dr. Robert Bauer and Dr. Jason Gu for their role in my supervisory committee and for their suggestions. Special thanks to Peter Jones and Kate Hide for their timely support with technical assistance. And a big thank you to all my colleagues at ACM Lab. Your honest suggestions have definitely help me. Last but not the least, my deepest gratitude to my parents for their unconditional support. Thank you.

Chapter 1

Introduction

In this chapter, an outline of consensus control of multi-agent systems and background of it's research history is presented. It also includes the literature review on the concept of consensus theory of multi-agent systems along with its applications. Finally, a contribution of research work is presented in this chapter.

1.1 Thesis Motivation

An agent is a computational entity that can sense and act as well decide on its actions in accordance with some assigned tasks or goals. A multi-agent system (MAS) is a specific type of system that is composed of multiple intelligent agents that interact with each other to achieve certain objectives. Each of them is autonomous, that means they have an onboard microcontroller to schedule their own tasks. MAS can be used to solve problems that are difficult or impossible for a monolithic or a single agent system to resolve. In recent years, many researchers are working in the field of networked multi-agent systems due to various technological advances in computation and communication. This breakthrough has also led to the development of new MASs, that are able to achieve improved performance, efficiency and robustness.

MAS works on the foundation of cooperative control that coordinates the motion of a group of dynamic agents as shown in Fig.1.1. In coordinating a number of agents, it is easy to implement centralized approach in which the control depends on a single central agent. However for a large network of agents, the centralized approach fails because the central agent is subjected to a huge load of communication and computation. So, it becomes necessary to delegate this working load to other connected agents. This type of approach is called as decentralized implementation of cooperative control.

In a cooperative control of multi-agent system, if the information states of each

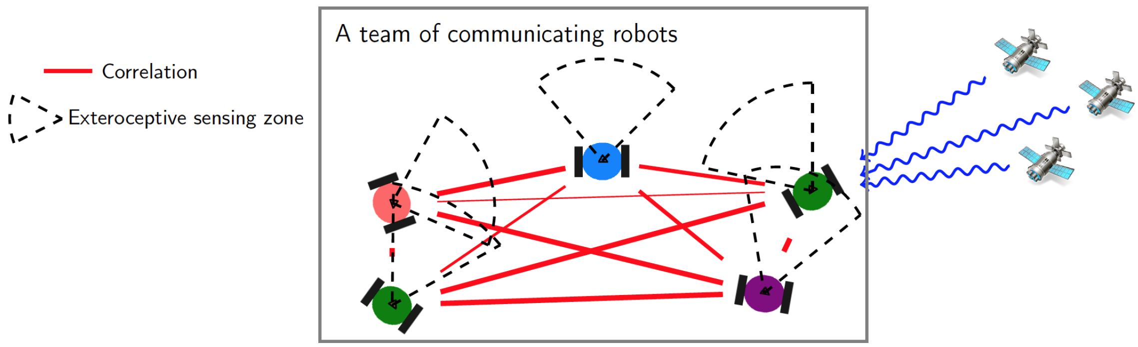

Figure 1.1: A team of communicating agents [8]

agent converge to a common value, then it is said that the MAS has reached a consensus. Leader-following consensus is a special type of consensus in which there is a single leader agent and multiple follower agents. The leader agent sends it's information states to the follower agents and also receives back the information states of the follower agents. It then calculates the difference or error between the information states and then tries to reduce that error gradually. At the same time, the follower agents also try to reduce the error of information states between them.

Using wireless networks as the communication media in MAS have broaden the scope of it's capabilities. However, wireless network communication have a variety of constraints that pose as great challenges to control engineers. These include time delays, packet data dropouts, switching topologies, noise, quantization errors etc. Time delays are ubiquitous in almost all networked control systems due to limited data communication speeds or narrow bandwidths as well as measurement, computation and execution time required for control actions. Time delays influence the system performance as they are largely dependent on both the quantity of delays and the properties of the system dynamics. Packet data dropouts are also a main concern in networked control system. Sometimes data packets cannot be sent and used at the same time because the controller needs time to process them. Time delay is mainly the time lag between the transmitted packet data from one agent and received packet data at the other agent. So there is a need to develop a system dynamics for the MAS such that it corporate both the time delays and packet data dropouts.

There have been considerable results published in some literature based on consensus of single-integrator MAS dynamics with communication delays. However when it comes to more complex systems, it is not enough to employ the single-integrator dynamics. So it is necessary to consider double-integrator dynamics for studying consensus of MAS. In modern era, with the introduction of digital communication technology and digital signal processing, discrete time systems become prevalent. To cope with this situation, the continuous-time system dynamics is transformed to discrete-time dynamics and then asymptotic properties of the corresponding discrete-time dynamics is studied.

Mainly, there are three theories involved in the research on consensus. They are control systems theory, matrix theory and graph theory. These theories are combined together to design a controller for MAS dynamics to achieve consensus. Researchers have used many control systems theories like Nyquist criterion, sliding mode control, model predictive control, neural network control, passivity based control. However the most suitable and reliable control theory for consensus of discrete-time MAS is found to be Lyapunov based control. It is a popular way for stability analysis after establishing the associated error dynamics.

Consensus of MAS has it's applications in many fields. The most obvious and popular one is rendezvous, in which the states of all agents converge to a common value. Leader-following consensus is a type of rendezvous application. The second most popular application is formation control, in which all agents maneuver in certain shape formation. In addition to the control field, consensus theory can be employed for filtering as well. This type of filtering is in a distributed way and is called consensus filtering. Due to it's inherently distributed feature, consensus approach has been used for decentralized parameter estimation, thus enhancing robustness of the overall system against node failure.

Motivated by the above discussion, the thesis work is based on achieving leader-following consensus of MAS in the discrete-time domain. Communication constraints like time-delays and packet data dropouts are also considered in system dynamics. A Lyapunov control theory is implemented to design a controller to achieve consensus of MAS. Linear matrix inequalities (LMIs) are derived from substituting error dynamics in assumed Lyapunov function and are solved by using MATLAB® LMI

Solver Toolbox. LMI techniques have emerged as powerful design tools in control engineering. A variety of design specifications and constraints can be expressed as LMIs. Once formulated in terms of LMIs, a problem can be solved exactly by efficient convex optimization algorithms.

1.2 An Overview of Consensus Problem and Communication Constraints in Multi-Agent Systems

In this section, the history of emergence of cooperative control of MAS and research work in it's early days is presented. Various literature works based on consensus theory along with the challenges associated with it are discussed and their contributions in developing this technology is highlighted. This section gives an idea about most of the problems that are faced when the consensus theory is applied in practice and also the solutions that various researchers have established to tackle these problems.

1.2.1 History and Background

In [1], researchers initially were attempting to model the aggregate motion of different animals such as flock of birds, a herd of land animals or a school of fish. These motion behaviors were studied to generate computer animations. Gradually, they realized that there are many potential engineering applications based on the control of independent systems to perform various collective behaviors. In early days, navigation strategies [2] for multiple robots were worked upon. However with significant technological advances in control methodologies over the past two decades along with simultaneous development of sophisticated communication and signal processing techniques, it became easier for multiple dynamical systems to interact with one another. Such systems came to be known as multi-agent system (MAS), where each agent in the multi-agent system represent an independent dynamical system. Using multi-agent system helps in breaking down the large complex network system into multiple simple systems, increases the possibility of incorporating more functionality in the overall system, and more importantly, reduces costs and risks of failure if a single system is used for the same operation. The control of such multi-agent system to achieve a collective or common goal is known as a cooperative control [3].

With the development of sensor networks, cooperative control in MASs has attracted more attention in recent years, for references see [4] [5] [6] [7]. Cooperative control is classified under two broad methods, namely, centralized control and decentralized control. A centralized control obtains and utilizes all the state information, computes the control signal and relays the relevant control signals to each agent. For a large scale system composed of a network of subsystems, it proves to be more costly in terms of communication of information for the system. Another disadvantage of centralized control is that it is not possible to include more subsystems into the existing system once the control has been designed. That means centralized control methods are not expandable. To avoid all these problems, decentralized control techniques have been proposed where controllers are developed for the subsystems of a large system. Various decentralized control techniques [8] [9] [10] [11] have been developed over last few decades.

1.2.2 Literature Review on Consensus and Associated Communication Constraints

The term *consensus* comes from Latin word *consentire* which means a general agreement made by all or majority. In [12], the authors have explained the agreement algorithm in terms of parallel computation, distributed optimization and signal processing. The research on consensus increased after the authors in [4] could successfully demonstrate the theoretical explanation of Vicsek model in [13]. Vicsek et al. proposed a simple but compelling discrete-time model of n autonomous agents all moving in same plane with the same speed but with different headings. The work in [4] opened a new way to achieve consensus by using algebraic graphs and necessary conditions on graph connectivity. These graphs characterize the interaction among agents. In [5], a direct connection between the algebraic connectivity (or Fiedler eigenvalue) of the network and the performance of a linear consensus protocol is established. The authors in [5] have discussed average consensus problems in a directed networks, considering switching topology and constant time delay as well. Methodology used in [4] is different than the use of common Lyapunov function in [5]. However both [4] and [5] have made an important contribution to the consensus problem in terms of graph connectivity. There were some limitations still left in [4] and [5] such as undirected graphs in [4] and directed and strongly connected graphs in [5]. These limitations were worked upon by studying the properties of general directed graphs in [6] in which Ren and Beard have stated a necessary and sufficient condition between the connectivity of a directed graph and the eigenvalues of its corresponding graph Laplacian matrix. They also stated that there is necessity of existence of directed spanning tree for reaching consensus in linear time-invariant single integrator agent system. In [14] and [15], similar conclusions were drawn on directed graphs, which helped the subsequent research which was largely based on the interaction topology.

Earlier research was mainly focused on the simple single-integrator dynamics, but as the time and control technology progressed, there was a need of paying more attention to general dynamics such as double-integrator dynamics [16] [17], state space models [18] [19] and Euler-Lagrange dynamics [20]. many control methods have been extensively studied in control field to solve the consensus problem in multi-agent system like the Lyapunov control theory in [21] [22] [23] [24] [25] [26] [27], sliding mode control in [28] [29] [30] [31], model predictive control in [32] [33] and neural network based control in [34]. If we assume LTI systems without any constraints, then the problem becomes very straight forward, which is not the case in real time conditions. There are many type of communication constraints which hamper the performance of control systems like time delays, packet data losses, quantization, switching topology, asynchronous sampling, etc. Among them, the major concerned problems are that of time delays and packet data losses which affect the system behavior considerably and may result to system failure in some cases. Olfati-Saber and Murray [5] studied the average consensus in a single integrator system with and without constant time delays for directed and undirected networks. The authors have also considered these cases for fixed and switching topology. The frequency domain approach was used to derive the stability condition dependent on the upper bound of delay. This research proved to be a breakthrough for considering constraints in consensus of MAS.

Afterwards, many researchers expanded their study in communication constraints experienced in consensus problem like time-varying delays and switching topology in [35] [36] for first-order dynamics. Later in [37], consensus problem in double-integrator dynamics with non-uniform time-varying delays was studied. For tackling time-varying delays, it is necessary to shift the system dynamics from frequency domain to time domain, for which the Lyapunov technique is widely used. For the purpose of analysis, time delays can be classified as constant time delays or time-varying delays. Constant delays are considered in a case where the values of time delays at all instants of time fluctuate near to its average value [38] [36]. For uncertain time-varying delays, the upper and lower bound of delays are considered [39] [40]. Apart from delays, packet data losses or dropouts have been a major concern experienced in control systems. Packet losses can affect the system performance badly and can slow down the entire working of the system. They occur as a result of unreliability of communication channels between two agents. There is a relation between the packet delay threshold and packet dropout such that when a quantity of delay exceeds the threshold, this long delay is treated as packet data dropout or loss. Switching topology is the commonly experienced effect as a result of packet data loss which is widely studied in [41] [42] [43]. In practice, communication channels often exhibit a corresponding probability of failure for each communication channel transmission. These probabilities of the availability of communication links can be modeled into analytical form and convergence results can be established as shown in [41] [44] [45]. The authors in [41] considered packet data loss and time delays simultaneously, however they had assumed that the delay was less than a sampling period. This assumption may not guarantee the consensus completely for non-predictive nature of packet losses. However, in regular working of control systems, it's been always observed that the maximum values of time delays are well below the sampling period and even if there is packet loss at some time instant, the next packet is received instantaneously or within a time delay of lesser value than sampling period of packet. The system based on this assumption will fail only if there is permanent discontinuity of packet transfer in communication channel, for which there is nothing much that can be done and hence this fear of instability will always exist. In addition to above mentioned communication constraints, there are many other constraints such as quantization errors, asynchronous clocking and noise measurements etc.

1.3 Control Theories and Approaches

The most commonly used theories involved in the research on consensus of MAS can be classified in following three categories as shown in Fig. 1.2.

- **Control Systems Theory.** There are many control systems theories that can be applied in the consensus problem like in any other control systems. For LTI systems, frequency-domain based approaches are widely used. The problem becomes more complicated when consensus protocols are extended to systems of second-order agents. In [46], a general frequency-domain framework to study the consensus problem for systems of high-order agents with non-uniform communication delays is presented. Using the spectral radius theorem, they obtained a frequency-domain consensus condition which is independent of the communication delays. Similarly in [47], the decentralized consensus conditions of second-order multi-agent systems with diverse input delays and symmetric coupling weights are obtained based on the frequency-domain analysis. In [5], a necessary and sufficient condition on upper bound of delay was derived using Nyquist criterion. Further, in [47], analysis of the discrete-time in the frequency domain.

  For time-varying systems,the Lyapunov method is a popular method adapted for stability analysis by establishing the associated error dynamics [5]. In [48], Lyapunov control approach is used to prove that the second-order consensus can be reached if the general algebraic connectivity of the communication topology is larger than a threshold value. In the category of Lyapunov type methods, the concept of vector Lyapunov function has been widely used for stabilizing control design. The subsystems linked through an interconnected network have associated Lyapunov functions, which are grouped in a vector Lyapunov function through which the stability of the overall system is analyzed by taking into account the interconnection terms. This analysis is performed by constructing a scalar Lyapunov function on the basis of the components of the vector Lyapunov function [49] [21]. In addition to it, the sliding mode control [28] [29] [30] [31], the model predictive control [32] [33], neural networks control [34] and passivity based control are some of the other control systems theories that are

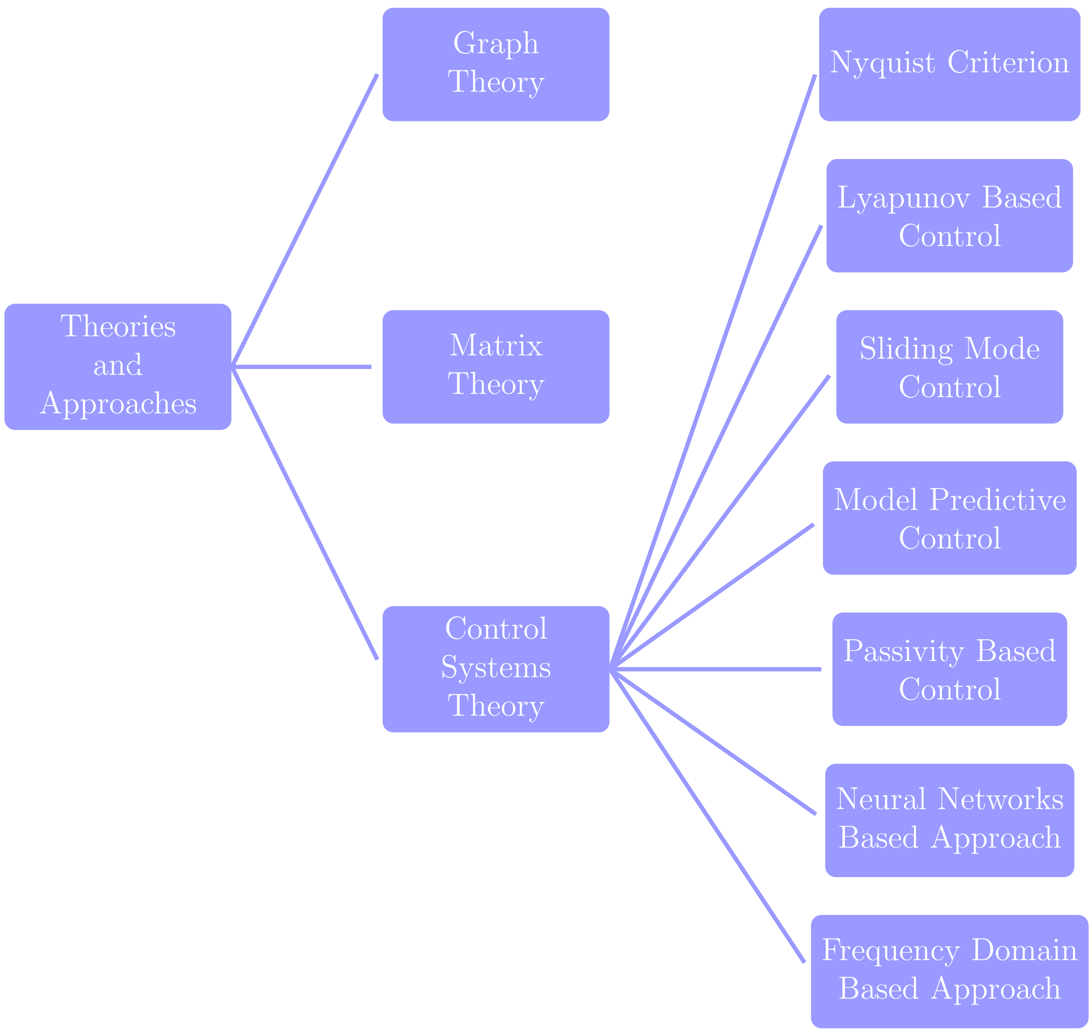

Figure 1.2: Classification of control theories and approaches

used for analyzing the stability of the system.

- **Matrix Theory.** Primary research on consensus of multi-agent systems is mostly dependent on Matrix Theory. Instead of using control algorithms, different properties of various matrices were used to predict the stability of the entire system. Jadbabaie *et al.* [4] introduced the theory of stochastic matrices in consensus problem. Its analysis was based on the asymptotic properties of the product of an infinite sequence of stochastic matrices [50]. In [51], the authors have used graph Laplacians and their spectral properties [52] [53] [54] [55] to prove that graph-related matrices play a crucial role in convergence analysis of consensus and alignment algorithms.

- **Algebraic Graph Theory.** Algebraic graph theory is a branch of mathematics that studies graphs by using algebraic properties. There are two main connections between graph theory and algebra. These arise from two algebraic objects associated with a graph : its adjacency matrix and its automorphism group. More details about algebraic graph theory is discussed in Chapter 2.

1.4 Applications of Consensus of MAS

In this section, a summary of many applications for cooperative control of multi-agent or multi-vehicle systems is presented. Different problems that involve interconnection of dynamic systems in various areas of science and engineering happen to be closely related to consensus problems of multi-agent system. The research on consensus finds its applications in many fields. The most obvious one is rendezvous, in which states of all agents converge to a common value. This is desirable if all agents are expected to meet at a certain location after a finite time. In [56], consensus seeking ideas are applied to a rendezvous problem for a group of mobile autonomous agents, where both the synchronous and asynchronous case are considered. Rendezvous in space [57] is another common form of consensus problems. This is equivalent to reaching a consensus in position by a number of agents with an interaction topology that is position induced. Another reference that can be provided is [58], wherein this type of rendezvous is described as an unconstrained consensus problem that becomes

Figure 1.3: Two F/A 18 aircrafts flying in formation

challenging under variations in the network topology. Depending on the application, the rendezvous time may either be fixed ahead of time or determined dynamically, based on when all vehicles reach the same area. Military applications of rendezvous include minimizing exposure to radar by allowing aircraft to fly individual path that are locally optimized [59].

The other application that has gained a lot of attention in recent years is formation control problems [60] [61] [62] [63] [64]. In [65], a detailed idea about information techniques are studied to improve stability margins and vehicle formation performance. Formation flight is one of the simplest cooperative control control problems in which a set of aircraft fly in a formation, specified by relative locations of nearby aircraft. Earliest work in this area is found in [66], where design of control laws that use a combination of local and global knowledge to maintain a formation. Also NASA in alliance with Boeing-Phantom Works started a revolutionary concept program called AFF (AFF) which helped reducing drag on a collection of aircraft [67] as shown in Fig. 1.3. The key idea is to locate the aircraft such that the tip vortices of one aircraft help reduce the drag of the tailing aircraft. This task requires precise alignment of an aircraft with the aircraft in front of it.

The development of cooperative rendezvous and cooperative target classification agents in a hierarchical distributed control system for unmanned aerospace vehicles have led to rise of another important application of cooperative classification and surveillance. Chandler et al. [59] define the cooperative classification problem as “the task of optimally and jointly using multiple vehicles’ sightings to maximize the probability of correct target classification.” The cooperative classification problem can be defined as a performance function which involves collection of maximal amounts of relevant information. Whereas cooperative surveillance problem is using a collection of vehicles to maintain a centralized or decentralized description of the state of a geographic area. The description can be related to any entity that is moving or spatially fixed in the region of interest.

In mobile sensor networks, different parts of the control system tend to work separately at their time-scale. However it becomes necessary to coordinate all at same time-interval. In decentralized environment, it is often not possible to have a global node that directly communicates with all other nodes. With the help of consensus theory, clock synchronization can be implemented in distributed systems and render the system more robust against node failures [68]. In addition to the control field, consensus theory can be employed for filtering as well in distributed system which is known to be consensus filtering [69] [70] [71]. There are a group of filter sensor nodes and each of them gives an estimate of the target signal. The goal is to design control protocols such that they not only reach consensus on their own estimates, but also give a good estimate of the target signal.

1.5 Thesis Contribution

The main contribution of the thesis is to develop a novel consensus control algorithm for the leader-following approach of co-operative control of multi-agent systems. This is like making the long story short. However, in achieving this objective, there are several steps that need to be developed to reach the final goal. Several rules, theorems, lemmas and assumptions from different research works are studied and their implementation the considered work is verified. While doing this, there are several assumptions and limitations drawn because of the obstructions in the proper implementation. For this, some simple solutions and mathematical conditions have been imposed to validate the genuinity of work and relating it's application in real-time working conditions.

Starting with the selection of dynamics suitable for the considered multi-agent systems follows the process of developing a relationship between the constraints observed in the system with the dynamics of the system. For example, there is introduction of Bernoulli's distribution principle to compensate for the packet data loss nature in the MAS. Similarly, the relation of constant time delay and time-varying delay on the system dynamics is achieved by targeting the sample event k in the dynamics.

The conversion of the complex form of the system dynamics for n number of agents is needed to be in some simplest form for derivations is necessary. The proper general form is achieved for studying the dynamics for n number of agents which leads to the derivation of general system dynamics.

The derivation of error dynamics for this general system dynamics is necessary to understand the parameters that needed to be eradicated or atleast tried to be reduced as far as possible and to correct errors.

The control objective of any control system is to derive a feedback to the system to reduce these errors simultaneously and make it perfect over a cycle of period. The different ways to control this system were studied and the Lyapunov based methodology to reduce the errors was finally decided to implement that would definitely reduce the error to achieve consensus of position state. However, this methodology does not guarantee the shortest possible time or quickest of the controllers. For the certain constraints within limited values, this methodology helps achieving consensus of the system and is targeted towards all the obstructions.

These Lyapunov based methodologies are to be further reduced to linear matrix inequalities(LMIs) which help find the control gain to the system. The derivation the these LMIs is a total novel algorithm achieved in this thesis work. The LMIs achieved in this thesis case have never been in any published work. Also the system dynamics incorporating the time-varying delays and packet loss based on Bernoulli's distribution principle is hard to notice in many research works.

The control gain achieved is the ultimate requirement to verify the consensus of all the agents. The value of control gain is fed up in the system to check the simulation results for the same. These results have been studied extensively in Chapter 4. The conclusions or observations based on all the communication constraints considered at same time in this type of multi-agent system at different conditions are also novel in some sense to this MAS field. Very few research work is published considering both the packet loss and time delays present to be both in same MAS. The relationship between the time delay and packet loss is considered in different fields of control systems. Same relation is applied in the considered MAS dynamics.

Another important contribution is the experiment conducted by using the hardware available. The MAS system considered for theoretical purpose is double-integrator system. However, in experimental setup, the robots used are single integrator dynamics. The conversion of dynamics to implement the same control gain on the practical system is carried out. The control gain found out by solving LMIs are commonly used for both the dynamics and have been successfully helped in achieving consensus in both cases.

Finally, the simulation results discussed in Chapter 4 and the experimental result in Chapter 5 approved the feasibility and effectiveness of the proposed controller.

1.6 Thesis Outline

The thesis outline is structured as follows. Chapter 1 is more about the introduction to multi-agent systems, control theories related to multi-agent systems and applications of multi-agent systems. Chapter 2 introduces the leader-follower approach of consensus of MAS, algebraic graph theory and its properties as well as introduction of communication constraints and system dynamics. Chapter 3 and Chapter 4 are the main work structure of the thesis. The novelty related to the thesis work is present in both these chapters. Chapter 3 is more about the system dynamics, and its incorporation with communication constraints and their ultimate conversion to general closed loop system for multi number of agents in a wireless networked system. The control ability of this close looped system is generated by designing a controller to tackle all these communication constraints at different conditions. All the mathematical derivations, proofs and assumptions are present in this chapter. Moving ahead to Chapter 4 are the results of the system in a working environment. All results have been presented and properly classified with the help of tables. Chapter 5 is about the introduction of the hardware which is used to perform experimental results. These experimental results are verified with simulation results. A single case of experimental setup is considered which covers all the factors under consideration to validate the simulation results. Chapter 6 discusses about conclusions and future works.

Chapter 2

Leader-following Multi-agent System Modeling with Algebraic Graph Theory and Communication Constraints

In this section, a detailed information about the leader-follower approach consensus is discussed. Also the important algebraic graph theory associated with multi-agent system is explained. Some mandatory assumptions and conditions which relate graph theory to consensus problem are presented and verified.

2.1 Leader-follower Approach

In the leader-follower approach, the agents are differentiated and identified as leaders or followers [72]. The leaders follow preassigned trajectories while each follower tracks the trajectory of it's leader, maintaining a relative distance. The leader and follower agent are at a certain distance with each other. The control objective is to reduce the distance between them by taking into consideration that they do not get obstructed with other agents at any point of time. Distance can be reduced linearly or non-linearly depending on external conditions. Each agent has two position states i.e. relative linear distance l and relative angular orientation ψ in a global co-ordinate frame. In case of n follower agents and one leader agent, the relative linear position of these agents can be expressed as l_{01}, l_{02}, l_{03}, l_{04}, $\ldots$, l_{0n} and the relative angular orientation of these agents can be expressed as ψ_{01}, ψ_{02}, ψ_{03}, ψ_{04}, $\ldots$, ψ_{0n} where '0' denotes the leader agent and $1, 2, 3, 4, \ldots, n$ represent the follower agents. The leader-follower strategy was first used in [73] to develop navigation strategies for multiple autonomous robots moving in a formation. A single leader was identified to specify the desired formation to the following robots. Other followers track the nearest neighbor to move in desired formation. This work was subsequently extended to maintain formations of multiple microspacecraft [72]. In [72], the authors considered a fleet of microspacecraft divided into groups with each group having a leader and each group then tracks nearest neighbor to achieve desired goal. The authors subsequently incorporated adaptive control techniques in the leader following consensus control scheme to achieve formation flying of spacecraft [74]. In [75], adaptive control laws for formation flying of multiple spacecraft were extended to include nonlinear dynamics. In [76] decentralized leader-follower control was designed for two robotic manipulators to achieve the cooperative task of moving a box. In [77], a leader-follower formation control strategy for mobile robots maintaining desired relative distance and orientation was proposed. The authors showed that the leader-follower assignment can be represented as a graph and can be used to mode changes in formation. Leader-follower assignment using graphs has also been used in [78] for formation flying of spacecraft. This approach incorporates ideas from LMIs and controller switching. In [79], the effects of leader behaviour on the errors in formation, defined as the deviation of agents from their desired positions is studied. In this work, error propagation in a leader follower network and methods to improve the safety, robustness and performance of a formation is considered.

The main advantage of the leader-follower strategy is its ease of implementation [80]. Another advantage is that stability of an individual agent implies stability of the formation [65] and the multi-agent system can be coordinated by specifying the trajectory of the leader. This also becomes one disadvantage since the leader becomes the single point of failure. Any failure of the leader to trace its defined trajectory or communication loss from the leader will result in failure of the control strategy to achieve its desired goal. Thus it becomes necessary to dynamically decentralize the system such that at any event of communication failure, the leader can trace back its all followers. However, decentralizing the MAS is quite a hectic and complicated task as it changes the system dynamics as well as communication setup. To ensure efficient working of the leader-follower system,

2.2 Algebraic Graph Theory

Algebraic graphic theory is one of the most fundamental concepts and approaches to completely understand the consensus control of multi-agent systems. The major advantage of graph based control methods is that it becomes easy for analyzing and designing associated control methods, also incorporating various communication topology and scalability. Graph based abstractions of a networked system contain virtually no information about what exactly is shared by the agents, instead it gives an idea about the network connectivity of topology by representing topology objects in terms of nodes and edges. Graph theoretic methods facilitate the design of distributed control algorithms by exploiting the properties of this graph. In this subsection, the methods which have utilized the properties of an algebraic graph to design distributed consensus algorithms is presented.

2.2.1 Graph Laplacian Matrix

Considering the agents as nodes and communication links as the edges, the network of the entire system can be classified into directed graphs, undirected graphs and mixed graphs. Directed graphs can send information data packets only from one node to another node at all time instants, but the undirected (bidirectional) graphs can send information data packets from one node to another node as well as vice-versa at any time instant. There also exists a third type of graphs called as mixed graphs which contains directional and bidirectional communication links in the same network topology. Fig.2.1 displays the communication linkage in directional graphs, bidirectional graphs and mixed graphs. The directed graph has its edges pointing at the one direction, whereas the undirected graph has its edges pointing in bidirection and mixed graph is a combination of both. Each of these graphs have their fair share of advantages and disadvantages. Directed graphs resemble that the information flow exchange is carried between the two nodes, which means in real time, two servers are connected but only one server can send information to another at same time instant. So in this case, the reliability of information exchange is based on the server which sends the information data packets to another. If in any case, if the sending server fails to send these data packets, then the system efficiency might reduce because of packet data loss or in worst case situation it can fail the entire working of the system, which is undesirable. However most of the TCP/IP connection servers that we use to connect Pioneer robots have data transmission limitations such that at any instant of time, it can either send or receive the information data packet from other server. It can never receive and transmit at the same time to another connected server. Hence it becomes necessary to consider the directed graph network topology

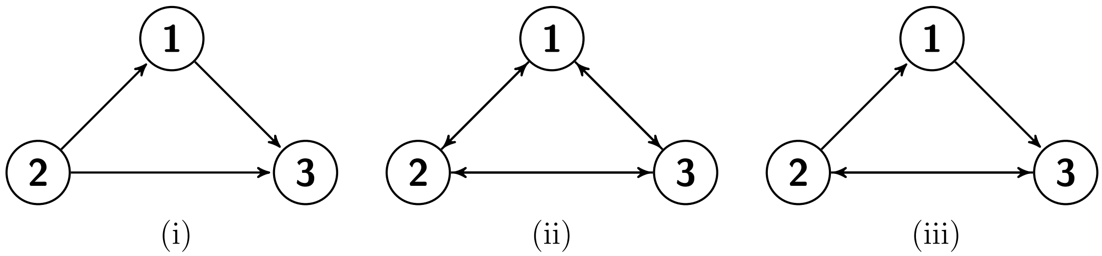

Figure 2.1: (i) Directed graph (ii) Undirected graph (iii)Mixed graph

for the experiment purpose. However this does not affect considering the mixed or undirected graph network topology to obtain simulation results based on controller design of MAS dynamics. Both the directed and undirected graph can be strongly connected based on few assumptions of their properties.

To understand their properties, it is necessary to understand few terminologies that are used in algebraic graph theory. The agent network communication topology is represented by graph $\mathcal{G} = (\mathcal{V},\ \mathcal{E},\ \mathcal{A})$ where $\mathcal{V} = \{1, 2, ..., N\}$ is the set of nodes and $\mathcal{E} = \{(i,j) : i,j \in \mathcal{V}\} \subset \mathcal{V} \times \mathcal{V}$ represent edges. In a directed graph case, there is communication only from one node to another node as it is shown in Figure 2.1 (i), the communication edges can be seen from one node to another node with arrowhead in single direction of information flow. Every digraph has associated with it an adjacency matrix $\mathcal{A}$ with non-negative binary elements. The order of adjacency matrix is $n \times n$, where n is the number of agents and is defined as $a_{ii} = 0$ and $a_{ij} \geq 0$. If there exists an edge between i and j, then the elements of matrix $\mathcal{A}$ is described as $a_{ij} > 0 \Leftrightarrow (i,j) \in \mathcal{E}$. The set of neighboring agents connected to any agent in topology is denoted by $\mathcal{N} = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. The graph Laplacian matrix $\mathcal{L} = (l_{ij})$ of the digraph $\mathcal{G}$ is defined by $l_{ij} = -a_{ij}$ for $i \neq j$, $l_{ii} = -\sum\limits_{j=1, j\neq i}^{n} l_{ij} (i,j = 1, 2, ..., N)$. So the adjacency matrix $\mathcal{A}$ and the graph Laplacian matrix $\mathcal{L}$ for the directed graph shown in 2.1 is given by,

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \text{ and } \mathcal{L} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{bmatrix}$$

However just finding the adjacency matrix and graph Laplacian matrix does not guarantee the connectivity of the topology. So it becomes necessary to draw some rules to ensure that the graph under consideration is strongly connected. The following properties of directed and undirected graphs are important with respect to the discussions and contributions of this thesis:

2.2.2 Properties of Graph Laplacian Matrix

The properties of $\mathcal{L}$ are as follows,

- For an undirected graph, the graph Laplacian matrix $\mathcal{L}$ is symmetric. However, for a directed graph, the sum of the rows is zero. So it is said that the digraph $\mathcal{G}$ has a globally reachable node if and only if the Laplacian matrix $\mathcal{L}$ of $\mathcal{G}$ has a simple zero eigenvalue with associated eigenvector $1_n \triangleq [1, ..., 1]^T \in R$, which is the $n \times 1$ column vector of ones and l is diagonally domain and has non-negative diagonal entries.

- If there exists a path between any two nodes i and j of a graph $\mathcal{G}$ with node i as the tail and j as the head then the graph $\mathcal{G}$ is connected. In the case of directed graphs, a directed graph is strongly connected if there also exists a path with j as the tail and i as the head.

- A graph $(\mathcal{G}_2)$ is a subgraph of $(\mathcal{G}_1)$ if $\mathcal{V}\ (\mathcal{G}_2) \subseteq \mathcal{V}\ (\mathcal{G}_1)$ and $\mathcal{E}\ (\mathcal{G}_2) \subseteq \mathcal{E}\ (\mathcal{G}_1)$.

- If $\mathcal{V}\ (\mathcal{G}_2) = \mathcal{V}\ (\mathcal{G}_1)$ then $\mathcal{G}_2$ is a spanning subgraph of $\mathcal{G}_1$.

- A connected graph $\mathcal{G}$ where each node has at least two neighboring nodes is said to contain a cycle i.e. starting from a node i as the tail in the graph $\mathcal{G}$, it is possible to have a path with finite number of edges to arrive at the same node i as the head.

- A spanning subgraph with no cycles is called as a spanning tree.

- The smallest eigenvalue of $\mathcal{L}$ is exactly zero and the corresponding eigenvector is given by $1 = (Col)\ (1. \dots 1)$.

- The graph Laplacian matrix $\mathcal{L}$ is always rank deficient and positive semi-definite.

- The rank of $\mathcal{L}$ is $N - 1$ if for $\mathcal{G}$ there exists a path from every node to all other nodes, i.e. the graph is connected.

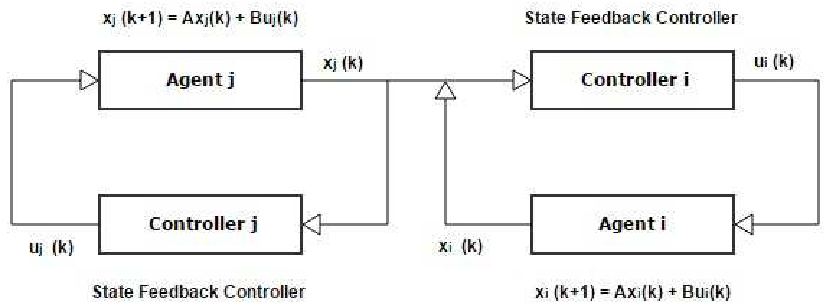

Figure 2.2: System model representation

2.3 Multi-agent System Modelling for Different Communication Constraints

Fig. 2.3 illustrates the working between a leader agent and a follower agent. Agent i is the leader agent and agent j is the follower agent. The communication link among the multi-agent system can be modeled by an algebraic graph. A system which is free of communication constraints can successfully transfer the information along this communication channel between these agents. The system dynamics for the MAS is discussed thoroughly in Chapter 3. However, in this section, an overview of system modeling is shown for communication constraints. Whenever there is a successful data transfer from one agent to another agent or agents in a networked MAS, there is a certain generation of topology which represents that network. It is desired that the topology should be non switching for any successful working of control system. However, this may not happen every time in real time conditions. There will be always some form of communication constraints which will affect the topology of the system. The two main communication constraints that affect the connected topology and indirectly the system behavior are packet data loss and time delays. Both these communication constraints are studied and modeled into sytem dynamics to study their effect on system performance. Chapter 3 basically deals with the controller design for a multi-agent system which comprises of these communication constraints. The final output of Chapter 3 is an LMI which may find a suitable control gain for the considered MAS with communication constraints. To understand the system dynamics in Chapter 3, it is necessary to understand the inclusion of communication

Figure 2.3: Packet data flow

constraints in system model as discussed in following subsections.

2.3.1 Modeling for Communication Channels and Packet Data Losses

In discrete time system, there is a communication event between two agents, which means that there is a packet data transfer in that communication event in a sampling period T_s. It is desired that the packet data transfer happen within that sampling period. If the packet data transfer takes more than a sampling period T_s, it is considered as a communication event failure or in simple terms as a packet data loss.

In [81], a Markov chain is adopted to model the packet data loss process in the communication channel. The Markov chain presents the transition from one state to another state between a finite number of possible states. The process is random and memoryless. The next state depends only on the current state and not on the sequence of events that preceded it as shown in the Fig. 2.3. The challenge is to find an optimal way to model the packet losses in every time step.

A Bernoulli distribution process is considered to model the packet losses in every time step. If there is no data receiving in current time step, then it will trace backward to previous data. This process guarantees the information received is the latest successful data transfer. So if the packet data is received after sampling period T_s or failed to receive, then in that case, a packet data loss is considered and the information state of earlier succesful transmission is passed on. The Bernoulli process is described as,

$$\mathbf{x_i}(k) \quad = \quad \theta_{ji}(k)\mathbf{x_i}(k) + (1 - \theta_{ji}(k))\mathbf{x_i}(k-1) \tag{2.1}$$

where $\theta_{ji} \in \mathbb{R}$ is a stochastic variable satisfying interval Bernoulli distribution and $\theta \in 0{,}1$. The subscript ji means that the signal is transferred from agent j to agent i. If $\theta_{ji} = 0$, then it means that the agent I has failed to receive signal from agent j and if $\theta_{ji} = 1$, then it means that agent i has received signal from agent j successfully. Eq.(2.1) is a recursive procedure. For example, if at $k = 1$, agent i can receive the signal, whle $k = 2{,}3$ cannot receive data serially, then

$$\mathbf{x_i}(3) \quad = \quad \theta_{ji}(3)\mathbf{x_i}(3) + (1 - \theta_{ji}(3))\mathbf{x_i}(2) = \mathbf{x_i}(2) \tag{2.2}$$

where

$$\mathbf{x_i}(2) \quad = \quad \theta_{ji}(2)\mathbf{x_i}(2) + (1 - \theta_{ji}(2))\mathbf{x_i}(1) = \mathbf{x_i}(1) \tag{2.3}$$

then

$$\mathbf{x_i}(3) = \mathbf{x_i}(1)$$

Therefore agent i receives the most recently successfully transferred data.

2.3.2 Summary

Chapter 2 presents a prerequisite information about the system modeling and the necessary theories associated with it. A perfect understanding of the algebraic graph theory is very important to determine the strongly connected topology for the successful working of networked system. Algebraic graph theory helps in determining the adjacency matrix and graph Laplacian matrix. In complex network systems, the understanding of rules of graph Laplacian matrix becomes quite important. In any of the rule is exploited, the controller design may fail at some stage. A leader-follower approach among different types of consensus is important in relation to the applications of the multi-agent systems. Most of the applications of the multi-agent systems require a decision maker or a team of decision maker which makes it favorable for leader-following approach. The application of packet data loss in the system dynamics is introduced in the Section 2.3.1. which makes use of Bernoulli's principle to model its working in system dynamics. In next chapter, the system dynamics with constraints like packet loss and time delays are studied. Many derivations related to error dynamics and system dynamics are shown. The controller designing process is also proved for two cases of constant time delay and time-varying delays. Chapter 3 is the main body of the thesis, in which the novelty of the work is presented.

Chapter 3

Consensus Control of Multi-Agent System using Lyapunov Based Approach

In this chapter, the controller design for the considered MAS dynamics is presented along with the derivation of its closed loop dynamics, error dynamics and finally a control gain value is attained by developing control algorithm.

3.1 Multi-Agent System Dynamics Considering Constant Time Delay and Data Packet Loss

A single control system dynamics is expressed in continuous-time domain as,

$$\dot{\mathbf{x}}_{\mathbf{i}}(t) = A_c \mathbf{x}_{\mathbf{i}}(t) + B_c u_i(t), \tag{3.1}$$

where, $A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}$ & $B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \mathbb{R}^{2\times 1}$ are constant matrices, $\dot{\mathbf{x}}_{\mathbf{i}}(\mathbf{t}) \in \mathbb{R}^{2\times 1}$ is the state vector, $u_i(t) \in \mathbb{R}$ is the control input at different time instant 't'. For a simple double integrator system,

$$\dot{x}_{1,i}(t) = x_{2,i}(t)$$
$$\dot{x}_{2,i}(t) = u_i(t)$$

State feedback controller is given by,

$$u_i(t) = -\boldsymbol{k}\mathbf{x}_{\mathbf{i}}(t), \tag{3.2}$$

where $i = 1, 2, 3, 4...n$ is the number of the agents, $\boldsymbol{k} = [k_1, k_2]$ is the control gain and $u_i(t)$ is the control input. Most of the practical control systems are computer based control signals which are controlled by digital controllers. These controllers accept analog or continuous signal, then convert them to digital or discrete signal for machine processing purpose and then convert back these discrete signals back to continuous signals or a form that the plant accepts as input. Discretization is done by using zero order hold and sampler. The sampler is used to convert continuous signals into a train of amplitude-modulated pulses and maintain the value of pulse for a prescribed time duration, and then can be read by digital controller. The zero-order hold is then to saturate the discrete signal into continuous ones. The discretization of a continuous state space system by zero-order hold is carried out in MATLAB®. The A_c and B_c are constant appropriate size of matrices. The A_d and B_d are varied according to the sampling time T_s. In our case, the sampling time T_s is set to 0.1 second. Converting continuous system to discrete system, the Equation (3.1) becomes,

$$\mathbf{x_i}(k+1) = A_d\mathbf{x_i}(k) + B_d u_i(k), \tag{3.3}$$

where,

$$A_d = \begin{bmatrix} 0 & 0.1 \\ 0 & 0 \end{bmatrix}; B_d = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix}$$

From Fig. 2.3, it is evident that the controller of the agent 'i' tries to reduce the error between x_j and x_i, i.e.,

$$u_i(t) = -\boldsymbol{k}\sum_{j=1}^{n} a_{ij}(\mathbf{x_j}(k) - \mathbf{x_i}(k)) \tag{3.4}$$

So the system becomes,

$$\mathbf{x_i}(k+1) = A_d\mathbf{x_i}(k) + B_d\boldsymbol{k}\sum_{j=1}^{n} a_{ij}(\mathbf{x_j}(k) - \mathbf{x_i}(k)), \tag{3.5}$$

Considering system having constant time delay and packet data loss,

$$\begin{aligned} \mathbf{x_i}(k+1) &= A_d\mathbf{x_i}(k) + B_d\boldsymbol{k}\sum_{j=1}^{n} a_{ij}[\theta(k)\mathbf{x_j}(k-\tau) + (1-\theta(k))\mathbf{x_j}(k-\tau-1) \\ &\quad -\theta(k)\mathbf{x_i}(k-\tau) - (1-\theta(k))\mathbf{x_i}(k-\tau-1)] \end{aligned} \tag{3.6}$$

where $\tau \in \mathbb{R}$ is the constant time delay and $\theta \in$ [0,1] is a probabilistic variable. The same equation can be represented as,

$$\mathbf{x}(k+1) = (I_n \otimes A_d)\mathbf{x}(k) + (L_1 \otimes B_d\boldsymbol{k})\mathbf{x}(k-\tau) + (L_2 \otimes B_d\boldsymbol{k})\mathbf{x}(k-\tau-1) \tag{3.7}$$

where $\otimes$ is the kronecker product, $\mathbf{x} = [x_1, x_2, ..., x_n]^T$, $L_1 \in \mathbb{R}^{n\times n}$ and $L_2 \in \mathbb{R}^{n\times n}$.

$$\begin{aligned} L_1 &= \begin{bmatrix} -\sum_{j=1}^{n} a_{1j}\theta & a_{12}\theta & \dots & a_{1n}\theta \\ a_{21}\theta & -\sum_{j=1}^{n} a_{2j}\theta & \dots & a_{2n}\theta \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}\theta & a_{n2}\theta & \dots & -\sum_{j=1}^{n} a_{nj}\theta \end{bmatrix} \\ &\triangleq L_1^0\theta \end{aligned}$$

where

$$L_1^0 = \begin{bmatrix} -\sum_{j=1}^{n} a_{1j} & a_{12} & \dots & a_{1n} \\ a_{21} & -\sum_{j=1}^{n} a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & -\sum_{j=1}^{n} a_{nj} \end{bmatrix}$$

$$\begin{aligned} L_2 = &\begin{bmatrix} -\sum_{j=1}^{n} a_{1j}(1-\theta) & a_{12}(1-\theta) & \dots & a_{1n}(1-\theta) \\ a_{21}(1-\theta) & -\sum_{j=1}^{n} a_{2j}(1-\theta) & \dots & a_{2n}(1-\theta) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(1-\theta) & a_{n2}(1-\theta) & \dots & -\sum_{j=1}^{n} a_{nj}(1-\theta) \end{bmatrix} \\ &\triangleq L_2^0(1-\theta) \end{aligned}$$

where

$$L_2^0 = \begin{bmatrix} -\sum_{j=1}^{n} a_{1j} & a_{12} & \dots & a_{1n} \\ a_{21} & -\sum_{j=1}^{n} a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & -\sum_{j=1}^{n} a_{nj} \end{bmatrix}$$

The cooperative control techniques are expected to be applied to actual vehicles such as UAVs, artificial satellites and autonomous mobile observation robots as well as distributed sensor networks. Consensus-based cooperative control, which is a distributed approach, has advantage of having network flexibility. Most works of the cooperative control problems using a consensus based algorithm assume that the vehicles are expressed in as a first order system [82]. The results in a first-order system can be directly extended to a second order system [83]. Works of a linear system are not directly extended from the results of in a first order system, because it is difficult to extend algorithms for a first-order system directly to the problems with a linear system. Therefor most authors who consider a linear system mainly use complicated ways such as an optimization approach [84], and a linear matrix inequality (LMI) [85]. In [86], a study of cooperative control problems is conducted to apply on formation control of multi-UAVs. They have expressed the dynamics of UAVs in the horizontal plane as s fourth-order system, then proposed a consensus-based algorithm for a group of UAVs to fly in a formation cooperatively.

3.2 Error Dynamics

3.2.1 Error Dynamics of the MAS with Constant Time Delay and Data Packet Loss

The error dynamics is defined as,

$$\mathbf{e_i}(k+1) = \mathbf{x_i}(k+1) - \mathbf{x_1}(k+1) \tag{3.8}$$

$$\begin{aligned}
\mathbf{e_i}(k+1) &= A_d\mathbf{x_i}(k) + B_d\boldsymbol{k}\sum_{j=1}^{n} a_{ij}[\theta(k)\mathbf{x_j}(k-\tau) + (1-\theta(k))\mathbf{x_j}(k-\tau-1) \\
&\quad -\theta(k)\mathbf{x_i}(k-\tau) - (1-\theta(k))\mathbf{x_i}(k-\tau-1)] - A_d\mathbf{x_1}(k) \\
&\quad -B_d\boldsymbol{k}\sum_{j=1}^{n} a_{1j}[\theta(k)\mathbf{x_j}(k-\tau) + (1-\theta(k))\mathbf{x_j}(k-\tau-1) \\
&\quad -\theta(k)\mathbf{x_1}(k-\tau) - (1-\theta(k))\mathbf{x_1}(k-\tau-1)]
\end{aligned} \tag{3.9}$$

$$\begin{aligned}\mathbf{e_i}(k+1) &= A_d[\mathbf{x_i}(k)-\mathbf{x_1}(k)] + B_d\boldsymbol{k}\sum_{j=1}^{n}(a_{ij}-a_{1j})[\theta(k)\mathbf{x_j}(k-\tau)\\ &\quad -(1-\theta(k))\mathbf{x_j}(k-\tau-1)] - B_d\boldsymbol{k}\sum_{j=1}^{n}(a_{ij}-a_{1j})[\theta(k)(\mathbf{x_i}(k-\tau)\\ &\quad -\mathbf{x_1}(k-\tau)) + (1-\theta(k))(\mathbf{x_i}(k-\tau-1)-\mathbf{x_1}(k-\tau-1))\end{aligned} \tag{3.11}$$

$$\begin{aligned}\mathbf{e_i}(k+1) &= A_d[\mathbf{x_i}(k)-\mathbf{x_1}(k)] + B_d\boldsymbol{k}\sum_{j=2}^{n}(a_{ij}-a_{1j})[\theta(k)\mathbf{x_j}(k-\tau)-\theta(k)\mathbf{x_1}(k-\tau)]\\ &\quad +B_d\boldsymbol{k}\sum_{j=2}^{n}(a_{ij}-a_{1j})[(1-\theta(k))\mathbf{x_j}(k-\tau-1)-(1-\theta(k))\mathbf{x_1}(k-\tau-1)]\\ &\quad -B_d\boldsymbol{k}\sum_{j=1}^{n}(a_{ij}-a_{1j})[(1-\theta(k))\mathbf{x_i}(k-\tau-1)-(1-\theta(k))\mathbf{x_1}(k-\tau-1)]\\ &\quad -B_d\boldsymbol{k}\sum_{j=1}^{n}(a_{ij}-a_{1j})[\theta(k)\mathbf{x_i}(k-\tau)-\theta(k)\mathbf{x_1}(k-\tau)]\end{aligned} \tag{3.12}$$

Let us declare
$\mathbf{x_i}(k)-\mathbf{x_1}(k)=\mathbf{e_i}(k)$, $\mathbf{x_j}(k-\tau-1)-\mathbf{x_1}(k-\tau-1)=\mathbf{e_j}(k-\tau-1)$,
$\mathbf{x_i}(k-\tau)-\mathbf{x_1}(k-\tau)=\mathbf{e_i}(k-\tau)$ and $\mathbf{x_i}(k-\tau-1)-\mathbf{x_1}(k-\tau-1)=\mathbf{e_i}(k-\tau-1)$.
So the above equation becomes,

$$\begin{aligned}\mathbf{e_i}(k+1) &= A_d\mathbf{e_i}(k) + B_d\boldsymbol{k}\sum_{j=2}^{n}(a_{ij}-a_{1j})\theta(k)\mathbf{e_j}(k-\tau)\\ &\quad +B_d\boldsymbol{k}\sum_{j=2}^{n}(a_{ij}-a_{1j})(1-\theta(k))\mathbf{e_j}(k-\tau-1)\\ &\quad -B_d\boldsymbol{k}\sum_{j=1}^{n}(a_{ij}-a_{1j})\theta(k)\mathbf{e_i}(k-\tau)\\ &\quad -B_d\boldsymbol{k}\sum_{j=1}^{n}(a_{ij}-a_{1j})(1-\theta(k))\mathbf{e_i}(k-\tau-1)\end{aligned} \tag{3.13}$$

In Matrix Form, this equation can be represented as,

$$\mathbf{e}(k+1) = (I_{n-1} \otimes A_d)\mathbf{e}(k) + (\bar{L_1} \otimes B_d \boldsymbol{k})\mathbf{e}(k-\tau) + (\bar{L_2} \otimes B_d \boldsymbol{k})\mathbf{e}(k-\tau-1) \quad (3.14)$$

where n $\geq$ 2, thus, $\text{O}_1 = \sum_{j=1}^{n} a_{2j}$, $\text{O}_2 = \sum_{j=1}^{n} a_{3j}$ and $\text{O}_n = \sum_{j=1}^{n} a_{nj}$ and

$$\bar{L_1^0} = \begin{bmatrix} (a_{22}-a_{12})-O_1 & (a_{23}-a_{13}) & \dots & (a_{2n}-a_{1n}) \\ (a_{32}-a_{12}) & (a_{33}-a_{13})-O_2 & \dots & (a_{3n}-a_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ (a_{n2}-a_{12}) & (a_{n3}-a_{13}) & \dots & (a_{nn}-a_{1n})-O_n \end{bmatrix}$$

Also,

where n $\geq$ 2, thus, $Q_1 = \sum_{j=1}^{n} a_{3j}$, $Q_2 = \sum_{j=1}^{n} a_{3j}$, $Q_n = \sum_{j=1}^{n} a_{nj}$ and

$$\bar{L_2^0} = \begin{bmatrix} (a_{22}-a_{12})-Q_1 & (a_{23}-a_{13} & \dots & (a_{2n}-a_{1n} \\ (a_{32}-a_{12} & (a_{33}-a_{13})-Q_2 & \dots & (a_{3n}-a_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ (a_{n2}-a_{12}) & (a_{n3}-a_{13}) & \dots & (a_{nn}-a_{1n})-Q_n \end{bmatrix}$$

3.2.2 Error Dynamics of the MAS with Time-Varying Delay and Data Packet Loss

The above discussion about controller design was based on the constant delay case in which we assume that at any given time, the time delays present in the system would either be zero or constant in all communication channels. The reason for considering constant time delay is when the time delays in the system are fluctuating very less and thesystem behavior overall is not affected by these fluctuations. In this case, we consider the maximum observed value of time delay as τ. However, this is not always possible while designing real-time network control. Time delays may vary within a high range, in which it becomes necessary to consider the lower and upper bounds of time delays. So, in this section, we will be considering a new variable τ_k in time varying delay case instead of τ in constant delay case. τ_k is an arbitrary real variable which can take any values between the lower bound time delay τ_m and the upper bound time delay τ_M, such that $\tau_m \leq \tau_k \leq \tau_M$.

After considering the system with time varying delay and packet data loss,

$$\begin{aligned} \mathbf{x_i}(k+1) &= A_d\mathbf{x_i}(k) + B_d\boldsymbol{k}\sum_{j=1}^{n} a_{ij}[\theta(k)\mathbf{x_j}(k-\tau_k) + (1-\theta(k))\mathbf{x_j}(k-\tau_k-1) \\ &\quad -\theta(k)\mathbf{x_i}(k-\tau_k) - (1-\theta(k))\mathbf{x_i}(k-\tau_k-1)] \end{aligned} \quad (3.15)$$

In Matrix form, the same equation can be represented as,

$$\begin{aligned} \mathbf{x}(k+1) &= (I_n \otimes A_d)\mathbf{x}(k) + (L_1 \otimes B_d\boldsymbol{k})\mathbf{x}(k-\tau_k) \\ &\quad +(L_2 \otimes B_d\boldsymbol{k})\mathbf{x}(k-\tau_k-1) \end{aligned} \quad (3.16)$$

where $\otimes$ is the kronecker product, $x = [x_1, x_2, ..., x_n]^T$, $L_1 \in \mathbb{R}^{n\times n}$ and $L_2 \in \mathbb{R}^{n\times n}$. The values of variables of A_d, L_1 and L_2 won't be affected as they are not dependent on time delays. These matrices are dependent on adjacency matrix elements and packet data loss variable.

The error dynamics of the system is as follows,

$$\begin{aligned} \mathbf{e}(k+1) &= (I_{n-1} \otimes A_d)\mathbf{e}(k) + (\bar{L_1} \otimes B_d\boldsymbol{k})\mathbf{e}(k-\tau_k) \\ &\quad +(\bar{L_2} \otimes B_d\boldsymbol{k})\mathbf{e}(k-\tau_k-1) \end{aligned} \quad (3.17)$$

where $\mathbf{e}(k) = [e_2(k), e_3(k), ..., e_n(k)] \in \mathbb{R}^{2(n-1)\times 1}$

3.3 Consensus Control of the MAS or Controller Design

3.3.1 Consensus Control of the MAS with Constant Time Delay and Data Packet Loss

Let us assume the following Lyapunov functional candidate for the system comprising of constant time delay and packet data loss as,

$$V(k) \quad = \quad V_1(k) + V_2(k) + V_3(k) \tag{3.18}$$

where,

$V_1(k) = \mathbf{e}^T(k)P\mathbf{e}(k)$, with $P \in \mathbb{R}^{2(n-1)\times 2(n-1)}$ and n is the number of the agents.

$V_2(k) = \sum_{i=1}^{\tau} \mathbf{e}^T(k-i)Q\mathbf{e}(k-i)$, with $Q \in \mathbb{R}^{2(n-1)\times 2(n-1)}$

$V_3(k) = \sum_{i=-\tau}^{0} \sum_{j=i+k-1}^{k-1} (\mathbf{e}^T(j+1) - \mathbf{e}^T(j))R(\mathbf{e}(j+1) - \mathbf{e}(j))$, with $R \in \mathbb{R}^{2(n-1)\times 2(n-1)}$

A Lyapunov function V is chosen such that $V(e) \geq 0, \forall\, e$ and $V = 0$ only when e = 0 and P, Q, and R are positive definite matrices. If it can be shown that $\Delta\ V = V(e, k+1) - V(e, k) < 0$, then e converges asymptotically to zero.

$$\Delta V_1 = \mathbf{e}^{\mathbf{T}}(k+1)P\mathbf{e}(k+1) - \mathbf{e}^{\mathbf{T}}(k)P\mathbf{e}(k)$$

$$= \begin{bmatrix} \mathbf{e}^T(k+1) \\ \mathbf{e}^T(k) \\ \mathbf{e}^T(k-\tau) \\ \mathbf{e}^T(k-\tau-1) \end{bmatrix} \begin{bmatrix} P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}(k+1) \\ \mathbf{e}(k) \\ \mathbf{e}(k-\tau) \\ \mathbf{e}(k-\tau-1) \end{bmatrix} -$$

$$\begin{bmatrix} \mathbf{e}^T(k+1) \\ \mathbf{e}^T(k) \\ \mathbf{e}^T(k-\tau) \\ \mathbf{e}^T(k-\tau-1) \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}(k+1) \\ \mathbf{e}(k) \\ \mathbf{e}(k-\tau) \\ \mathbf{e}(k-\tau-1) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{e}^T(k+1) \\ \mathbf{e}^T(k) \\ \mathbf{e}^T(k-\tau) \\ \mathbf{e}^T(k-\tau-1) \end{bmatrix} \begin{bmatrix} P & 0 & 0 & 0 \\ 0 & P & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}(k+1) \\ \mathbf{e}(k) \\ \mathbf{e}(k-\tau) \\ \mathbf{e}(k-\tau-1) \end{bmatrix}$$

Let $\mathbf{z}(k) = \begin{bmatrix} \mathbf{e}(k+1) \\ \mathbf{e}(k) \\ \mathbf{e}(k-\tau) \\ \mathbf{e}(k-\tau-1) \end{bmatrix} \in \mathbb{R}^{8(n-1)\times 1}$, so we can write above equation as,

$$\Delta V_1 = \mathbf{z}^T(k) \begin{bmatrix} P & 0 & 0 & 0 \\ 0 & -P & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{z}(k)$$

$$\Delta V_2 = \sum_{i=1}^{\tau} \mathbf{e}^T(k+1-i)Q\mathbf{e}(k+1-i) - \sum_{i=1}^{\tau} \mathbf{e}^T(k-i)Q\mathbf{e}(k-i)$$

$$= \mathbf{e}^T(k)Q\mathbf{e}(k) - \mathbf{e}^T(k-\tau)Q\mathbf{e}(k-\tau)$$

$$= \begin{bmatrix} \mathbf{e}^T(k+1) \\ \mathbf{e}^T(k) \\ \mathbf{e}^T(k-\tau) \\ \mathbf{e}^T(k-\tau-1) \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & Q & 0 & 0 \\ 0 & 0 & -Q & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}(k+1) \\ \mathbf{e}(k) \\ \mathbf{e}(k-\tau) \\ \mathbf{e}(k-\tau-1) \end{bmatrix}$$

$$\Delta V_2 = \mathbf{z}^T(k) \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & Q & 0 & 0 \\ 0 & 0 & -Q & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{z}(k)$$

$\Delta V_3 = \sum_{i=-\tau}^{0} \sum_{j=i+k+1}^{k} (\mathbf{e}^T(j+1) - \mathbf{e}^T(j))R(\mathbf{e}(j+1) - \mathbf{e}(j)) - \sum_{i=-\tau}^{0} \sum_{j=i+k}^{k-1} (\mathbf{e}^T(j+1) - \mathbf{e}^T(j))R(\mathbf{e}(j+1) - \mathbf{e}(j))$

$= (\tau+1)((\mathbf{e}^T(k+1) - e^T(k))R(\mathbf{e}(k+1) - e(k)) - \sum_{i=-\tau}^{0} ((\mathbf{e}^T(k+i+1) - \mathbf{e}^T(k+i))R(\mathbf{e}(k+i+1) - \mathbf{e}(k+i))$

The last summation term needs to be canceled out. So we need to add quadratic inequality summed up from 0 to τ at this end to get the final identity.

$-2\mathbf{z}^T(k)N \sum_{i=-\tau}^{0} (\mathbf{e}(k+i+1) - \mathbf{e}(k+i)) \leq (\tau+1)\mathbf{z}^T(k)NR^{-1}N^T\mathbf{z}(k) + \sum_{i=-\tau}^{0} ((\mathbf{e}^T(k+i+1) - \mathbf{e}^T(k+i))R(\mathbf{e}(k+i+1) - \mathbf{e}(k+i))$

This identity does help to remove the inconvenient summation term, but at the same time leaves behind another inconvenient summation term. This newly generated summation term can be cancelled out by forming a zero -equation ψ_0. But,

$$\sum_{i=-\tau}^{0}(\mathbf{e}(k+i+1)-\mathbf{e}(k+i)) = (\mathbf{e}(k+1)-\mathbf{e}(k-\tau))$$

$$\psi_0 = 2z^T(k)N[(\mathbf{e}(k+1)-\mathbf{e}(k-\tau)) - \sum_{i=-\tau}^{0}(\mathbf{e}(k+i+1)-\mathbf{e}(k+i))] = 0$$

where, N $= \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix}$, $N_i \in \mathbb{R}^{2(n-1)\times 2(n-1)}$ is an arbitrary design parameter.

Therefore, the zero-equation simplifies ΔV_3 to

$$\Delta V_3 = \Delta V_3 + \psi_0$$

$$\Delta V_3 = \mathbf{z}^T(k)\begin{bmatrix} (\tau+1)rR + N_1 + N_1^T & -(\tau+1)(1-r)R + N_2^T & N_3^T - N_1 & 0 \\ -(\tau+1)(1-r)R^T + N_2 & (\tau+1)rR & -N_2 & 0 \\ N_3 - N_1^T & -N_2^T & -N_3 - N_3^T & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\mathbf{z}(\mathbf{k})$$
$$+(\tau+1)\mathbf{z}^{\mathbf{T}}(k)NR^{-1}N^T\mathbf{z}(\mathbf{k})$$

Another zero-equation ψ_1 uses the error dynamics to introduce gain in the system where $M = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}$, $M_i \in \mathbb{R}^{2(n-1)\times 2(n-1)}$ is another arbitrary matrix introduced, where M_1, M_2, M_3 and M_4 are symmetric matrices.

$$\psi_1 = 0 = 2z^T(k)M[\mathbf{e}(k+1) - (I \otimes A_d)\mathbf{e}(k) - (\bar{L}_1 \otimes B_d\boldsymbol{k})\mathbf{e}(k-\tau)$$
$$-(\bar{L}_2 \otimes B_d\boldsymbol{k})\mathbf{e}(k-\tau-1)]$$

$$\psi_1 = \mathbf{z}^{\mathbf{T}}(k)\begin{bmatrix} M_1 + M_1^T & -M_1(I\otimes \mathrm{A}_d) + M_2^T & -M_1(\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k}) & -M_1(\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k}) + M_4^T \\ -M_1^T(I\otimes \mathrm{A}_d)^T + M_2 & -M_2(I\otimes \mathrm{A}_d) - (I\otimes \mathrm{A}_d)^T M_2^T & -\mathbf{M_2}(\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k}) - (I\otimes \mathrm{A}_d)^T M_3^T & -_2(\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k}) - (I\otimes \mathrm{A}_d)^T M_4^T \\ -M_1^T(\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k})^T & -M_2^T(\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k})^T - (I\otimes \mathrm{A}_d)M_3 & -M_3(\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k}) - (\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k})^T M_3^T & -M_3(\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k}) - (\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k})^T M_4^T \\ -M_1^T(\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k})^T + M_4 & -M_2^T(\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k})^T - (I\otimes \mathrm{A}_d)M_4 & -M_3^T(\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k})^T - (\bar{L}_1\otimes \mathrm{B}_d\boldsymbol{k})M_4 & -M_4(\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k}) - (\bar{L}_2\otimes \mathrm{B}_d\boldsymbol{k})^T M_4^T \end{bmatrix}\mathbf{z}(k)$$

Let $(I \otimes A_d) = T_1$, $(\bar{L}_1 \otimes B_d\boldsymbol{k}) = T_2K$, $(\bar{L}_2 \otimes B_d\boldsymbol{k}) = T_3K$, $M_2 = \omega_1 M_1$, $M_3 = \omega_2 M_1$ and $M_4 = \omega_3 M_1$ and $K = diag[\boldsymbol{k}, ..., \boldsymbol{k}] \in \mathbb{R}^{(n-1)\times 2(n-1)}$, hence

$$\psi_1 = \mathbf{z^T}(k) \begin{bmatrix} M_1 + M_1^T & -M_1T_1 + \omega_1^T M_1^T & -M_1T_2K & -M_1T_3K + \omega_3^T M_1^T \\ -M_1^T T_1^T + \omega_1 M_1 & -\omega_1 M_1 T_1 - T_1^T \omega_1^T M_1^T & -\omega_1 M_1 T_2 K - T_1^T \omega_2 M_1^T & -\omega_1 M_1 T_3 K - T_1^T \omega_3^T M_1^T \\ -M_1^T T_2^T K^T & -\omega_1^T M_1^T T_2^T K^T - T_1 \omega_2 M_1 & -\omega_2 M_1 T_2 K - T_2^T K^T \omega_2^T M_1^T & -\omega_2 M_1 T_3 K - T_2^T K^T \omega_3 M_1^T \\ -M_1^T T_3^T K^T + \omega_3 M_1 & -\omega_1^T M_1^T T_3^T K^T - T_1 \omega_3 M_1 & -\omega_2^T M_1^T T_3^T K^T - T_2 K \omega_3 M_1 & -\omega_3 M_1 T_3 K - T_3^T K^T \omega_3^T M_1^T \end{bmatrix} \mathbf{z}(k)$$

Then $\mathbb{E}(\Delta V) = \Delta V_1 + \Delta V_2 + \Delta V_3 + \psi_0 + \psi_1$

$$\mathbb{E}(\Delta V) = \mathbf{z^T}(k) \begin{bmatrix} \Xi_{11} & \Xi_{21}^T & \Xi_{31}^T & \Xi_{41}^T \\ \Xi_{21} & \Xi_{22} & \Xi_{32}^T & \Xi_{42}^T \\ \Xi_{31} & \Xi_{32} & \Xi_{33} & \Xi_{43}^T \\ \Xi_{41} & \Xi_{42} & \Xi_{43} & \Xi_{44} \end{bmatrix} \mathbf{z}(k) + (\tau+1)\mathbf{z^T}(k) N R^{-1} N^T \mathbf{z}(k)$$

where,

$$\Xi_{11} = P + (\tau+1)R + N_1 + N_1^T + M_1 + M_1^T$$

$$\Xi_{21} = -(\tau+1)R^T + N_2 - M_1^T T_1^T + \omega_1 M_1$$

$$\Xi_{22} = -P + Q + (\tau+1)R - \omega_1 M_1 T_1 - T_1^T \omega_1 M_1^T$$

$$\Xi_{31} = N_3 - N_1^T - M_1^T \bar{T_2}^T K^T$$

$$\Xi_{32} = -N_2^T - \omega_1 M_1^T \bar{T_2}^T K^T - T_1 \omega_2 M_1$$

$$\Xi_{33} = -Q - N_3 - N_3^T - \omega_2 M_1 \bar{T_2} K - \bar{T_2}^T K^T \omega_2 M_1^T$$

$$\Xi_{41} = -M_1^T \bar{T_3}^T K^T + \omega_3 M_1$$

$$\Xi_{42} = -\omega_1 M_1^T \bar{T_3}^T K^T - T_1 \omega_3 M_1$$

$$\Xi_{43} = -\omega_2 M_1^T \bar{T_3}^T K^T - \bar{T_2} K \omega_3 M_1$$

$$\Xi_{44} = -\omega_3 M_1 \bar{T_3} K - \bar{T_3}^T K^T \omega_3 M_1^T$$

$$\bar{T_2} = \mathbb{E}(T_2) = (\bar{L_1^0} \otimes B_d \boldsymbol{k}) r$$

$$\bar{T_3} = \mathbb{E}(T_3) = (\bar{L_2^0} \otimes B_d \boldsymbol{k})(1-r)$$

The Schur complement for any matrix X of the form

$\mathrm{X} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, gives the following condition, $X < 0 \Leftrightarrow A - BD^{-1}C < 0$

The Schur complement gives new equation, such that if it holds true, then the error will be bounded, i.e. $\Delta V < 0$

$$\Xi = \begin{bmatrix} \Xi_{11} & \Xi_{21}^T & \Xi_{31}^T & \Xi_{41}^T & (\tau+1)N_1 \\ \Xi_{21} & \Xi_{22}+S & \Xi_{32}^T & \Xi_{42}^T & (\tau+1)N_2 \\ \Xi_{31} & \Xi_{32} & \Xi_{33} & \Xi_{43}^T & (\tau+1)N_3 \\ \Xi_{41} & \Xi_{42} & \Xi_{43} & \Xi_{44} & (\tau+1)N_4 \\ (\tau+1)N_1^T & (\tau+1)N_2^T & (\tau+1)N_3^T & (\tau+1)N_4^T & -(\tau+1)R \end{bmatrix} < 0$$

where r is the data loss rate and S is the arbitrary symmetric positive definite matrix. This above equation is actually a stability test. This LMI contains arbitrary symmetric positive definite matrices P, Q, R, S, M and N. If the values for these matrices can be found out by calculated control gain, then we can say that the considered system will be stable. It is necessary to linearize the above inequality for which we need to pre and post multiply by M_1^{-1}. Since M_1 is a symmetric matrix, $M_1^T = M_1$. Assume $X = M_1^{-1}$,which is also a symmetric matrix and $Y = KX$, we get,

$$\Phi_c = M_1^{-1}\Xi M_1^{-1} = \begin{bmatrix} \Phi_{11} & \Phi_{21}^T & \Phi_{31}^T & \Phi_{41}^T & (\tau+1)\hat{N}_1 \\ \Phi_{21} & \Phi_{22} & \Phi_{32}^T & \Phi_{42}^T & (\tau+1)\hat{N}_2 \\ \Phi_{31} & \Phi_{32} & \Phi_{33} & \Phi_{43}^T & (\tau+1)\hat{N}_3 \\ \Phi_{41} & \Phi_{42} & \Phi_{43} & \Phi_{44} & (\tau+1)\hat{N}_4 \\ (\tau+1)\hat{N}_1^T & (\tau+1)\hat{N}_2^T & (\tau+1)\hat{N}_3^T & (\tau+1)\hat{N}_4^T & -(\tau+1)\hat{R} \end{bmatrix} < 0$$

where,

$$\Phi_{11} = \hat{P} + (\tau+1)\hat{R} + \hat{N}_1 + \hat{N}_1^T + 2X$$

$$\Phi_{21} = -(\tau+1)^T\hat{R}^T + \hat{N}_2 - T_1^T X + \omega_1 X$$

$$\Phi_{22} = -\hat{P} + r\hat{Q} + (\tau+1)r\hat{R} - \omega_1 T_1 X - \omega_1 T_1^T X + \hat{S}$$

$$\Phi_{31} = \hat{N}_3 - \hat{N}_1^T - \bar{T}_2^T Y$$

$$\Phi_{32} = -\hat{N}_2^T - \omega_1\bar{T}_2^T Y - \omega_2 T_1 X$$

$$\Phi_{33} = -\hat{Q} - \hat{N}_3 - \hat{N}_3^T - \omega_2\bar{T}_2 Y - \omega_2\bar{T}_2^T Y$$

$$\Phi_{41} = -T_3 Y + \omega_3 X$$

$$\Phi_{42} = -\omega_1\bar{T}_3^T Y - \omega_3\bar{T}_3 Y$$

$$\Phi_{43} = -\omega_2\bar{T}_3^T Y - \omega_3\bar{T}_2 Y$$

$\Phi_{44} = -\omega_3 \bar{T_3} Y - \omega_3 \bar{T_3}^T Y$

and

$\hat{P} = M_1^{-1} P M_1^{-1}$

$\hat{Q} = M_1^{-1} Q M_1^{-1}$

$\hat{R} = M_1^{-1} R M_1^{-1}$

$\hat{S} = M_1^{-1} S M_1^{-1}$

$\hat{N}_1 = M_1^{-1} N_1 M_1^{-1}$

$\hat{N}_2 = M_1^{-1} N_1 M_1^{-1}$

$\hat{N}_3 = M_1^{-1} N_3 M_1^{-1}$

$\hat{N}_4 = M_1^{-1} N_4 M_1^{-1}$

3.3.2 Consensus Control of the MAS considering time-varying delay and data packet loss

Let us assume the following Lyapunov functional candidate for the system comprising of time-varying delays and packet data loss as,

$V(k) = V_1(k) + V_2(k) + V_3(k) + V_4(k)$

where,

$$V_1(k) = \mathbf{e^T}(k) P \mathbf{e}(k)$$

$$V_2(k) = \sum_{i=k-\tau_k}^{k-1} \mathbf{e^T}(i) Q \mathbf{e}(i)$$

$$V_3(k) = \sum_{i=-\tau_M+2}^{-\tau_m+1} \sum_{j=k+i-1}^{k-1} \mathbf{e^T}(j) Q \mathbf{e}(j)$$

$$V_4(k) = \sum_{i=-\tau_M}^{-1} \sum_{j=k+i}^{k-1} (\mathbf{e^T}(j+1) - \mathbf{e^T}(j)) R (\mathbf{e}(j+1) - \mathbf{e}(j))$$

In practical applications, it is impossible to know the exact value of the delay, but a lower and upper bound of time delays can always be determined. In this case, there is addition of one more term $V_4(k)$. The process to assume $V_4(k)$ is same as $V_3(k)$ in earlier section. However $V_3(k)$ in this case or section is an addition because there is presence of two ranges of delays i.e. one is between τ_m and τ_k and another is between τ_k and τ_M.

$$\Delta V_1 = \mathbf{z}^T(k) \begin{bmatrix} P & 0 & 0 & 0 \\ 0 & -P & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{z}(k)$$

$$\Delta V_2 = \sum_{i=k-\tau_k+1}^{k} \mathbf{e^T}(i)Q\mathbf{e}(i) - \sum_{i=k-\tau_k}^{k-1} \mathbf{e^T}(i)Q\mathbf{e}(i)$$

Since,

$$\sum_{i=k-\tau_{k+1}+1}^{k-1} \mathbf{e^T}(i)Q\mathbf{e}(i) = \sum_{i=k-\tau_m+1}^{k-1} \mathbf{e^T}(i)Q\mathbf{e}(i) + \sum_{i=k-\tau_{k+1}+1}^{k-\tau_m} \mathbf{e^T}(i)Q\mathbf{e}(i) \leq \sum_{i=k-\tau_k+1}^{k-1} \mathbf{e^T}(i)Q\mathbf{e}(i) + \sum_{i=k-\tau_M+1}^{k-\tau_m} \mathbf{e^T}(i)Q\mathbf{e}(i)$$

which is used to eliminate the random variable in summation term.

Thus, $\Delta V_2 = \mathbf{e^T}(k)Q\mathbf{e}(k) - \mathbf{e^T}(k-\tau_k)Q\mathbf{e}(k-\tau_k) + \xcancel{\sum_{i=k-\tau_M+1}^{k-\tau_m} \mathbf{e^T}(i)Q\mathbf{e}(i)}$

$$\Delta V_3 = \sum_{i=-\tau_M+2}^{-\tau_m+1} \sum_{j=k+i}^{k} \mathbf{e^T}(i)Q\mathbf{e}(i) - \sum_{i=-\tau_m+2}^{-\tau_m+1} \sum_{j=k+i-1}^{k-1} \mathbf{e^T}(i)Q\mathbf{e}(i)$$

$$= (\tau_M - \tau_m)\mathbf{e^T}(k)Q\mathbf{e}(k) - \xcancel{\sum_{j=k+\tau_M+1}^{k-\tau_m} \mathbf{e^T}(j)Q\mathbf{e}(j)}$$

$$\Delta V_1 + \Delta V_2 + \Delta V_3 = \mathbf{z^T}(k) \begin{bmatrix} P & 0 & 0 & 0 \\ 0 & (\tau_M - \tau_m + 1)Q - P & 0 & 0 \\ 0 & 0 & -Q & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{z}(k)$$

ΔV_4 is same as that of ΔV_3 of constant delay case derivation except for change of limits.

$$\Delta V_4 = \mathbf{z^T}(k) \begin{bmatrix} (\tau_M+1)R + N_1 + N_1^T & -(\tau_M+1)R + N_2^T & N_3^T - N_1 & 0 \\ -(\tau_M+1)R^T + N_2 & (\tau_M+1)R & -N_2 & 0 \\ N_3 - N_1^T & -N_2^T & -N_3 - N_3^T & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{z}(\mathbf{k})$$

$$+(\tau_M+1)\mathbf{z^T}(k)NR^{-1}N^T\mathbf{z}(k)$$

$= \Delta V_1 + \Delta V_2 + \Delta V_3 + \Delta V_4 + \psi_0 + \psi_1$, where ψ_0, ψ_1 have the same values as that in constant time delay case. For their derivation, kindly refer constant-time delay controller design. So adding up all matrices, we get ΔV as

$$\mathbb{E}(\Delta V) = \mathbf{z^T}(k) \begin{bmatrix} \Xi_{11} & \Xi_{21}^T & \Xi_{31}^T & \Xi_{41}^T \\ \Xi_{21} & \Xi_{22} & \Xi_{32}^T & \Xi_{42}^T \\ \Xi_{31} & \Xi_{32} & \Xi_{33} & \Xi_{43}^T \\ \Xi_{41} & \Xi_{42} & \Xi_{43} & \Xi_{44} \end{bmatrix} \mathbf{z}(k) + (\tau_M+1)\mathbf{z^T}(k)NR^{-1}N^T\mathbf{z}(k)$$

where,

$$\Xi_{11} = P + (\tau_M+1)R + N_1 + N_1^T + M_1 + M_1^T$$

$$\Xi_{21} = -(\tau_M+1)R^T + N_2 - M_1^T T_1^T + \omega_1 M_1$$

$$\Xi_{22} = -P + (\tau_M - \tau_m + 1)Q + (\tau_M+1)R - \omega_1 M_1 T_1 - T_1^T \omega_1 M_1^T$$

$$\Xi_{31} = N_3 - N_1^T - M_1^T \bar{T_2}^T K^T$$

$$\Xi_{32} = -N_2^T - \omega_1 M_1^T \bar{T_2}^T K^T - T_1 \omega_2 M_1$$

$$\Xi_{33} = -Q - N_3 - N_3^T - \omega_2 M_1 \bar{T}_2 K - \bar{T_2}^T K^T \omega_2 M_1^T$$

$$\Xi_{41} = -M_1^T \bar{T_3}^T K^T + \omega_3 M_1$$

$$\Xi_{42} = -\omega_1 M_1^T \bar{T_3}^T K^T - T_1 \omega_3 M_1$$

$$\Xi_{43} = -\omega_2 M_1^T \bar{T_3}^T K^T - \bar{T}_2 K \omega_3 M_1$$

$$\Xi_{44} = -\omega_3 M_1 \bar{T}_3 K - \bar{T_3}^T K^T \omega_3 M_1^T$$

$$\bar{T}_2 = \mathbb{E}(T_2) = (\bar{L_1^0} \otimes B_d \boldsymbol{k})r$$

$$\bar{T}_3 = \mathbb{E}(T_3) = (\bar{L_2^0} \otimes B_d \boldsymbol{k})(1-r)$$

Applying Schlur Complement on above equation, we get a new equation in form,

$$\Xi = \begin{bmatrix} \Xi_{11} & \Xi_{21}^T & \Xi_{31}^T & \Xi_{41}^T & (\tau_M+1)N_1 \\ \Xi_{21} & \Xi_{22}+S & \Xi_{32}^T & \Xi_{42}^T & (\tau_M+1)N_2 \\ \Xi_{31} & \Xi_{32} & \Xi_{33} & \Xi_{43}^T & (\tau_M+1)N_3 \\ \Xi_{41} & \Xi_{42} & \Xi_{43} & \Xi_{44} & (\tau_M+1)N_4 \\ (\tau_M+1)N_1^T & (\tau_M+1)N_2^T & (\tau_M+1)N_3^T & (\tau_M+1)N_4^T & -(\tau_M+1)R \end{bmatrix} < 0$$

The variables in this Ξ matrix are same as that derived Ξ matrix in constant-time delay case. If the values for these matrices can be found out by calculated control gain, then we can say that the considered system will be stable. It is necessary to linearize the above inequality for which we need to pre and post multiply by M_1^{-1}. Since M_1 is a symmetric matrix, $M_1^T = M_1$. Assume $X = M_1^{-1}$,which is also a symmetric matrix and $Y = KX$, we get,

$$\Phi_d = \mathbf{M_1^{-1}}\Xi\mathbf{M_1^{-1}} = \begin{bmatrix} \Phi_{11} & \Phi_{21}^T & \Phi_{31}^T & \Phi_{41}^T & (\tau_M+1)\hat{N}_1 \\ \Phi_{21} & \Phi_{22} & \Phi_{32}^T & \Phi_{42}^T & (\tau_M+1)\hat{N}_2 \\ \Phi_{31} & \Phi_{32} & \Phi_{33} & \Phi_{43}^T & (\tau_M+1)\hat{N}_3 \\ \Phi_{41} & \Phi_{42} & \Phi_{43} & \Phi_{44} & (\tau_M+1)\hat{N}_4 \\ (\tau_M+1)\hat{N}_1^T & (\tau_M+1)\hat{N}_2^T & (\tau_M+1)\hat{N}_3^T & (\tau_M+1)\hat{N}_4^T & -(\tau_M+1)\hat{R} \end{bmatrix} < 0$$

where,

$$\Phi_{11} = \hat{P} + (\tau_M+1)\hat{R} + \hat{N}_1 + \hat{N}_1^T + 2X$$

$$\Phi_{21} = -(\tau_M+1)\hat{R}^T + \hat{N}_2 - T_1^T X + \omega_1 X$$

$$\Phi_{22} = -\hat{P} + (\tau_M - \tau_m + 1)\hat{Q} + (\tau_M+1)\hat{R} - \omega_1 T_1 X - \omega_1 T_1^T X + \hat{S}$$

$$\Phi_{31} = \hat{N}_3 - \hat{N}_1^T - \bar{T}_2^T Y$$

$$\Phi_{32} = -\hat{N}_2^T - \omega_1 \bar{T}_2^T Y - \omega_2 T_1 X$$

$$\Phi_{33} = -\hat{Q} - \hat{N}_3 - \hat{N}_3^T - \omega_2 \bar{T}_2 Y - \omega_2 \bar{T}_2^T Y$$

$$\Phi_{41} = -\bar{T}_3 Y + \omega_3$$

$$\Phi_{42} = -\omega_1 \bar{T}_3^T Y - \omega_3 \bar{T}_3 Y$$

$$\Phi_{43} = -\omega_2 \bar{T}_3^T Y - \omega_3 \bar{T}_2 Y$$

$$\Phi_{44} = -\omega_3 \bar{T}_3 Y - \omega_3 \bar{T}_3^T Y$$

and

$$\hat{P} = M_1^{-1} P M_1^{-1}$$

$$\hat{Q} = M_1^{-1} Q M_1^{-1}$$

$$\hat{R} = M_1^{-1} R M_1^{-1}$$

$\hat{S} = M_1^{-1} S M_1^{-1}$

$\hat{N}_1 = M_1^{-1} N_1 M_1^{-1}$

$\hat{N}_2 = M_1^{-1} N_1 M_1^{-1}$

$\hat{N}_3 = M_1^{-1} N_3 M_1^{-1}$

$\hat{N}_4 = M_1^{-1} N_4 M_1^{-1}$

3.4 Summary

Chapter 3 is the main work structure of this thesis. The controller is the essential part of any control system. The controller has to take into consideration the system dynamics as well the feedback gain to control the system to automate in such a way that it should be self sufficient to take proper decisions without human interaction. Multi-agent system dynamics with incorporated packet losses is derived along with it's error dynamics.Error dynamics is an essential requirement for designing controller as the main objective of controller is reduce that error. Solving the Lyapunov equation results in final linear matrix inequalities which then solved further by LMI Solver MATLAB® toolbox, yields out the value of control gain, which is the final motive of controller

Chapter 4

Simulation results

In this chapter, the simulation results based on different conditions are presented along with detailed observations and conclusions. The consensus time is the common factor around which different conditions are studied and their effect on it.

4.1 Effect of Data Loss Rates on Consensus of MAS

Data transfer rate is defined as the ratio of number of data packets transferred through a communication channel to its maximum capacity of data packets that can be transferred over the same communication channel. Data loss rate is defined as the ratio of the difference between data packets transmitted by the transmitter and actual data packets received by the receiver to the total number of packets transmitted by the transmitter. Packet loss can reduce throughput for a given sender, either unintentionally due to network malfunction, or intentionally as a means to balance available bandwidth between multiple senders when a given router or network link reaches nears its maximum capacity. In general, throughput is the rate of production or the rate at which something can be processed. When used in the context of communication networks, such as Ethernet or packet radio, throughput or network throughput is the rate of successful message delivery over a communication channel.

When reliable delivery is necessary, packet loss increases the latency due to additional time needed for retransmission. Assuming no retransmission, the packets experiencing the worst delays might be preferentially dropped resulting in a lower latency overall at the price of a data loss. During a typical network congestion, not all the packets in a stream are dropped. This means that the undropped packets will arrive with low latency compared to the retransmitted packets, which arrive with a high latency. Not only do the retransmitted packets have to travel part of the way twice, but the sender will not realize the packet has been dropped until it either fails

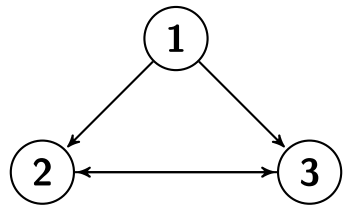

Figure 4.1: Directed graph topology of three agents

to receive an acknowledgement of receipt in the expected order, or fails to receive acknowledgement for a long enough time that it assumes the packet has been dropped as opposed to merely delayed.

The amount of packet loss that is acceptable depends on the type of data being sent. For example, for Voice over IP traffic, missing one or two packets every now and then will not affect the quality of the conversation. Losses between 5% and 10% of the total packet stream will affect the quality significantly. On the other hand, when transmitting a text document or web page, a single dropped packet could result in losing part of the file, which is why a reliable delivery protocol would be used for this purpose (i.e. to retransmit dropped packets).

In this section, the maximum permissible value of data loss rate is to be determined for the considered multi-agent system. The agents are considered to be at a distance of 5 meters from each other and the maximum permissible consensus time is limited to 40 seconds, considering the average distance that will be traveled based on their average speeds. Different examples based on different data loss rates are established by substituting the value of data loss rate r from 0 % to gradually increasing to 100 %. In between, the consensus time of all agents is observed and based on its nature, the conclusions are made.

4.1.1 Topology Consideration for Three Agents Case

The considered topology of the networked MAS is modeled by a directed graph as shown in Figure 4.1. Agent 1 is considered as a leader agent which is only capable of sending signals to the rest of the agents and agent 2 and agent 3 are the follower agents. Follower agents can receive signals from the leader agent but cannot transmit back to the leader agent. They can transmit and receive signals to each other. This

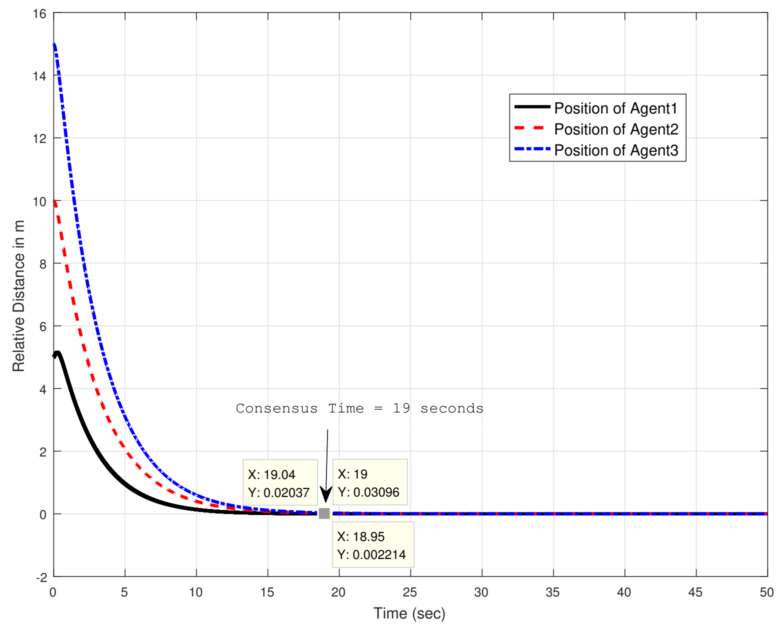

Figure 4.2: Consensus of position state of all three agents at 0% data loss rate

topology is strongly connected and can sustain any event failure during the operation. The communication weight $\omega_{ij} = 1$ is considered for all the communication channels at all the communication events. Based on Fig. 4.1, the adjacency matrix $\mathcal{A}_a$ and the Laplacian Matrix $\mathcal{L}$ are as follows:

$$\mathcal{A}_a = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \text{ and } \mathcal{L} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

The eigenvalues of $\mathcal{L}$ are 0, 1 and 3 which satisfies the condition that zero is a simple eigenvalue of Laplacian matrix $\mathcal{L}$. Hence this guarantees consensus as per algebraic graph theory.

4.1.2 Simulations Results for Different Data Loss Rates

- **Example 1:** Data Loss Rate, $r = 0$ % i.e. no data loss rate (ideal condition).

When data loss rate is kept at zero, $\bar{T}_2$ terms in the LMI get eliminated. This is the ideal condition and in real time it won't exists as there will always be some packet loss. Since this control gain is only a sufficient condition of LMI, it might not

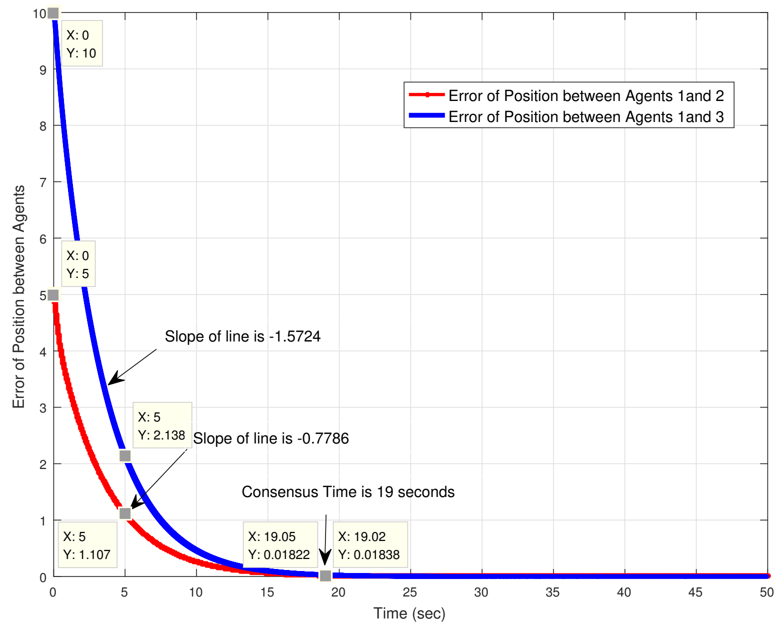

Figure 4.3: Consensus of position error state of all three agents at 0% data loss rate

be the optimal gain. However this value of control gain $\boldsymbol{k}$ = [-0.3062 -0.9907] does guarantee the consensus of the system when the packet data loss rate is no more than 0%. x_1 and x_2 are the position and velocity states of the agent respectively. Both the information states of the follower agents 2 and 3 converge to the information states of the leader agent 1.

Fig. 4.2 and 4.4 are one dimensional, meaning all agents are located in one straight line. As it can be seen from Fig. 4.2 and 4.4, the consensus time is around 19 seconds. The consensus time will always be less in this case as their is not a single packet loss.

All the information carrying packet data are received in time and the system works in highly stable environment. This can be observed by its linear reduction and steep slope in position error as shown in Fig. 4.3. From Fig. 4.4, it can be observed that the agent 1 starts from its initial position from 0 m/s to 0.9 m/s. Agents 2 and 3 move in the opposite direction to agent 1 with 2.698 m/s and 4.047 m/s, but after that they start to increase their velocity in the direction of the leader agent 1.

The MSE mean square error is another parameter through which the consensus

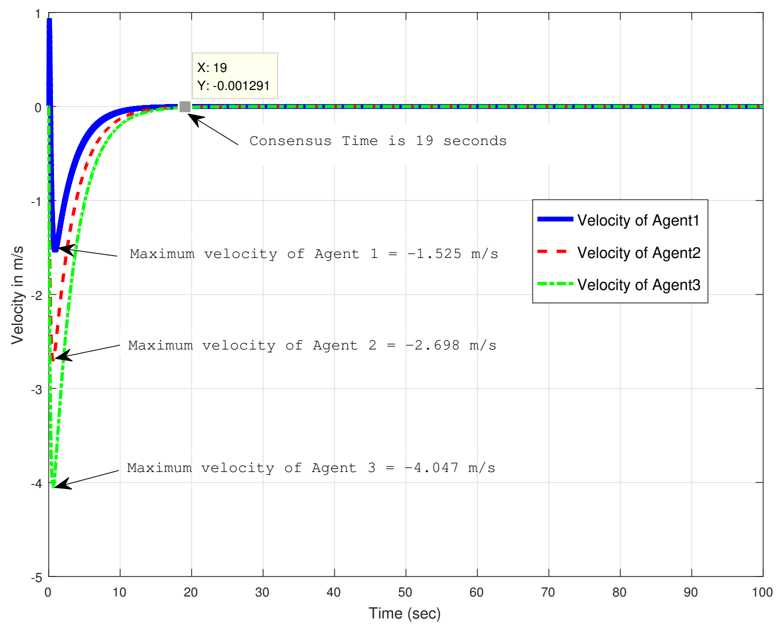

Figure 4.4: Consensus of velocity state of all three agents at 0% data loss rate

nature of the MAS can be observed. It determines the stability nature of the trajectory of agents to achieve the consensus. As seen in Fig. 4.5, the nature of the mean square error is quite simple as as same as that of position error. The mean square error (MSE) is defined as:

$$MSE = \sqrt{(x_{1i} - x_{11})^2 + ((x_{2i} - x_{21})T_s)^2} \tag{4.1}$$

where x_{i1} and x_{2i} are the current position and speed of an agent i, the i being 1,2,3,...,n. There needs to be a finite value to decide the offset distance between all the agents. In this case, the offset distance between each agents is assumed as 0.02 m. In practice, it needs to be more, but since this is theoretical setup, the least reasonable reading is considered.

For all other examples in this subsection, the conditions are maintained same except for the change of data loss rate. The data loss rate is further increased gradually in different examples.

- **Example 2:** Data Loss Rate, $r = 10$ %

In second example, the data loss rate r is set to 10%, which means that for every

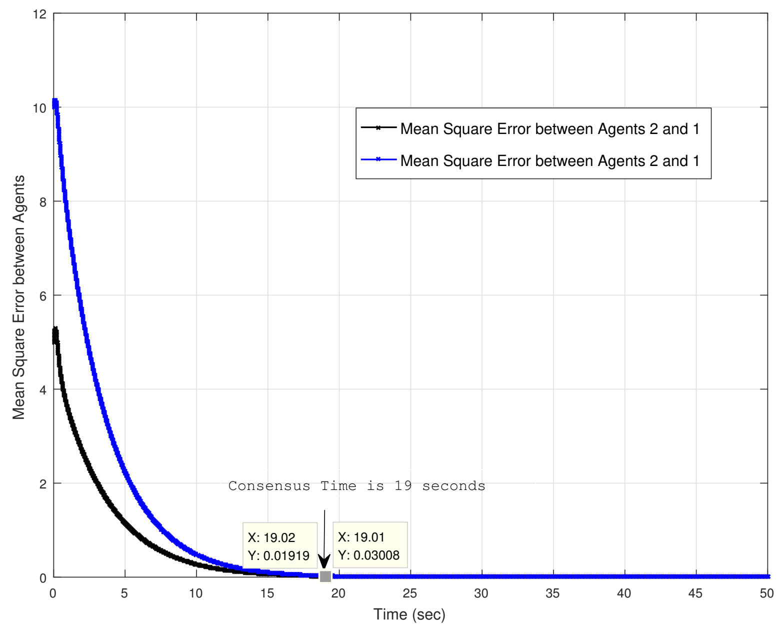

Figure 4.5: Mean square error between leader and follower agents at 0% data loss rate

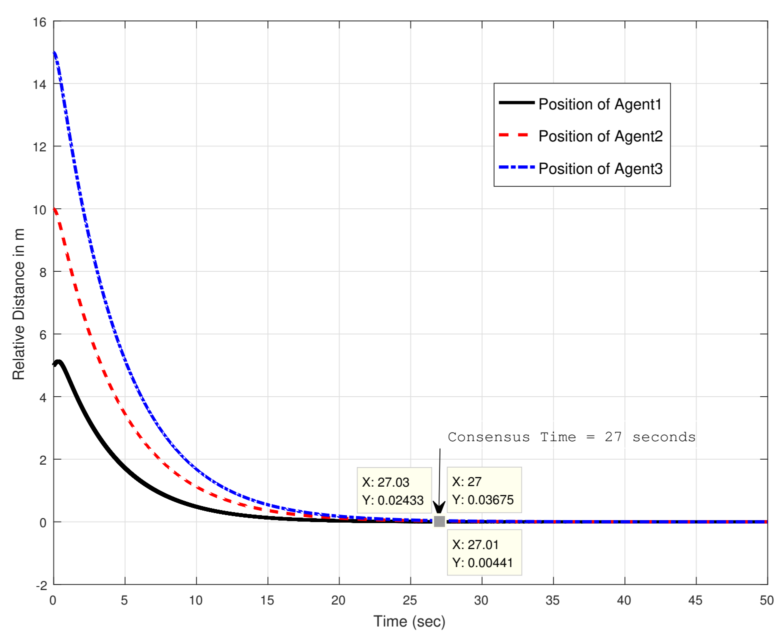

Figure 4.6: Consensus of position state of all three agents at 10% data loss rate

100 data packets transmitted, there is a leakage or loss of 10 data packets and only 90 data packets reach the final destination. The value of control gain achieved by substituting r = 10% or 0.1 in the LMI is $\boldsymbol{k}$ = [-0.1689 -0.7931].

Fig. 4.6 shows that the consensus time required for all agents is more as compared to the Example 1. The obvious reason for slow consensus is primarily the data loss rate. The loss of 10% packet rate results in slowing down the same system by 8 seconds and hence it takes 27 seconds to achieve consensus.

Fig. 4.7 clearly shows that the maximum velocity that each agent achieved in this example is less than as in the previous example, which led to slow consensus of the agents in this example. The nature of mean square error in this example is more or less same than the previous example as the difference in consensus time is very less and is not affected significantly. The mean square error will show up significant or random change in conditions when there is instability or at high data loss rates.

The maximum consensus time which can be considered as a reasonable value of achieving consensus is limited to 40 seconds. The value is decided based on the various factors like space constraints and various other constraints related to actual robots. The threshold value is fixed to 40 seconds for all the cases and examples beyond which even if there is consensus achieved, it is still not a feasible value for consensus.

- **Example 3:** Data Loss Rate, r = 20 %

The data loss rate is set to 20% i.e. $r = 0.2$ in the LMI. It is certain that the consensus time will increase as there is increase in data loss rate. Fig. 4.8 shows that the consensus time has increased compared to other examples. The consensus time for data loss rate at 20% is 36 seconds, which is near to being twice the consensus time for ideal case. The data loss rate of 20% has increased the consensus time by twice the amount and also reduced the consensus efficiency by half. Still the consensus time value is less than the threshold value of 40 seconds. Hence this example can be considered as a valid consensus case and data loss rate of 20% is acceptable for the considered MAS. So it becomes necessary to check the maximum acceptable data loss rate that the considered MAS can sustain as the consensus time in this example has reached near the threshold permissible value of consensus time. In the next example, the data loss rate is increased further to check whether it can make consensus time within permissible limit of 40 seconds.

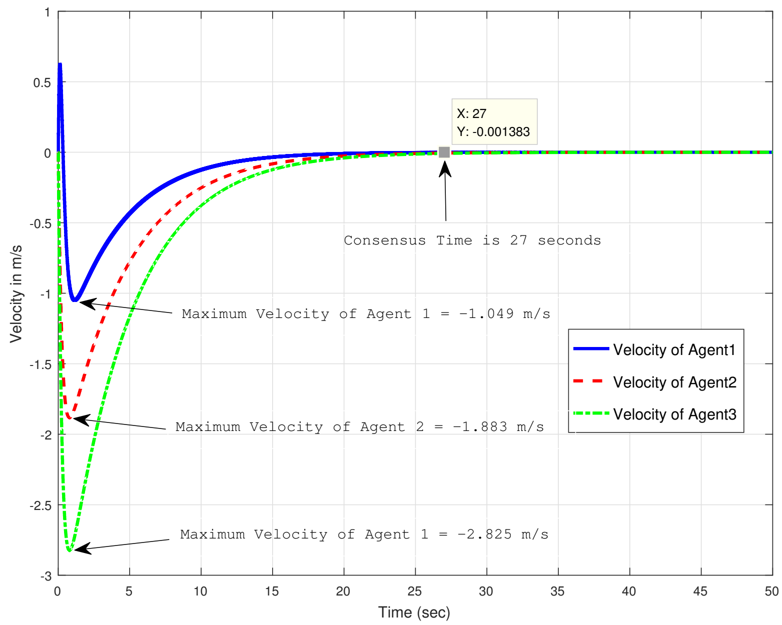

Figure 4.7: Consensus of velocity state of all three agents at 10% data loss rate

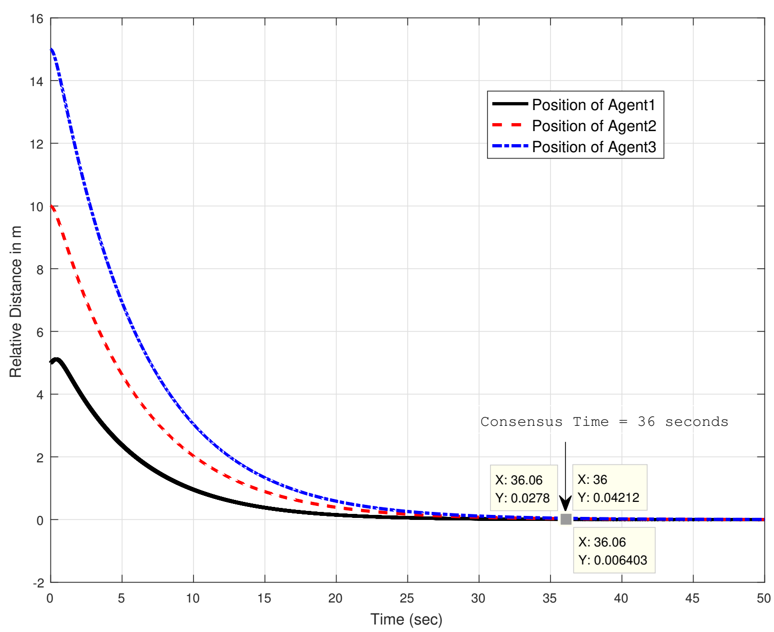

Figure 4.8: Consensus of position state of all three agents at 20% data loss rate

- **Example 4:** Data Loss Rate, $r = 30$ %

The data loss rate is increased further to 30 %, that means now value of $r = 0.3$. In communication networks, it is observed that 30% data loss rate is the maximum permissible value beyond which most of networked control systems (NCSs) are likely to expose to a temporary or permanent outage depending on applications. This can affect the system stability as well fail the operation in some cases. In this case, as it can be seen in Fig. 4.9, all agents are undergoing stable consensus at around 102 seconds. The difference between the consensus time in this example and permissible consensus time is very large. It is more than 2.5 times the permissible limit of consensus time. This explains that at $r = 30\%$, the system is stable and achieving consensus very slowly. Such a case in our considered application is equal to being redundant and hence it can be concluded from these observations that the maximum permissible value of data loss rate is between the range of 20% to 30%. The objective of this exercise is not to find this maximum permissible value of data loss rate but a certain lower bound value of data loss rate that can satisfy all the conditions of the consensus. Since there are more communication constraints to be added up in upcoming sections like constant time delays and packet data loss, the maximum permissible data loss rate that can guarantee a safe and stable working of the system is adjusted to 20%. So it is now a need to check the system stability and working under the influence of data loss rate as well as time delays in further sections.

As far as the effect of a data loss rate is concerned, the data loss rate is increased further just to observe the system behavior despite knowing that it is not going to be a stable consensus condition. So, only two cases are going to be studied for data loss rate of 80% and 98% as in Examples 5 and 6 respectively. The reason for not considering 100% data rate is very obvious as that means all the communication packets for that event are dropped and the system needs to be dependent either on another communication channel if its connected to another agent or reconsider the previous event reading.

- **Example 5:** Data Loss Rate, $r = 80$ %

Increasing the data loss rate to a gradual value like 40% or 50% will give a consensus value and give same observations like the previous example 4. So, the data loss rate is increased to 80% where chances of instability are more and the nature of

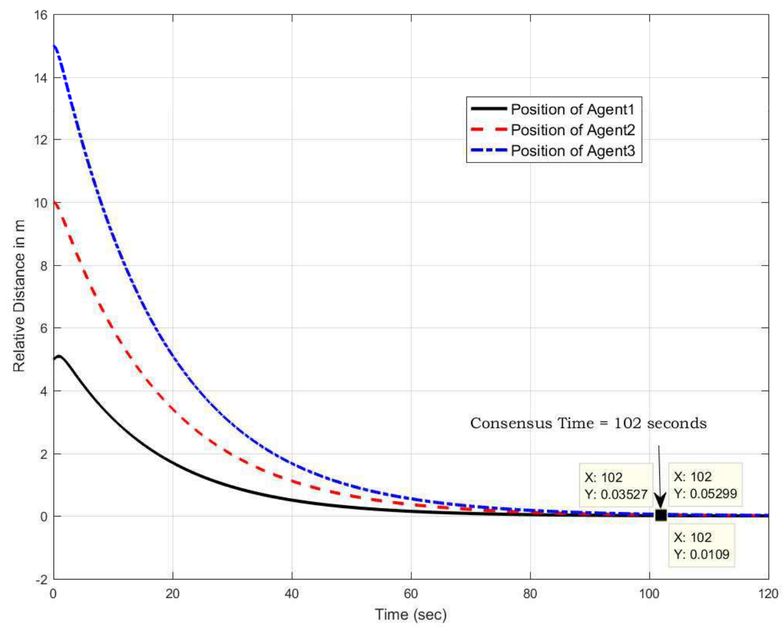

Figure 4.9: Consensus of position state of all three agents at 30% data loss rate

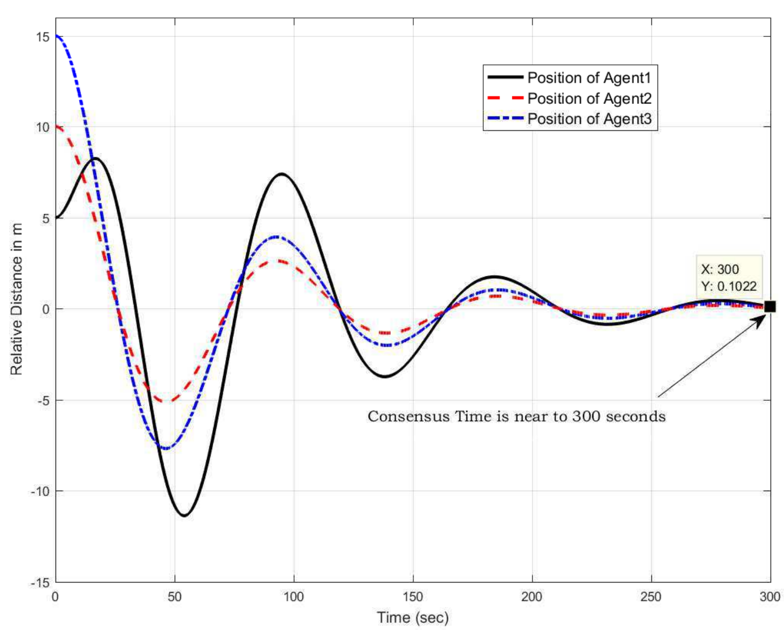

Figure 4.10: Consensus of position state of all three agents at 80% data loss rate

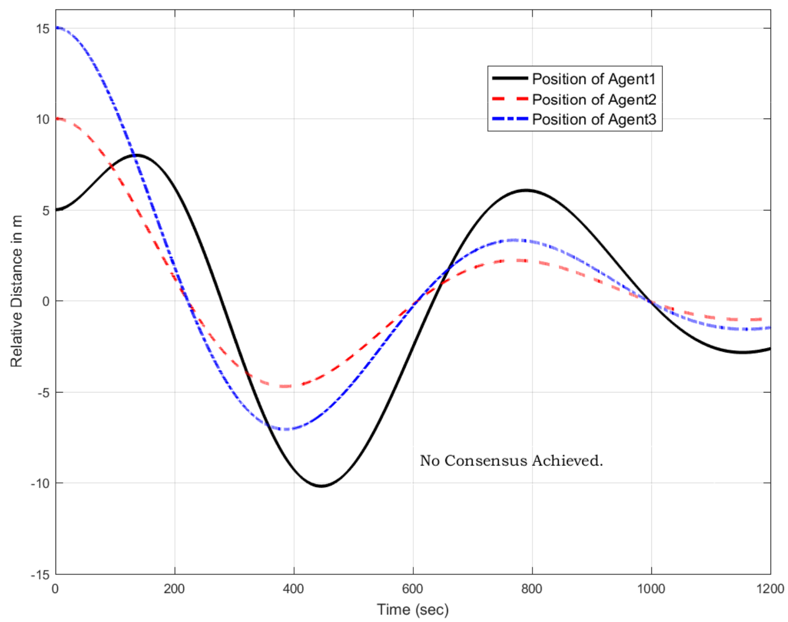

Figure 4.11: Consensus of position state of all three agents at 98% data loss rate

consensus trajectory is going to be random and sinusoidal with many fluctuations.

It can viewed in Fig. 4.10, that the consensus time has exceeded the permissible limit of 40 seconds by way higher value of 300 seconds i.e. 5 minutes which is a very sluggish working in consensus operation. To understand more about the unstable nature, it is necessary to have a look into Figure 4.12, which shows so many perturbtions in a single consensus operation. This means that the system agents are not in accord with the states of each other and at times are behaving in random manner.

- **Example 6:** Data Loss Rate, $r = 98$ %

The data loss rate is increased further to 98%, which will be the worst case scenario for a networked control system to operate. This is an impractical case. But still this example is considered to verify the control design and to fulfill the observation that as the data loss rate increases, the system efficiency decreases and reaches to a point of failure or follow some random behavior.

As r is substituted as 0.98 in LMI, Fig. 4.11 shows that the system is converging very slow. Comparing it to the real time conditions, the agents are not moving at all for a long time. This is a clear case of system failure as the agents are not able to

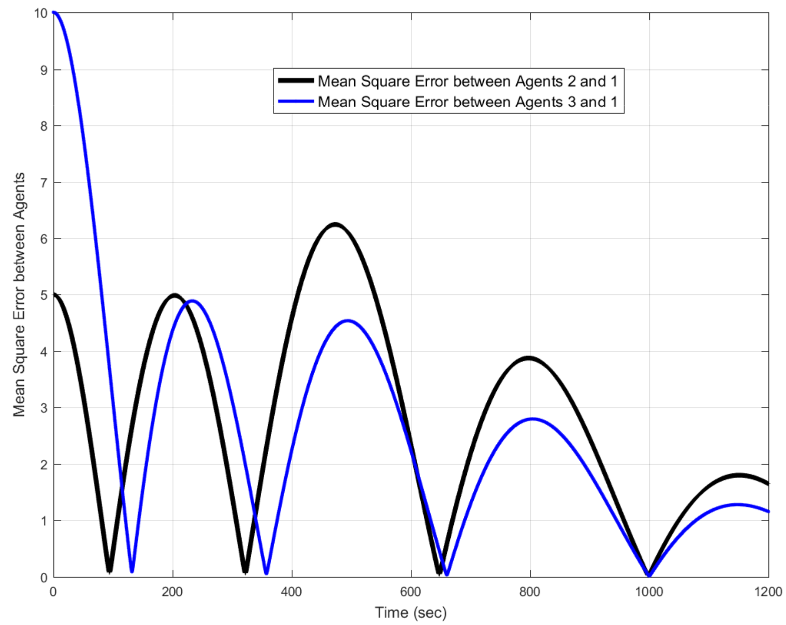

Figure 4.12: Mean square error between leader and follower agents at 98% data loss rate

achieve consensus over a long time span of 1200 seconds i.e. 20 minutes. By looking at Fig. 4.12, it is quite evident that the system is unstable and has no means of achieving consensus.

Table 4.1 and Fig. 4.13 give a proper summary of important parameters which are to be studied in the effect of data loss rate. It also states that as the data loss rate increases, the consensus time increases. Data loss rate up to 20% is endurable for the system. After that the consensus is seem to be achieved but with non-feasible consensus timings up to 30% to 40%, finally leading to randomness or non-consensus after 50% data loss rate.

4.2 Effect of Time Delay

Time delays are integral part of the practical network control systems and can never be neglected while simulating any results. In this section, the simulation results of consensus of MAS are shown considering the effect of time delays on consensus of MAS. As per discussion in Sections 3.3.2 and 3.4, it is necessary to classify the

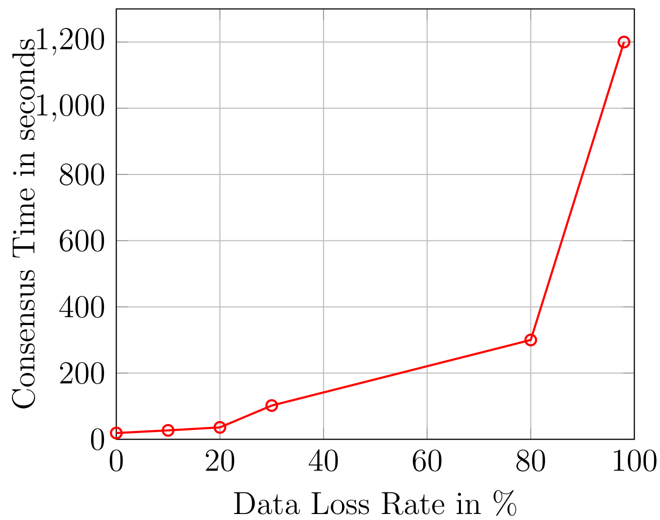

Figure 4.13: Relation between data loss and consensus time

Table 4.1: Summary of consensus at all data loss rates.

Example	Data Loss Rate (r) in %	Consensus Time (seconds)	Control Gain ($\boldsymbol{k}$)
1	0	19	[-0.3062 -0.9907]
2	10	27	[-0.1689 -0.7931]
3	20	36	[-0.1036 -0.6603]
4	30	102	[-0.0157 -0.2924]
5	80	300+	[-0.0019 -0.0057]
6	98	1200 or ∞	[-0.00084 -0.00078]

time-delays into two cases:

• Case A: Constant or fixed delay, where τ is fixed and known.

• Case B: Time-varying delay, where τ_k is an arbitrary value. It switches values between bounded minimum delay τ_m and maximum delay τ_M.

Time-delays can remain constant or vary depending upon the communication network of control systems. However, in either case, the maximum value of time delay is known. There needs to be the maximum permissible value of time delay that should be allowed in the system beyond which if there is any data packet transmission, then it should be considered as data loss. The maximum permissible time delay in this system is 0.1 seconds or 100 milliseconds. If a packet takes more than 0.1 seconds to transmit from one agent to another agent, then it is considered as a packet loss in this case. Hence in Case A, the maximum value of time delay is consider as 0.1

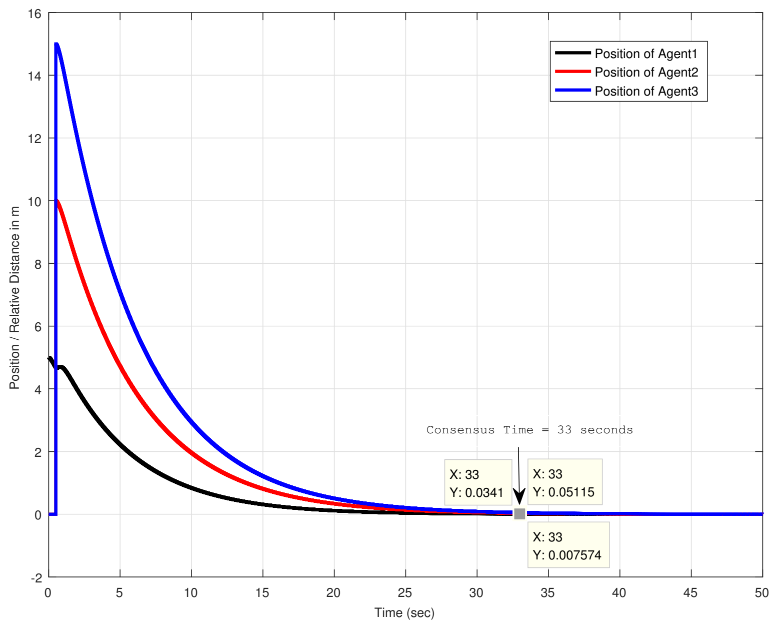

Figure 4.14: Consensus of position state of all agents at 10% data loss rate and $\tau = 0.001$ seconds

seconds. Also the data loss rate of 10% is considered along with time delays to make sure that there are packet losses as well as time delays in the system. The topology is the same as that shown in the Fig. 4.1 for both cases of A and B.

4.2.1 Case A: Simulation Results for Constant Time Delay

In this subsection, simulation results for closed loop controlled MAS is studied for 10% data loss rate and time delay is kept fixed for a operation. Different values of fixed time delay have been considered for consensus operation of the MAS through these different examples.

• **Example 1:** Data Loss Rate, r = 10 % and Time delay $\tau = 0.001$ seconds or 1 millisecond.

At 10% data loss rate, the time for three agents to achieve consensus is 27 seconds without any time delays. However the presence of time delays in the system are going to affect the same system adversely. This means that the time taken for the same MAS will increase to achieve consensus between them. It is necessary to keep the consensus time within the permissible limit of 40 seconds. It took 36 seconds for all agents to reach a consensus when there was no time delay and 20% data loss rate, which is very near to permissible value of 40 seconds. That is the another reason for considering data loss rate of 10% as it there is still a gap of 13 seconds to achieve a consensus. The value of τ is substituted as 0.001 seconds in LMI, which gives the following value of control gain $\boldsymbol{k}$ = [-0.1319 -0.7816].

Fig. 4.14 shows that the constant time delay of 0.001 seconds has affected the overall consensus time. The consensus time is shifted by nearly six seconds. The new consensus time after considering the time delay of 0.001 seconds is 33 seconds. Time delay of 0.001 second is of very small order, but cannot be neglected in case of network communication systems. However, it is necessary to check the system stability at higher values of time delays which may affect the system stability or at least slow down the consensus ability of the MAS.

The maximum time delay that we are considering for our system to sustain is 0.1 seconds. So it is better to divide this time range in five intervals of 1 ms or 0.001 seconds, 5 ms or 0.005 seconds, 10 ms or 0.01 seconds and 100 ms or 0.1 seconds. These four equal intervals form the four examples out of which Example 1 is already shown. As the time delay increases, it is necessary to check the nature of consensus time, whether it will increase or decrease and the magnitude of that change.

- **Example 2:** Data Loss Rate, r = 10 % and Time delay τ = 0.005 seconds or 5 milliseconds.

Increasing the time delay by five times may affect the system stability. The value of τ is substituted as 0.005 seconds in the LMI to get the value of control gain. Figure 4.15 shows that the consensus time has increased for the same system by 1.5 seconds, which is not a much difference considering delay increase by five times from previous example. This is showing that the system is recuperating well for low values of time delays despite they being order of five times. It is necessary to check whether the same system can cope up with higher values of delays. Let us check whether the derived LMI for the system works for twice the time delay considered in this example i.e. when τ = 0.01 second or 10 milliseconds in further example.

- **Example 3:** Data Loss Rate, r = 10 % and Time delay τ = 0.01 seconds or 10 milliseconds.

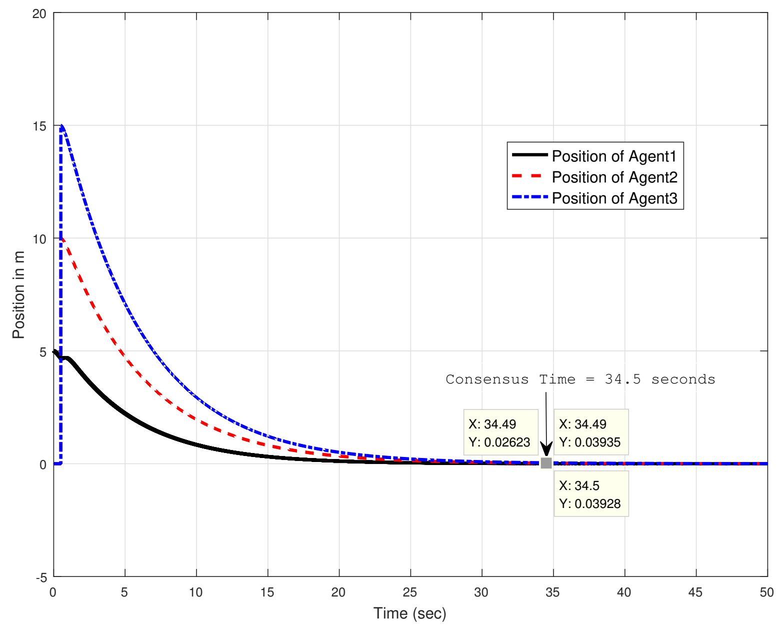

Figure 4.15: Consensus of position state of all agents at 10% data loss rate and $\tau =$ 0.005 seconds

Increasing the value of time delay to 0.01 second will definitely lower down the system performance by increase in consensus time. The control gain value achieved is $\boldsymbol{k}$ = [-0.1007 -0.6102]. From Fig. 4.16, the consensus time taken by all three agents is near to 37 seconds which is like 2.5 seconds more than previous case. Even though the time delay was doubled in this example as compared to previous example, the considered MAS is able to achieve consensus within the permissible value of 40 seconds. At the same time, the difference between the consensus time in Example 1 and Example 3 is very less i.e. about 4 seconds which is very much considerable. It depends further on MAS and the LMIs that we have derived, whether they can sustain more higher time delays. Also it is necessary to check whether they are able to achieve consensus before the permissible limit of 40 seconds.

• **Example 4:** Data Loss Rate, r = 10 % and Time delay τ = 0.1 seconds or 100 milliseconds.

Increasing the time delay by 10 times has resulted into an increase of consensus time which has shifted about 4 seconds from the previous example. Also by comparing

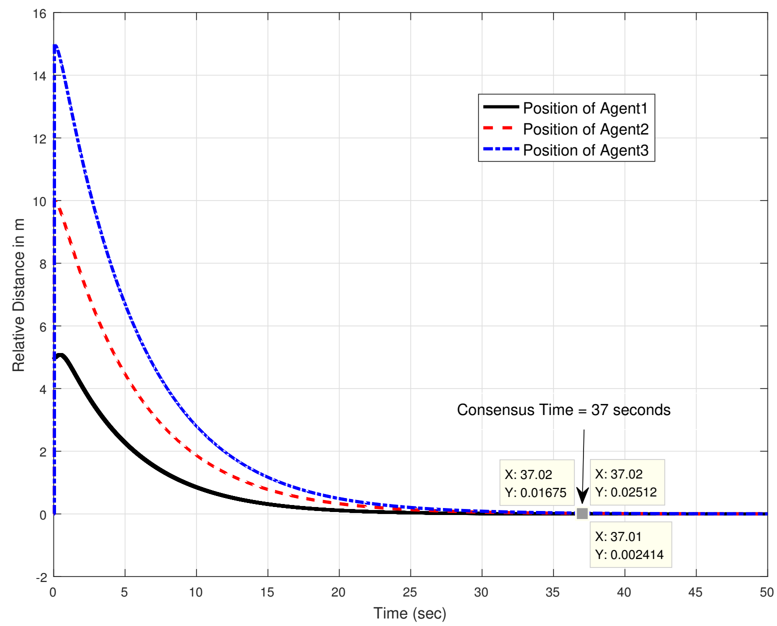

Figure 4.16: Consensus of position state of all agents at 10% data loss rate and $\tau = 0.01$ second

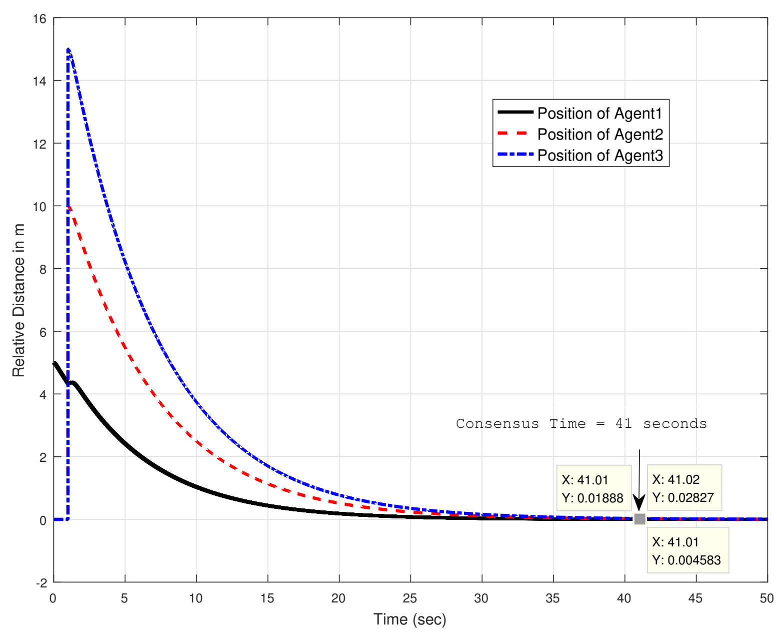

Figure 4.17: Consensus of position state of all agents at 10% data loss rate and $\tau = 0.1$ second

it with Example 1, the overall consensus time has increased by 8 seconds. Overall, the system consensus efficiency has decreased by a total of 25% from Example 1 to Example 4. The consensus time taken by all agents when there is 10% data loss rate and 0.1 second fixed time delay is 41 seconds which surpasses the permissible limit of consensus time. However, as discussed earlier, this is not an optimal type of control, so there will always be a corridor of development of control gain which will certainly reduce the consensus time. Also the difference between the achieved consensus time and permissible consensus time limit is very less of 1 second. So this case can still be considered successful and the there are no uncertainties or abnormalities in the system that suggest that system will not be stable.

After looking at all examples, it has been confirmed that the introduction of time delays does affect the consensus capability of the MAS considering packet data loss of 10% at the same time. Comparing the MAS of Example 4 in Section 4.2.1 with the Example 1 in Section 4.1.2, the difference in the consensus time has been significant and the consensus time is doubled. However, this will be the worst possible case that the considered MAS may need to sustain for its successful operation under the assumed conditions. The designed MAS may achieve consensus in a stable manner even after an increase in the values of constraints like data loss rate and time delays, however the time required for the consensus will be very high and unacceptable for the desired operation. In the design, there is always a parameter called a 'safety factor'which need to be followed. In our designed MAS system along with it's controller, the example 4 of Section 4.2.1 is the safety factor. Beyond these values of constraints, there is no guarantee of successful operation if we consider 40 seconds as our permissible limit of consensus time. So for any run time conditions, the data loss rate should not exceed 10% and 100 milliseconds of time delay.

Table 4.2: Summary of consensus at fixed time delay and 10% data loss rate.

Example	Time Delay (milliseconds)	Consensus Time (seconds)	Control Gain ($\boldsymbol{k}$)
1	1	33	[-0.1319 -0.7816]
2	5	34.5	[-0.1015 -0.6210]
3	10	37	[-0.1007 -0.6102]
4	100	41	[-0.1527 -0.9982]

Graph in Figure 4.18 shows that the slope of the consensus time increases at

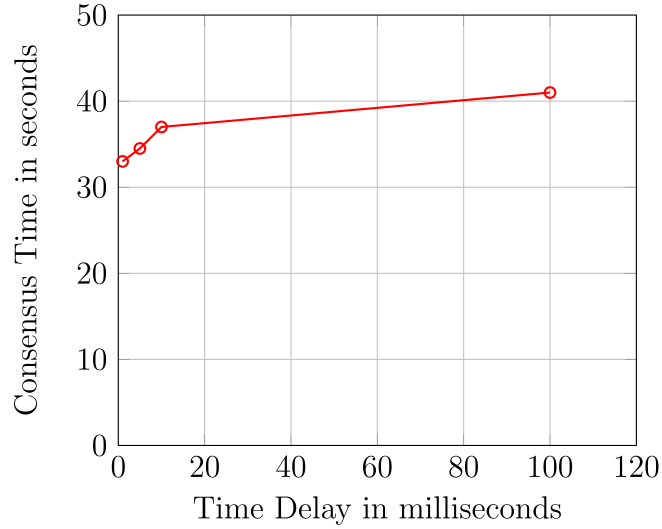

Figure 4.18: Relation between fixed time delay and consensus time at 10% data loss rate

lower time delays whereas decreases at higher time delays, which means the system capability to reduce the consensus time is more for higher values of time delays as compared to lower values of time delays.

4.2.2 Case B: Simulation Results for Time-Varying Delay

In this subsection, simulation results for closed loop controlled MAS is studied for 10% data loss rate and time delay is bounded and is varying between the lower and upper bounds for consensus operation. Different values of upper and lower bounds of time-varying delays have been considered for consensus operation of the MAS through these different examples. At 10% data loss rate, the consensus time was 27 seconds. Introducing the fixed time delay into the system increased the overall consensus time of the system to range from 33 seconds to 41 seconds for different examples. In Case B, the upper most bound and the lower most bound that the system will be subjected to is 0.1 second or 100 ms and 0.001 second or 1 ms respectively. However, the examples would be based on the different range between these upper and lower bound limits. Three examples will be considered for case B, in which the first example is between 1 millisecond to 10 milliseconds, the second example is between 10 milliseconds to 100 milliseconds, the third example is increase in range from 1 millisecond to 100

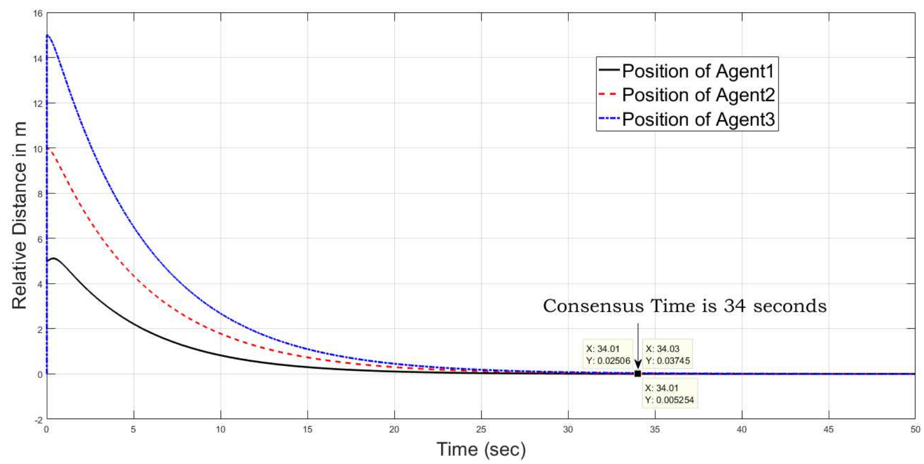

Figure 4.19: Consensus of position state of all agents at 10% data loss rate and τ_m = 0.001 second and τ_M = 0.01 second

millisecond. The range in first two examples is of same order of 10, but the lower and upper bounds are different. The range in third example is very high of order 100. It is necessary to check that the designed MAS and its controller is able to cope up with dynamic varying time delays and the amount by which it has affected the consensus ability of the system.

- **Example 1:** Data Loss Rate, r = 10 % and Time-varying delay range is 0.001 second or 1 millisecond to 0.01 second to 10 milliseconds

In this example, the lower bound of the time delay is 1 millisecond and the upper bound of the time delay is 10 milliseconds as shown in Figure 4.20. The order of the range is 10. This means that the value of τ_m is substituted as 0.001 and the value of τ_M is substituted as 0.01 in LMI to get the value of control gain.

Figure 4.19 shows that the consensus time taken for all agents at 10% data loss rate and time varying range between 1 millisecond and 10 milliseconds is around 34 seconds. The time- varying delay is same at any instant between follower agents and leader agent. In Example 2, the range order is kept same but is operated at higher values of time delays.

- **Example 2:** Data Loss Rate, r = 10 % and Time-varying delay range is 0.01 second or 10 millisecond to 0.1 second to 100 milliseconds.

In this example, the lower bound of the time delay is 10 millisecond and the upper

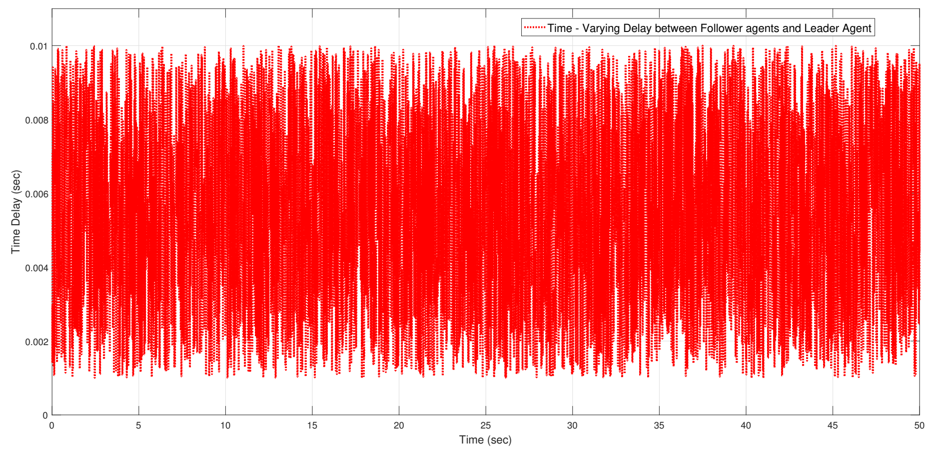

Figure 4.20: Time-varying Delay between $\tau_m = 0.001$ second and $\tau_M = 0.01$ second

bound of the time delay is 100 milliseconds. The order of the range is kept same as previous example. This means that the value of τ_m is substituted as 0.01 and the value of τ_M is substituted as 0.1 in LMI to get the value of control gain.

Figure 4.21 shows that all agents reach consensus at 36 seconds which is late by meager two seconds as in previous example. This shows that the higher bound time delay limits with same order do not deviate the consensus time very badly. It is now necessary to check the effect on consensus time by increasing the order of the time delays. The maximum order that can be tried is of 100 between 1 millisecond and 100 milliseconds. So, in example 3, the range order is increased to 100, than in previous examples.

- **Example 3:** Data Loss Rate, $r = 10$ % and Time-varying delay range is 0.001 second or 1 millisecond to 0.1 second to 100 milliseconds.

In this example, the lower bound of the time delay is 1 millisecond and the upper bound of the time delay is 100 milliseconds. The order of the range is kept same as previous example. This means that the value of τ_m is substituted as 0.001 and the value of τ_M is substituted as 0.1 in LMI to get the value of control gain.

From Figure 4.22, the consensus time taken by all agents is 46 seconds, which is ten seconds more than in the previous example. Increasing the order or range between the time delay has affected the consensus performance significantly. However the consensus nature is quite stable as in previous examples which means that there is

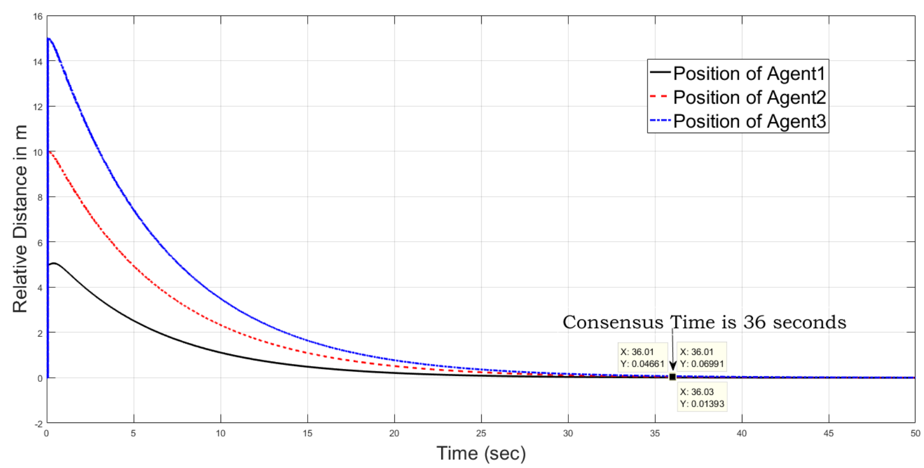

Figure 4.21: Consensus of position state of all agents at 10% data loss rate and τ_m = 0.01 second and τ_M = 0.1 second

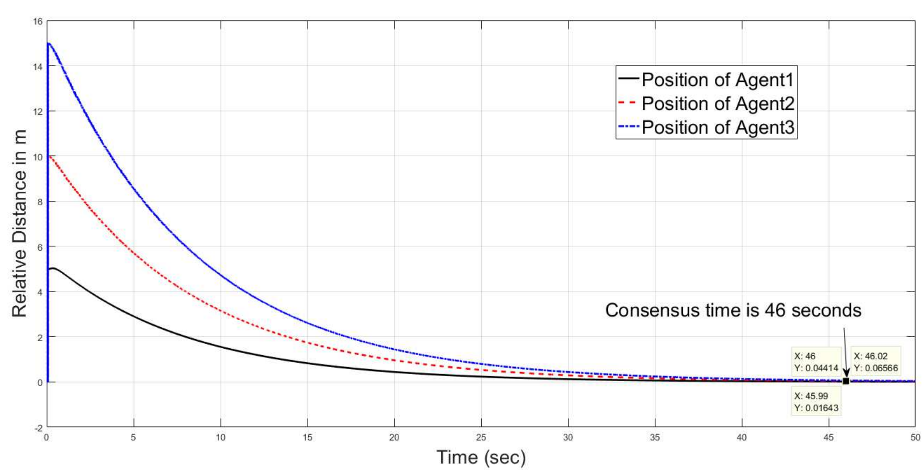

Figure 4.22: Consensus of position state of all agents at 10% data loss rate and τ_m = 0.001 second and τ_M = 0.1 second

scope of improvement in consensus timing by tuning the controller to achieved value of control gain. This result has made to conclude that at higher order delays, it is must to consider a certain value or extra range above the permissible consensus limit. So, in Case B, the permissible limit of consensus time is increased by ten seconds in order to ensure safe operation.

Table 4.3: Summary of consensus at time-varying delay and 10% data loss rate.

Example	Time Delay Range (milliseconds)	Consensus Time (seconds)
1	1-10	34
2	10-100	36
3	1-100	46

Table 4.3 gives a summary of these three examples considered, which simply states that the consensus time is not affected much at higher operating values of time delays as compared to lower operating value of time delays as long as the order of range is kept same or reduced. However, the system consensus performance is affected severely if the order of time-delay range is increased. Till now, in all the examples, the range of time-varying delays was less than sampling time i.e. $\tau_M - \tau_m \leq T_s$. The sampling time as discussed in Section 3.3.1 is kept fixed at 0.1 seconds. A time delay of 0.1 seconds is quite high in application of MAS. Still, to verify the consensus of MAS for $\tau_M - \tau_m \geq T_s$, an extra example is studied for $\tau_M - \tau_m \geq 0.1$ seconds at same other conditions.

- **Example 4:** Data Loss Rate, $r = 10$ % and Time-varying delay range is 0.001 second or 1 millisecond to 0.5 f to 500 milliseconds i.e. $\tau_M - \tau_m \geq 0.5$ seconds.

In this example, the lower bound of time delay is kept again at 1 millisecond and the upper bound of time delay is increased to 500 milliseconds or 0.5 seconds. From Fig. 4.23, it is quite evident that the increase in upper bound of time delay by five times has seriously affected the consensus time. The consensus time has increased from 46 seconds in Example 3 to 103 seconds in Example 4, which means there is increase of 57 seconds to reach consensus. This concludes that the system is capable to achieve consensus for time delay of 500 milliseconds but affects the system performance badly by increasing the total consensus time. The value of consensus time achieved in this example is much more than the permissible limit of 50 seconds.

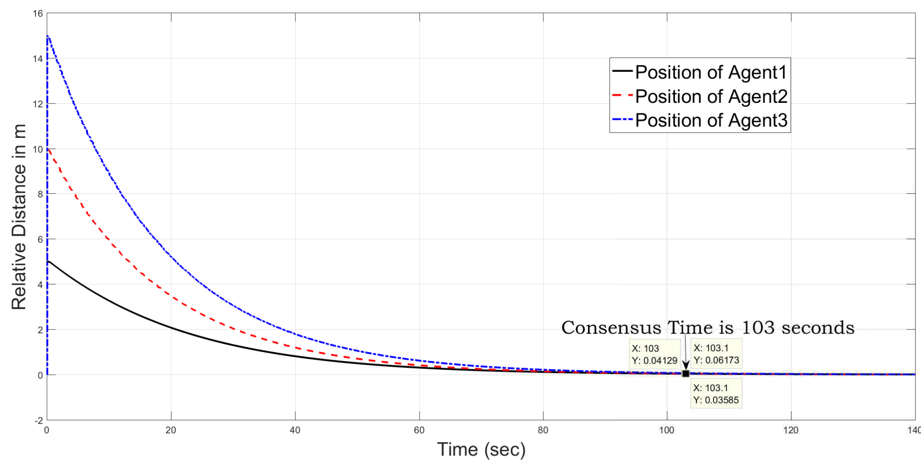

Figure 4.23: Consensus of position state of all agents at 10% data loss rate and τ_m = 0.001 seconds and τ_M = 0.5 seconds

It is not feasible to consider the consensus time achieved in this example. Hence, for the designed control system, the permissible value of maximum time delay is 0.1 seconds or 100 milliseconds or equal to sampling time T_s, beyond which the value of time delay will be considered as a packet loss.

4.3 Effect of Increasing the Number of Agents

The system dynamics and the controller proposed for the same is valid for n number of the agents. It is necessary to check the simulation results on higher number of agents and compare them with the results discussed with three agents case in Section 4.1 and 4.2. In this section, only two cases are considered, one without time delay and other with constant time delay. Results with time-varying delay are not shown because it is already evident from the Section 4.2.2 that the consensus time increases in the case of time-varying delays. But it is still not evident about the effect on consensus time by increasing the number of agents. There is a mystery about the trade off between the communication time and error processing time between the agents, which can be solved by studying the consensus results based on increasing the number of agents. Increasing the number of agents will definitely increase the communication time between them. However, if the new increased agents topology is strongly connected, there is a chance that the leader agent is more informative and

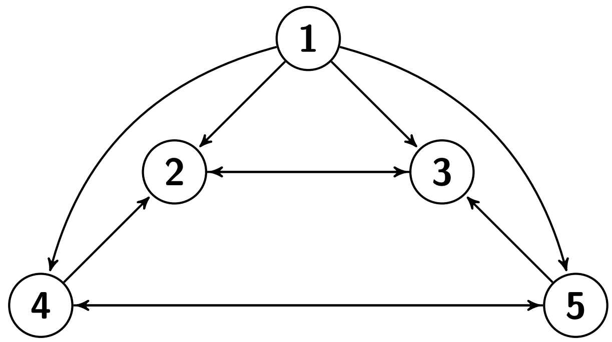

Figure 4.24: Directed graph topology of five agents

may help in decreasing the error sooner than in earlier cases discussed. In any ways, it is necessary to consider a strongly connected topology for this case. A topology of five agents is decided to be considered to study the effect on consensus time.

4.3.1 Topology Consideration for Five Agents Case

The considered topology of the networked MAS is modeled by a directed graph as shown in Figure 4.24. Agent 1 is considered as a leader agent which is only capable of sending signals to the rest of the agents and agents 2, 3, 4 and 5 are follower agents. Follower agents can receive signals from the leader agent but cannot transmit back to the leader agent. They can transmit and receive signals to each other. This topology is strongly connected and can sustain any event failure during the operation. The communication weight $\omega_{ij} = 1$ is considered for all the communication channels at all the communication events. It is also assumed that the system is free of time delays. Based on Figure 4.24, the adjacency matrix $\mathcal{A}_a$ and the Laplacian Matrix $\mathcal{L}$ are as follows:

$$\mathcal{A}_a = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } \mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

The eigenvalues of $\mathcal{L}$ are 0, 1, 2, 3 and 4 which satisfies the condition that zero is a simple eigenvalue of Laplacian matrix $\mathcal{L}$. Hence this guarantees consensus as per algebraic graph theory.

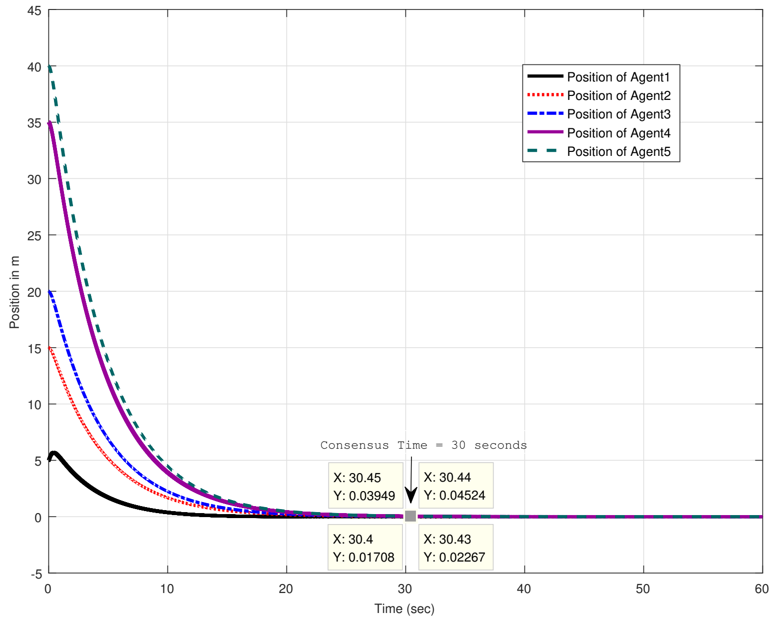

Figure 4.25: Consensus of position state of all agents at 10% data loss rate and no time delay

4.3.2 Simulations Results for Five Agents Case Without Time Delay

In this subsection, the consensus of five agent case is considered for 10% data loss rate and without any time delay. So in this case, r is substituted as 0.1 and value of time delay τ is substituted as zero in the LMI. The values of all variables in LMI are substituted directly in MATLAB® LMI Toolbox program which solves the entire LMI to get the suitable value of control gain $\boldsymbol{k}$ = [-0.1728 -0.8103]. In Fig. 4.25, the consensus time taken for all five agents is 30 seconds, which is three seconds more as in case of three agents in same conditions. Increasing two agents would had impacted a significant increase in the communication time and processing time between the agents. But with better and strongly connected topology, the average error processing time must have reduced that impact and lowered down the total consensus time to a significant amount. The increase in three seconds is very less as compared to the impact that would had felt in case the topology was not strongly connected. In previous sections, the threshold or permissible value of consensus time was limited to 40 seconds. But as it is pre-judged in this Section, that there will be

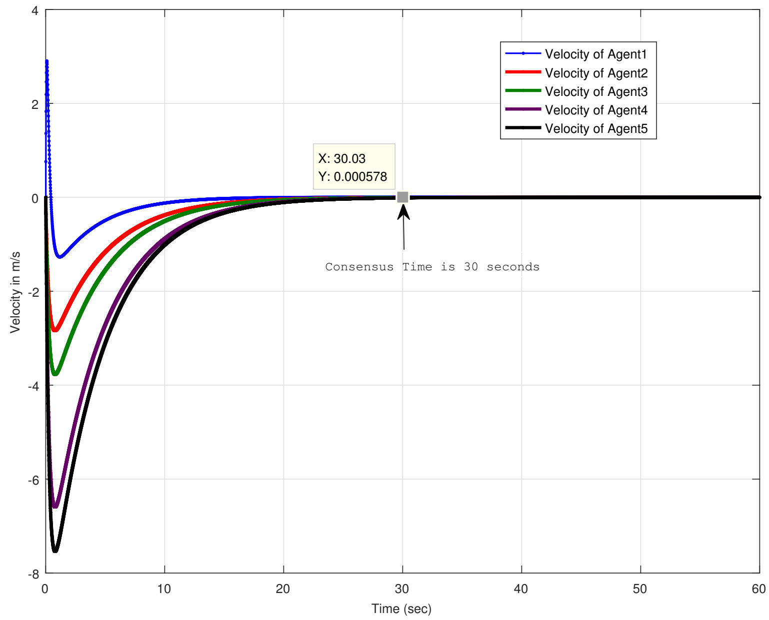

Figure 4.26: Consensus of velocity state of all agents at 10% data loss rate and no time delay

a increase in consensus time, the threshold value of consensus time is shifted to 50 seconds from 40 seconds.

Such iterations keep controllers busy and take away some time for consesus processing of the same.

From Fig 4.26, it can be observed that the maximum velocity that an agent achieves is increased by twice the amount as it was in example 2 of Section 4.1.2. The fifth agent in this case reaches velocity of 7.7 m/s to catch up with all agents whereas third agent in section 4.1.2 reaches a maximum of 2.8 m/s to achieve consensus. In this case, the third agent reaches upto 3.85 m/s to achieve consensus. So the maximum velocity value has increased in this case to achieve consensus to nearby timings. This suggests that there are many iterations that are taking place while reducing errors which consume the consensus time.

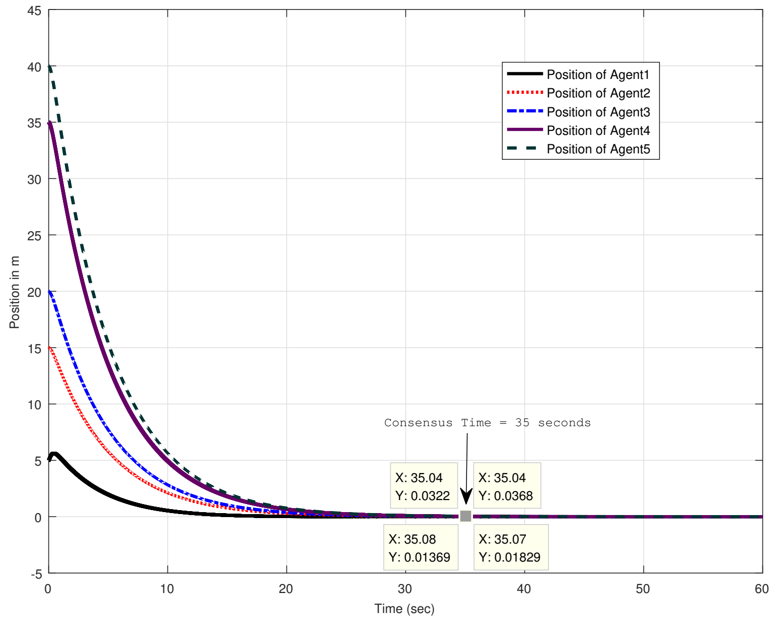

Figure 4.27: Consensus of position state of all agents at 10% data loss rate and constant time delay of 1 millisecond

4.3.3 Simulations Results for Five Agents Case With Constant Time Delay

In previous subsection, the value of time delay τ was substituted as zero in LMI. However in this subsection, there is consideration of constant time delay of 1 millisecond or 0.001 second. So the comparisons of this subsection are to be drawn with the example 1 of section 4.2.1. Rest all conditions are same like in previous subsection 4.3.2.

From Fig. 4.27, it is observed that the all agents achieve consensus at 35 seconds at constant time delay of 1 millisecond. Conditions in this case are similar to conditions in example 1 of section 4.2.1 except for the number of agents. Increasing the number of agents increased the consensus time by meager 2 seconds. Considering a increase in number of agents by two should have increased the consensus time by a considerable amount. However, the topology in five agent case as shown in Fig. 4.24 is strongly connected as compared to topology in three agent case as shown in Fig. 4.1. This has reduced the consensus time increment to 2 seconds. Table 4.4 shows the comparison

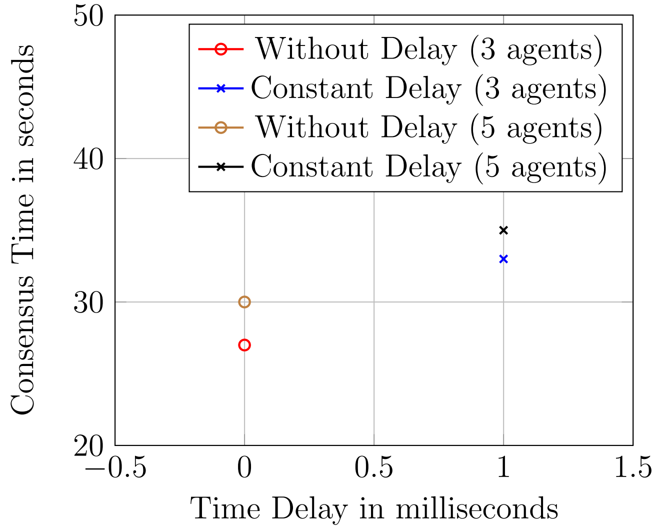

Figure 4.28: Comparison between the three agent case and five agent case for no delay and constant delay case

between the three agent case and five agent case for no delay and constant time delay case.

Table 4.4: Comparison of summary between three agent case and five agent case for no delay and constant delay case

Example	Consensus Time for Three Agent Case (seconds)	Consensus Time for Five agent Case (seconds)
Without Delay	27	30
Constant Time Delay (1 millisecond)	33	35

Fig. 4.28 summarizes the comparison between the consensus time increment for three agents and five agents topology for no time delay and constant time delay case. The consensus time increases with the increase in number of agents as well as increase in time delay. The nature of this increase for three and five agents is quite parallel as it can be seen in Fig. 4.28, which concludes that there is some trade off between the communication time and error processing time in case the topology is strongly connected to keep the consensus within the limits.

4.4 Summary

In this Chapter 4, the necessary observations of the controller behavior on the system to tackle constraints like time delays and packet data losses are studied. In Section 4.1.2, only data loss rates are considered in the system assuming that there are no delays. It was clearly observed from all the examples and Table 4.1 of Section 4.1.2, that as the data loss rate increases, the consensus time increases. The rate of consensus is fast at the start, then it goes on decreasing till a point, where there is no valid or feasible consensus.

The next condition taken in case was further divided in two conditions, one with constant time delay and at fixed data loss rates and another with time-varying delay at fixed data loss rates. It was clearly observed from the four examples and Table 4.2 of Section 4.2.1, that at fixed data loss rate of 10%, the consensus time increases for constant time delay case if we increase the time delay till it's sampling period. Once it passes the sampling period, the consensus is achieved, but the consensus timing is not a feasible solution. So it is better to keep the time delay value less than or equal to sampling time. For time-varying case, the range between the upper bound and lower bound of the time delays is important. For a fixed range at lower bounds, the consensus time is less as compared to the same fixed range at higher bounds. If the range between the upper and lower bound is less, the consensus time taken is less. If the range is high between the upper and lower limits, the consensus time increases. This is observed from Table 4.3 and three examples considered in Section 4.2.2.

The last condition considered in this Chapter 4 is effect of number of agents to the entire system. In Section 4.3, total two subsections were considered. First condition was the increase in number of agents for a fixed data loss rate and no time delay and the second condition umber of agents to five for constant time delay and fixed data loss rates. Certainly there was increase in the consensus time for increase in number of agents but for time delay there was not difference observed as the topology becomes stronger by increasing the number of agents.

Chapter 5

Experimental results

5.1 Hardware Setup

An experimental verification is conducted by using two Pioneer P3-DX robots as shown in Fig. 5.1. Pioneer P3-DX has 16 ultrasonic sensors of which 8 sensors are located in front and 8 sensors are located in rear. It has a two wheel differential drive system which is controlled by motors. Also it has an independent rear balancing caster. This motor has a 500 tick encoder. The maximum rated speed of the robot is 1.6 m/s and the rotation speed is 140 deg/s at maximum payload of 23 kg. A local wireless network over robots is created by a model DIR-615 D-Link Router. These robots are connected to laptops or local computers which are then connected to the wireless network. The hardware connection between the local computer and Pioneer robot is done with the help of USB to serial converter as shown in Fig. 5.2. This serial cable acts as a connecting media between the local computer and mobile robot. The data transmission between the robot and the local computer is wired, thus ensuring reliability. The time delay in this case is zero and no chance of packet loss. However the laptops present on each robot is connected to virtual leader local computer over a wireless network. The experimental part was carried out in basketball court of Dalhousie University Sexton Campus. The Pioneer robot P3-DX

Figure 5.1: Pioneer P3-DX mobile robot from Adept Robotics

Figure 5.2: Connection between Pioneer Robot and local computer

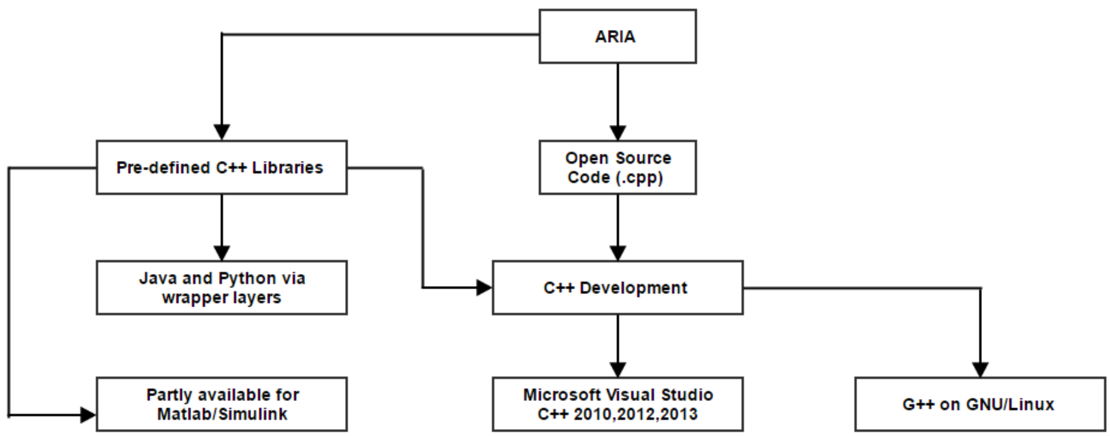

Figure 5.3: ARIA infrastructure

needs a very smooth surface to operate and basketball court is well polished for it's successful operation.

The translation between the mobile robots and local computer is made possible with the help of software development kit called ARIA. Adept Mobile Robots' ARIA is a C++ library for all Mobile Robots platform. ARIA can dynamically control the mobile robot's velocity, heading, relative heading and other motion parameters either through simple low level commands or through its high level actions infrastructure. ARIA also receives back the position estimates, sonar readings and all other current operating data sent by robot platform. ARIA also includes a library called ArNetworking which implements an extensible infrastructure for easy remote network operations for robots, user interfaces, and other networked services. Through a server executing on the robot's PC, ArNetworking-enabled clients connect from another computer on the network to get data and issue commands. So TCP is the best among two for this experimental environment.

A summary of the ARIA development structure is shown in Fig. 5.3. ARIA packages come with an open source code environment and pre-defined C++ libraries. These C++ libraries are accessible for Java and Python platforms along with basic C++ platforms like Microsoft Visual Studio C++. In the performed experiment, the C++ development is done by using Microsoft Visual Studio C++ 2010 Express Edition. The coding is generated in open source (.cpp) file and the solution is build by using Visual Basic Studio C++ 2010 Express Edition compiler to get an executable (.exe) file which finally runs the mobile robot. ArSocket command has two options of choosing internet protocol suite, one is called the User Datagram Protocol (UDP) and other one is called as Transmission Control Protocol (TCP). TCP is the most reliable internet transmission protocol, and can retransmit any dropped packets and buffer out-of-order packets to be able to redeliver the original the original data stream in the proper order to the receiver. At the other end, UDP emphasizes reduced latency over reliability. Therefore it does not guarantee successful transmission. The reason for choosing TCP is that the data loss rate can be preset by a function called $rand()\%100$, which randomly generates integer from 0 to 100 integrates. For example, if the data loss rate is required to 20%, then it is just simply set $rand()\%100 \geq 80$. It means 80% data can be successfully transmitted.

5.2 Pioneer Robot Modeling

As discussed in Chapter 4, all figures are one dimensional and they only show the system convergence. In this section, an actual Pioneer mobile robot is modeled as an agent. Therefore each agent represents a robot and then the proposed controller is implemented to study the performance in a real hardware model. In [3], the hand position of the the robot as the point $h \triangleq [h_x, h_y]^T$ that lies at a distance L ($L \neq 0$) along the line that is normal to the wheel axis and intersects the wheel axis at the center point $\mathbf{x} \triangleq [x_x, x_y]^T$, as shown in the Fig.5.4. The orientation angle (θ) is zero at 0 deg and counter-clockwise rotation is defined as positive.

$$\dot{x_x} = vcos(\theta)$$
$$\dot{x_y} = vsin(\theta) \tag{5.1}$$

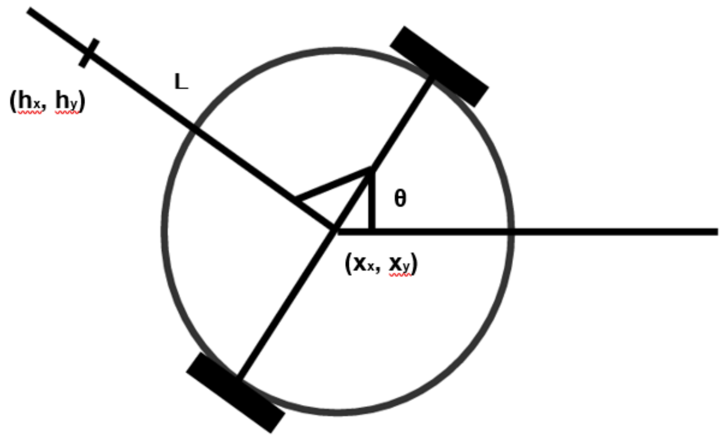

Figure 5.4: Hand position for P3 mobile robot

and

$$\dot{\theta} = w \tag{5.2}$$

where $[x_x, x_y]$ is the inertial position of the Pioneer 3 mobile robot, θ is the orientation of the robot and $[v, w]$ denote the linear and angular speeds of the robot. Moreover the hand position can be represented as Eq.(5.3)

$$\begin{aligned} h_x &= x_x + Lcos(\theta) \\ h_y &= x_y + Lsin(\theta) \end{aligned} \tag{5.3}$$

Now, differentiate Eq.(5.3) with respect to the time and substitute Eq.(5.1) into it, then

$$\begin{aligned} \dot{h_x} &= vcos(\theta) - Lsin(\theta)w \\ \dot{h_y} &= vsin(\theta) + Lcos(\theta)w \end{aligned} \tag{5.4}$$

Eq.(5.4) can be rewritten in matrix form as,

$$\begin{bmatrix} \dot{h_x} \\ \dot{h_y} \end{bmatrix} = \begin{bmatrix} cos\theta & -Lsin\theta \\ sin\theta & Lcos\theta \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}, \tag{5.5}$$

Therefore, based on Eq.(5.4) and Eq.(5.5) the hand position of robot can be controlled by manipulating linear and angular speed (v, w). Let

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ \frac{-1}{L} sin\theta & \frac{1}{L} cos\theta \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{5.6}$$

then

$$\begin{bmatrix} \dot{h_x} \\ \dot{h_y} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{5.7}$$

and

$$\dot{\mathbf{h}} = \mathbf{u}. \tag{5.8}$$

The *ith* agent in the double integrator system is described as

$$\begin{bmatrix} \dot{x_{i1}} \\ \dot{x_{i2}} \end{bmatrix} = \begin{bmatrix} x_{i2} \\ u_i \end{bmatrix}. \tag{5.9}$$

Comparing Eq.(5.9) with the Eq.(5.7) which is a single integrator system, it is necessary to convert double integrator system Eq.(5.9) to a single integrator system in order to model Pioneer 3 robots. Then

$$\dot{\mathbf{x}} = \mathbf{u}, \tag{5.10}$$

$$\begin{bmatrix} \dot{x_x} \\ \dot{x_y} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \tag{5.11}$$

The standard system equation is

$$\dot{x} = A\mathbf{x} + B\mathbf{u}, \tag{5.12}$$

and

$$\begin{bmatrix} \dot{x_x} \\ \dot{x_y} \end{bmatrix} = A \begin{bmatrix} x_x \\ x_y \end{bmatrix} + B \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{5.13}$$

where $A \in \mathbb{R}^{2\times2}$ and $B \in \mathbb{R}^{2\times2}$. In order to become Eq.(5.10) from Eq.(5.12), then let

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.14}$$

With the sampling time set as 0.1 second, the A and B in discrete time domain become

$$A_d = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; B_d = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{5.15}$$

This procedure has enabled to convert the double integrator system dynamics to the single integrator dynamics. Double integrator is preferred over single integrator dynamics as it can be applied to more complex systems. The advantage of this protocol is that different consensus dynamics including linear, periodic and positive exponential dynamics can be realized by choosing different gains. Sufficient conditions for solving the consensus problem with the considered general protocol are obtained, namely, all the gains realizing the consensus can be described.

5.3 Experimental Results

The primary objective of this experiment is to make the physical consensus of two Pioneer P3-DX robots which will individually act as a follower agent to a virtual leader agent. These agents are basically the laptops placed on each Pioneer robot. The leader agent is connected to these follower agents or laptops on the P3-DX robots via a common Dalhousie WPA-2 wireless network. Each agent is represented by its unique internet protocol address individually assigned when connected to this network. The virtual leader as a computer should be present only to assign the trajectory co-ordinates to the follower agents. So to ease the experimental setup,

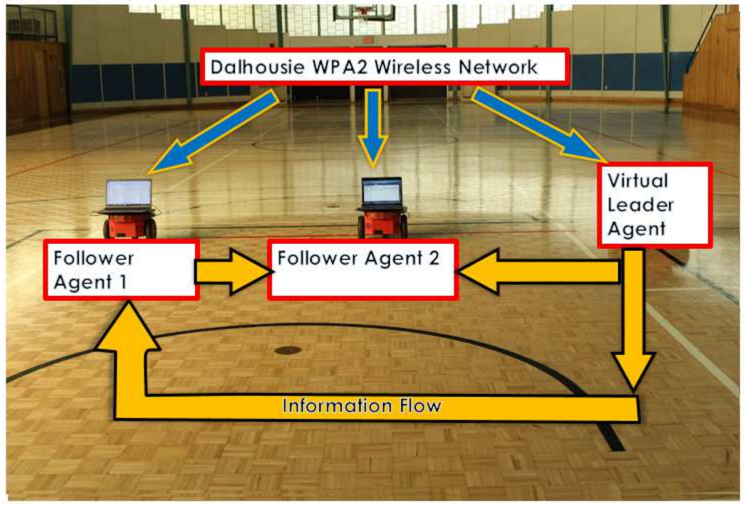

Figure 5.5: Network connectivity flow between the agents

only the follower agents along with the real robots should be present for consensus. The robots used in this experiment are Pioneer P3-DX robots owned by the Advanced Control and Mechatronics laboratory of the department of mechanical engineering of Dalhousie university. Fig. 5.5 shows the information flow between the connected agents to the network.

The main problem of using the multiple robots is that the co-ordinate system along which the robot traverse should be a global co-ordinate system. Global co-ordinate system means a system in which the position co-ordinates of each of the robots should be in a common co-ordinate system. The position co-ordinates of all robots or agents should be relative to each other and based on their values should be able to detect the exact position of the robot. However, this is not the case in real time working of the robots. All the Pioneer P3-DX robots have their individual local co-ordinate system. The default starting position co-ordinate of any robot is (0,0) which means the x-axis co-ordinate and y-axis co-ordinate of starting point is zero for each robot. So, in order to study the consensus behavior of the robot, all the robots involved in multi-robot cooperation should be present in a common virtual global co-ordinate system. The virtual global co-ordinate system is the local co-ordinate system of the virtual leader agent and the global co-ordinates of the follower agents are translated by calculating the relative position of the robots with respect to the virtual leader agent. The original dynamics is designed for a double integrator dynamics system, while the controller is to adjust the acceleration. Instead of controlling the acceleration, it is theoretically feasible to control wheel speeds for Pioneer 3 robots. The values of the linear velocity v and the angular velocity w can be obtained by the Eq.(5.5) and then converted to be the right wheel speed v_r and the left wheel speed v_l in Eq.(5.15)

$$\begin{aligned} v_r &= v + \frac{wl}{2} \\ v_l &= v - \frac{wl}{2} \end{aligned} \tag{5.16}$$

where, v_r and v_l are the right and left wheel speeds and l is the axial distance between two wheels. Therefore, the controller input is easily commanded by robot.setVel2(v_l, v_r).

The TCP/IP connection is the most advanced internet transmission protocol. So it is difficult to experience time delays of order that we are considering as well as packet data loss, unless there is packet traffic at some end. So it is necessary to artificially introduce the phenomena of packet loss as well as time delay. For packet loss, a preset function called rand()%100, which randomly generates integer 0 to 100 integrates. In this experimental case, the preset function is set as rand()%100 $\geq$ 90. It means that there is 10% data loss at all time but still 90% data is still successfully transmitted. In real time, all the packets are received by the hardware, but 10% of the information packets are dropped which means their values are neglected. The reason for considering 10% data loss rate is because the experimental result can be compared with the simulation result for 10% data loss case which showed low consensus time for data loss rate condition. For communication time-delay, there is no specific arrangement to track the time delay between the two packets received. But there is chance of implying transmission delay at the start of the robot equal to its sampling period which is 0.5 seconds in this experimental case. Because, after every 0.5 seconds, a data is stored for further analysis. So this experimental case is equivalent to a constant time-delay case with 10% packet data loss..

The local co-ordinate system of the virtual leader agent is the global co-ordinate system of the follower agents. The position local co-ordinates of the follower agents are translated as per the local co-ordinate system of virtual leader, thus making it work as a global co-ordinate system. As seen in the Fig. 5.6, the initial starting

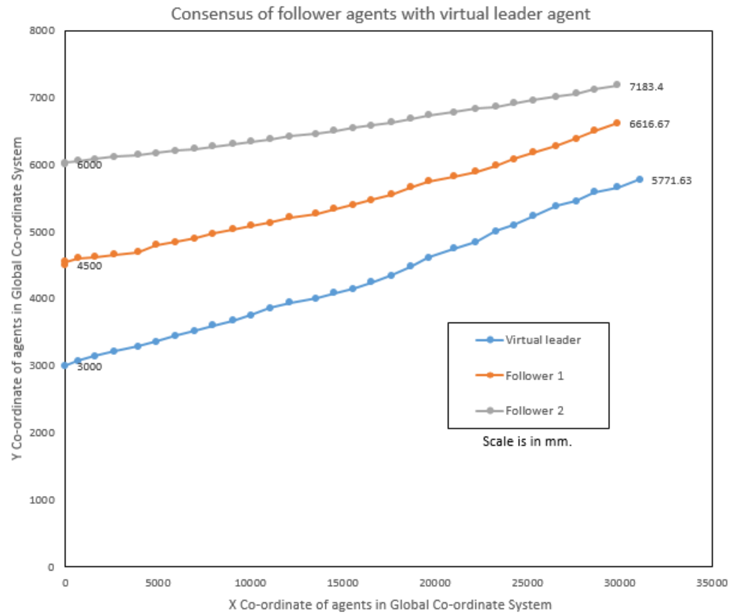

Figure 5.6: X-Y position co-ordinate of virtual leader and two follower system having consensus at 10% data loss rate and initial constant time delay $\tau = 0.5$ seconds in 31 seconds.

position of the virtual leader agent is (0,3000)mm. Pioneer P3-DX follower robot 1 is at a distance of 1500 mm from virtual leader robot along Y-axis at (0,4500)mm. Another Pioneer P3-DX follower robot 2 is at a distance of 3000 mm from virtual leader robot along Y-axis at (0,6000)mm. the X co-ordinate of all the robots at the initial starting point is zero. As seen in Fig. 5.5, the robots are kept along the same horizontal line. The trajectory of the robots is quite linear, hence this arrangement was preferred. Most of the change in trajectory will be observed along Y co-ordinate. The minimum safe stopping distance kept at every sampling time is 500 mm from each robots. If at any sampling time, if the minimum distance between two robots is going to be less than 500 mm, the robot is made to stop at that specific sampled time. In Fig 5.6, the difference between the Y co-ordinate is 566.73 mm. At the next sampling period, the distance between the two robots must be less than 500 mm, because of which it stopped at that moment. The running time of experiment took around 31 seconds.

The experimental case studied in this section can be compared with the Example

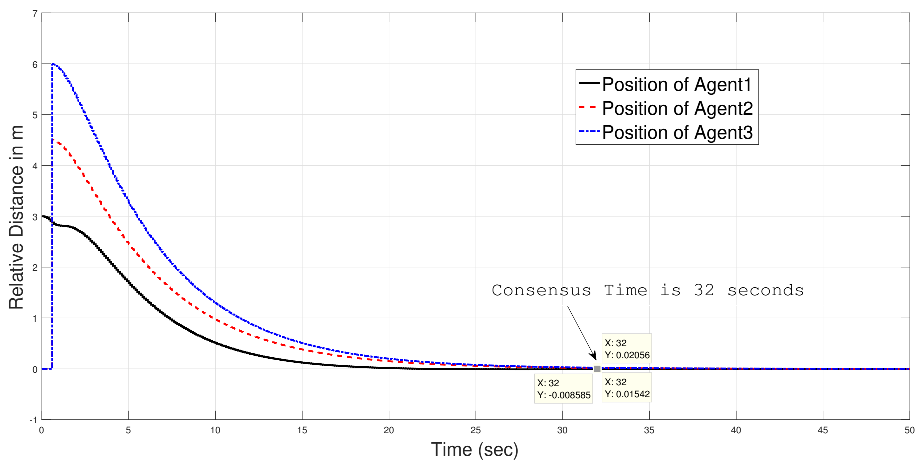

Figure 5.7: Consensus of position state of all agents at 10% data loss rate and $\tau =$ 0.5 seconds

4 of the Section 4.2.1. Obviously in this case, there is only initial presence of transmission delay of 0.5 second and the distance between each agents is quite less at 1500 mm as compared to 5000 mm in Example 4. Naturally, as the distance between the agents increases, the consensus time will increase.

Simulation results based on conditions similar to experimental are performed. The consensus time taken for the simulation is near to the actual experimental case. As seen in Fig. 5.8, it takes 32 seconds to achieve consensus, just one second more as compared to experimental result. The initial starting points in the simulations are same as that in experimental setup. The initial position of agents 1,2 and 3 are 3000mm, 4500 mm and 6000mm respectively. The maximum velocity by agent 3 is 0.73 m/s which is less than the actual maximum velocity of Pioneer robot as can be seen in Fig. 5.8. This also proves that the controller keeps a restraint on velocity as compared to actual conditions and does not overload the input. The initial constant time delay of 0.5 seconds is consider for simulation case as shown in Fig. 5.7, because the time delay in experimental case is kept as 0.5 seconds. The sampling time in this simulation is also kept to 0.5 second as in experimental case.

However, the simulation results are basically ideal results and don't take into consideration the non predicted factors, which are experienced in the experimental setups. The rate of convergence in simulation results is very high compared to experimental

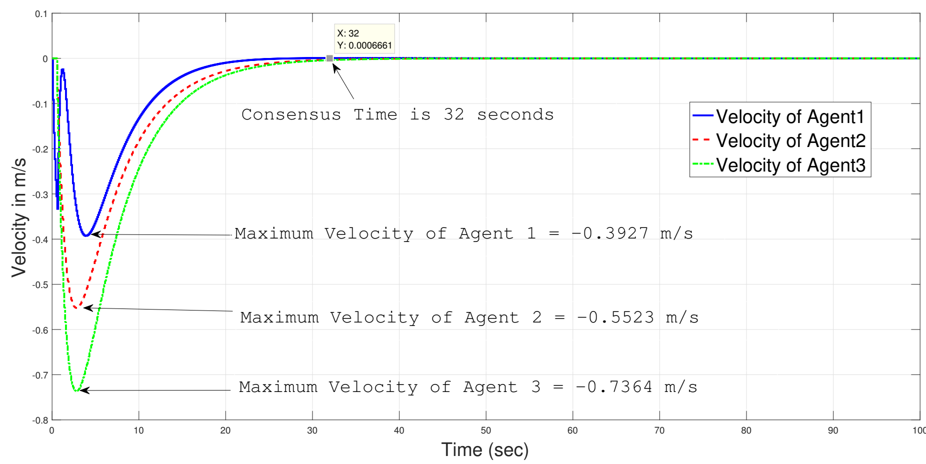

Figure 5.8: Consensus of velocity state of all agents at 10% data loss rate and $\tau =$ 0.5 seconds

results. Also in hardware experiment, the speeds of the robots are limited, whereas in simulation results, there is no such limit. However, in simulation results, there is always a trade off between the rate of convergence and processing time, which tries to limit the speed of agent at a sensible value. Moreover the position of the robots in hardware are calculated on odometry, any uncertainties and error could increase the inaccuracy of the experimental results. The robot needs time to receive, process and send data in the experiment rather than the simulation, which is not considered. However based on the experimental results with consideration of the limitation of the hardware, the hardware experiment certainly verified the simulation results in Chapter 4.

Chapter 6

Conclusions and Future Works

This chapter summarizes the the findings and the results of this thesis work and also at the same time suggests the developments that can be pursued in the future.

6.1 Conclusions

In this thesis, a novel consensus algorithm or protocol for the MAS in the event of communication link failure or time-delays over the network was developed and tested. For a certain limitations and assumptions, it is possible to develop a wireless network of agents that can work together in an objective which requires a leader to make decisions and other follower agents to work as a team. In may reserach works, there is either a incorporation of time-delays or packet loss in the system dynamics. There are different reasons for this consideration. Some researchers in the field of wireless networked control systems give importance to either of two. Because for few applications, packet losses are not a experienced phenomena and chances of packet losses to affect the system are quite bleak. In some applications packet losses are major cause of system failures. Depending on the requirements of applications of their fields, researchers chose to consider either of two communication constraints. Very few researchers have recently started to consider both the communication constraints together and to develop a relationship between them, so that when it is applied, the system is more dynamic to changes and problems faced during different processing times. The multi-agent systems that was considered for research work included the likes of nonholonomic mobile robots, quadcopters or drones etc. These system have chances of facing both the problems of time delays and packet loss. If there is packet loss in such kind of the system, chances are that of fatal errors. So in this research work, both the communication constraints were considered. The control techniques to tackle the problems of packet losses as well as the time delays are different for different system dynamics. A Lyapunov based methodologies were assumed to control these problems. The control gain achieved by this control technique have helped solve the LMIs which contain terms related to these communication constraints. The control gain developed by this methodology have helped achieve the leader following consensus of the multi-agent system with both the communication constraints smoothly within quite a impressive limited amount of time. The simulation results were studied in Chapter 4 based on the effect of data loss rate, constant time delay and time-varying delay. It was observed that beyond 30% data loss rate, the consensus was possible till 55% but it was not feasible value. So, the permissible value of data loss rate was tightened at 30%. The consensus time increases with increase in data loss rate. For constant time delays, the time delays are permissible till the sampling period of the system. Increasing the value of time delay beyond a certain value over sampling period achieves consensus but again the observed consensus time is not feasible. Over increasing the value of constant time delay by a large amount destroys the consensus nature of the system and look absurd which cannot be read or explained. With time-varying delays, the concern is with two factors, one is the upper bound of the time-varying delay as well as the range of the time-varying delay. Time-varying delay range extending the sampling period of the system makes it unfeasible and the nature of consensus is destroyed thus making the system fail randomly. Increasing the number of agents increases the consensus time of the system, as well the observations regarding constant time delay and time-varying delay related to increase in number of agents show the same effects and affect the consensus time by a little margin. An experimental setup was also conducted based on available hardware, in which a virtual leader was created and Pioneer P3-DX robots are used as two real followers. The same control gain is used for their dynamics and the observed results match with the simulation results thus validating the experimental results.

6.2 Future Works

There are some interesting work extensions that are very challenging and can be applied to same system dynamics.

- The topology considered in this this thesis is static and is fixed at all communication events and the entire process. One can definitely try to apply the same theoretical concepts and design a controller for switching topology or dynamic topology case.

- The assumption considered for the controller design was that the time delay should not exceed the sampling time and if it exceeds the sampling period, then it should be considered as a packet data loss. For systems having low sampling time but higher values of experienced time delays, this assumption might not work. For this, there needs to be special importance given to time delays and can be classified further as per controller requirements.

- Try to apply the same concept on higher integrator dynamics.

Bibliography

[1] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM Siggraph Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 25–34.

[2] P. Wang, "Interaction dynamics of multiple mobile robots with simple navigation strategies," *Journal of robotic systems*, vol. 6, no. 1, pp. 77–101, 1989.

[3] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

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

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

[6] W. Ren, R. W. Beard *et al.*, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on automatic control*, vol. 50, no. 5, pp. 655–661, 2005.

[7] J. Wu and Y. Shi, "Consensus in multi-agent systems with random delays governed by a markov chain," *Systems & Control Letters*, vol. 60, no. 10, pp. 863–870, 2011.

[8] S.-H. Wang and E. J. Davison, "On the stabilization of decentralized control systems," *Automatic Control, IEEE Transactions on*, vol. 18, no. 5, pp. 473–478, 1973.

[9] J.-P. Corfmat and A. Morse, "Decentralized control of linear multivariable systems," *Automatica*, vol. 12, no. 5, pp. 479–495, 1976.

[10] D. Cruz, J. McClintock, B. Perteet, O. A. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control-a multivehicle platform for research in networked embedded systems," *Control Systems, IEEE*, vol. 27, no. 3, pp. 58–78, 2007.

[11] D. Dimarogonas, M. Zavlanos, S. Loizou, and K. Kyriakopoulos, "Decentralized motion control of multiple holonomic agents under input constraints," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 4. IEEE, 2003, pp. 3390–3395.

[12] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[13] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical review letters*, vol. 75, no. 6, p. 1226, 1995.

[14] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *Automatic Control, IEEE Transactions on*, vol. 50, no. 1, pp. 121–127, 2005.

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

[16] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Transactions on Automatic Control*, vol. 6, no. 53, pp. 1503–1509, 2008.

[17] W. Yu, G. Chen, and M. Cao, "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems," *Automatica*, vol. 46, no. 6, pp. 1089–1095, 2010.

[18] C.-Q. Ma and J.-F. Zhang, "Necessary and sufficient conditions for consensusability of linear multi-agent systems," *Automatic Control, IEEE Transactions on*, vol. 55, no. 5, pp. 1263–1268, 2010.

[19] L. Scardovi and R. Sepulchre, "Synchronization in networks of identical linear systems," *Automatica*, vol. 45, no. 11, pp. 2557–2562, 2009.

[20] W. Ren, "Distributed leaderless consensus algorithms for networked euler–lagrange systems," *International Journal of Control*, vol. 82, no. 11, pp. 2137–2149, 2009.

[21] M. C. De Oliveira, J. C. Geromel, and J. Bernussou, "Extended h 2 and h norm characterizations and controller parametrizations for discrete-time systems," *International Journal of Control*, vol. 75, no. 9, pp. 666–679, 2002.

[22] 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.

[23] Y. Hong, L. Gao, D. Cheng, and J. Hu, "Lyapunov-based approach to multiagent systems with switching jointly connected interconnection," *Automatic Control, IEEE Transactions on*, vol. 52, no. 5, pp. 943–948, 2007.

[24] X. Wang and Y. Hong, "Finite-time consensus for multi-agent networks with second-order agent dynamics," in *Proceedings of the IFAC World Congress*, 2008, pp. 15 185–15 190.

[25] W. Ni, X. Wang, and C. Xiong, "Distributed control of networked linear systems via observer-based consensus algorithms," in *Control Conference (CCC), 2012 31st Chinese*. IEEE, 2012, pp. 1174–1178.

[26] Z. Feng and G. Hu, "Passivity-based consensus and passification for a class of stochastic multi-agent systems with switching topology," in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*. IEEE, 2012, pp. 1460–1465.

[27] W. Chen and X. Li, "Observer-based consensus of second-order multi-agent system with fixed and stochastically switching topology via sampled data," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 567–584, 2014.

[28] A.-M. Zou, K. D. Kumar, and Z.-G. Hou, "Distributed consensus control for multi-agent systems using terminal sliding mode and chebyshev neural networks," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 3, pp. 334–357, 2013.

[29] H. Liu, L. Cheng, M. Tan, Z. Hou, and Y. Wang, "Consensus tracking of general linear multi-agent systems: Fast sliding-mode algorithms," in *Control Conference (CCC), 2013 32nd Chinese*. IEEE, 2013, pp. 7302–7307.

[30] L. Dong, S. Chai, B. Zhang, and S. K. Nguang, "Sliding mode control for multi-agent systems under a time-varying topology," *International Journal of Systems Science*, no. ahead-of-print, pp. 1–8, 2014.

[31] Y. Jiang, J. Liu, and S. Wang, "Robust integral sliding-mode consensus tracking for multi-agent systems with time-varying delay," *Asian Journal of Control*, 2014.

[32] J. Lee, J.-S. Kim, H. Song, and H. Shim, "A constrained consensus problem using mpc," *International Journal of Control, Automation and Systems*, vol. 9, no. 5, pp. 952–957, 2011.

[33] Z. Cheng, H.-T. Zhang, M.-C. Fan, and G. Chen, "Distributed consensus of multi-agent systems with input constraints: a model predictive control approach," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 62, no. 3, pp. 825–834, 2015.

[34] G.-X. Wen, C. Chen, Y.-J. Liu, and Z. Liu, "Neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems," *Control Theory & Applications, IET*, vol. 9, no. 13, pp. 1927–1934, 2015.

[35] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *Automatic Control, IEEE Transactions on*, vol. 53, no. 8, pp. 1804–1816, 2008.

[36] Y. G. Sun, L. Wang, and G. Xie, "Average consensus in networks of dynamic agents with switching topologies and multiple time-varying delays," *Systems & Control Letters*, vol. 57, no. 2, pp. 175–183, 2008.

[37] Y. G. Sun and L. Wang, "Consensus problems in networks of agents with double-integrator dynamics and time-varying delays," *International Journal of Control*, vol. 82, no. 10, pp. 1937–1945, 2009.

[38] P. Lin and Y. Jia, "Average consensus in networks of multi-agents with both switching topology and coupling time-delay," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 1, pp. 303–313, 2008.

[39] Y.-P. Tian and C.-L. Liu, "Robust consensus of multi-agent systems with diverse input delays and asymmetric interconnection perturbations," *Automatica*, vol. 45, no. 5, pp. 1347–1353, 2009.

[40] F. Xiao and L. Wang, "State consensus for multi-agent systems with switching topologies and time-varying delays," *International Journal of Control*, vol. 79, no. 10, pp. 1277–1284, 2006.

[41] Y. Zhang and Y.-P. Tian, "Consensus of data-sampled multi-agent systems with random communication delay and packet loss," *Automatic Control, IEEE Transactions on*, vol. 55, no. 4, pp. 939–943, 2010.

[42] U. Munz, A. Papachristodoulou, and F. Allgower, "Consensus in multi-agent systems with coupling delays and switching topology," *Automatic Control, IEEE Transactions on*, vol. 56, no. 12, pp. 2976–2982, 2011.

[43] Y. Gao and L. Wang, "Sampled-data based consensus of continuous-time multi-agent systems with time-varying topology," *Automatic Control, IEEE Transactions on*, vol. 56, no. 5, pp. 1226–1231, 2011.

[44] Y. Hatano and M. Mesbahi, "Agreement over random networks," *Automatic Control, IEEE Transactions on*, vol. 50, no. 11, pp. 1867–1872, 2005.

[45] M. Porfiri and D. J. Stilwell, "Consensus seeking over random weighted directed graphs," *Automatic Control, IEEE Transactions on*, vol. 52, no. 9, pp. 1767–1773, 2007.

[46] D. Lee and M. W. Spong, "Agreement with non-uniform information delays," in *American Control Conference, 2006.* IEEE, 2006, pp. 6–pp.

[47] Y.-P. Tian and C.-L. Liu, "Consensus of multi-agent systems with diverse input and communication delays," *Automatic Control, IEEE Transactions on*, vol. 53, no. 9, pp. 2122–2128, 2008.

[48] G. Wen, Z. Duan, W. Yu, and G. Chen, "Consensus in multi-agent systems with communication constraints," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 2, pp. 170–182, 2012.

[49] J. C. Geromel, J. Bernussou, and P. Peres, "Decentralized control through parameter space optimization," *Automatica*, vol. 30, no. 10, pp. 1565–1578, 1994.

[50] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society*, vol. 14, no. 5, pp. 733–737, 1963.

[51] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[52] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[53] B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann, "The laplacian spectrum of graphs," *Graph theory, combinatorics, and applications*, vol. 2, pp. 871–898, 1991.

[54] R. Merris, "Laplacian matrices of graphs: a survey," *Linear algebra and its applications*, vol. 197, pp. 143–176, 1994.

[55] C. Godsil and G. F. Royle, *Algebraic graph theory.* Springer Science & Business Media, 2013, vol. 207.

[56] J. Lin, A. S. Morse, and B. D. Anderson, "The multi-agent rendezvous problem," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 2. IEEE, 2003, pp. 1508–1513.

[57] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 5, pp. 818–828, 1999.

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

[59] P. R. Chandler, M. Pachter, and S. Rasmussen, "Uav cooperative control," in *American Control Conference, 2001. Proceedings of the 2001*, vol. 1. IEEE, 2001, pp. 50–55.

[60] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[61] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A feedback architecture for formation control," in *Proceedings of the American Control Conference*, vol. 6, 2000, pp. 4087–4091.

[62] C. Belta and V. Kumar, "Trajectory design for formations of robots by kinetic energy shaping," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 2593–2598.

[63] M. B. Egerstedt and X. Hu, "Formation constrained multi-agent control," 2001.

[64] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3. IEEE, 2001, pp. 2968–2973.

[65] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1465–1476, 2004.

[66] L. E. Parker, "Designing control laws for cooperative agent teams," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on.* IEEE, 1993, pp. 582–587.

[67] E. Lavretsky, "F/a-18 autonomous formation flight control system design," *AIAA paper*, vol. 4757, 2002.

[68] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Decision and Control, 2007 46th IEEE Conference on.* IEEE, 2007, pp. 2289–2294.

[69] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on.* IEEE, 2005, pp. 6698–6703.

[70] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on.* IEEE, 2005, pp. 8179–8184.

[71] W. Li and Y. Jia, "Consensus-based distributed multiple model ukf for jump markov nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 227–233, 2012.

[72] P. Wang and F. Y. Hadaegh, "Coordination and control of multiple microspacecraft moving in formation," *Journal of the Astronautical Sciences*, vol. 44, no. 3, pp. 315–355, 1996.

[73] P. Wang, "Navigation strategies for multiple autonomous mobile robots moving in formation," *Journal of Robotic Systems*, vol. 8, no. 2, pp. 177–195, 1991.

[74] F. Hadaegh, W.-M. Lu, and P. Wang, "Adaptive control of formation flying spacecraft for interferometry," 1998.

[75] M. S. De Queiroz, V. Kapila, and Q. Yan, "Adaptive nonlinear control of multiple spacecraft formation flying," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 3, pp. 385–390, 2000.

[76] T. Sugar and V. Kumar, "Decentralized control of cooperating mobile manipulators," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4. IEEE, 1998, pp. 2916–2921.

[77] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 905–908, 2001.

[78] M. Mesbahi and F. Y. Hadaegh, "Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 369–377, 2001.

[79] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 3, pp. 443–455, 2004.

[80] W. Ren and R. W. Beard, "Formation feedback control for multiple spacecraft via virtual structures," in *Control Theory and Applications, IEE Proceedings*, vol. 151, no. 3. IET, 2004, pp. 357–368.

[81] L. Sheng, "Lyapunov-based control approaches for networked single and multi-agent systems with communication constraints," 2010.

[82] Z. Meng, W. Ren, Y. Cao, and Z. You, "Leaderless and leader-following consensus with communication and input delays under a directed network topology," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 41, no. 1, pp. 75–88, 2011.

[83] W. Ren *et al.*, "Information consensus in multivehicle cooperative control," 2007.

[84] G. De La Torre, T. Yucelen, and E. Johnson, "Constrained formation protocols for networked multiagent systems," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. IEEE, 2013, pp. 824–830.

[85] R. Ghadami and B. Shafai, "Distributed h 2 control of multi-agent dynamic systems: Continuous-time case," in *Proceedings of the 2010 American Control Conference*. IEEE, 2010, pp. 3969–3974.

[86] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-uav system," in *2014 American Control Conference*. IEEE, 2014, pp. 2077–2082.

Appendices

Appendix A

Operations Manual

A.1 MATLAB and Simulink Files

- Run the LMIRevisedConstantDelay.m file to run the constant delay LMIs. The value of state matrices depend on the system considered.
- Run the LMIMULTIAGENTNEWBASED.m file to run time-varying delay LMIs.
- Running these m files will yield the value of control gain. This value of control gain needs to be substituted in ConsensusModel.slx file.
- Next step is to run MASConsensus.m file which will run this simulink model file(.slx) to get the graphs of different states.

A.2 Microsoft Visual Studio C++ 2010 files

- There are two network options to connect the Pioneer robots. One is the Dalhousie WPA/2 wireless network which is present all over Dalhousie campus and another is the ACM LAb network for connecting all laptops in ACM lab. Only condition is that all the robots should be connected to same wireless network. One needs to keep track of IP address and enter the same in programming (.cpp) file.
- Each laptop needs to be connected to their Pioneer robot with the help of RS-232 Serial Cable. The USB end is connected to the laptop.
- Connect the DELL Inspiron to one of the Pioneer P3-DX robot and open the file. Open service project file stating name 'Robot1outof2'. There will be all files like (.cpp) file, (.exe) file having same name. Just double click on (.exe) file. This will connect the Robot 1 to wireless network.

- Connect the Toshiba Satellite to one of the Pioneer P3-DX robot and open the file. Open service project file stating name 'Robot2outof2'. There will be all files like (.cpp) file, (.exe) file having same name. Just double click on (.exe) file. This will connect the Robot 2 to wireless network and the experiment will start.

Appendix B

Author's Publication List

- X. Gong, Y.J. Pan and A. Pawar, "A Novel Leader Following Consensus Approach for Multi-Agent Systems with Data Loss", International Journal of Control, Automation and Systems, Accepted, April 2016.

- A. Pawar and Y.J. Pan, "Leader-following Consensus Control of Multi-Agent Systems with Communication Delays and Random Packet Loss", In Proceedings of the IEEE American Control Conference, June 2016, Boston, USA, pp.4464-4469.