CONSENSUS ANALYSIS ON NETWORKED MULTI-AGENT SYSTEMS WITH STOCHASTIC COMMUNICATION LINK FAILURE

by

Xiang Gong

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

at

Dalhousie University
Halifax, Nova Scotia
February 2013

DALHOUSIE UNIVERSITY

DEPARTMENT OF MECHANICAL ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "CONSENSUS ANALYSIS ON NETWORKED MULTI-AGENT SYSTEMS WITH STOCHASTIC COMMUNICATION LINK FAILURE" by Xiang Gong in partial fulfillment of the requirements for the degree of Master of Applied Science.

Dated: February 15, 2013

Supervisor: _____

Readers: _____

_____

DALHOUSIE UNIVERSITY

DATE: February 15, 2013

AUTHOR: Xiang Gong

TITLE: CONSENSUS ANALYSIS ON NETWORKED MULTI-AGENT SYSTEMS WITH STOCHASTIC COMMUNICATION LINK FAILURE

DEPARTMENT OR SCHOOL: Department of Mechanical Engineering

DEGREE: M.A.Sc. CONVOCATION: May YEAR: 2013

Signature of Author

To my parents, girlfriend, and friends
without your encouragement and support
this would not have been possible.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis is to develop a novel consensus algorithm or protocol for multi-agent systems in the event of communication link failure over the network. The structure or topology of the system is modeled by an algebraic graph theory, and defined as a discrete time-invariant system with a second-order dynamics. The communication link failure is governed by a Bernoulli process. Lyapunov-based methodologies and Linear Matrix Inequality (LMI) techniques are then applied to find an appropriate controller gain by satisfying the sufficient conditions of the error dynamics. Therefore, the controller with the calculated gain is guaranteed to drive the system to reach a consensus. Finally, simulation and experiment studies are carried out by using two Mobile Robot Pioneer 3-DX sand one Pioneer 3-AT as a team to verify the proposed work.

Acknowledgements

I am heartily thankful to my supervisor, Dr.Ya-Jun Pan, whose encouragement, guidance and support from the initial to the final level of this thesis, and it would not be possible without her guidance. I would also like to thank Dr. Mae Seto and Dr. Jason Gu for their role on my supervisory committee, Peter Jones and Jon MacDonald for their technical assistance, and the Advanced Controls and Mechatronics research group for their suggestions and comments.I also would like to thank NSERC and CFI Canada for the funding support. Last, but certainly not least, I would like to give my deepest gratitude to my parents, Thank them for their unconditional support.

Chapter 1

Introduction

The following sections outline the background for the research work, including a research motivation, multi-agent system overview with applications, and research contributions.

1.1 Research Motivation

Multi-agent systems (MAS) are composed of several intelligent agents which are capable to interact each other through communication channels. Since the advantage of the larger amount of agents, the multi-agent system are widely used at a relatively large scale of projects which are difficult or impossible for a single agent system to achieve. In order to control a multi-agent system for a task, the most important challenges is to ensure each agent in the system is able to cooperative each others by sharing information, and eventually reach consensus. The consensus is like a agreement among the system, each agent needs to follow this agreement in order to successfully complete the job or task. For example, in a formation control, the goal is to make all agents move at certain shape with the same speed. It is just like a parade. Therefore, in this case, the consensus happens when all agents asymptotically move with the same velocity and form a flock without collisions. In order to reach this consensus, each agent needs to share its elemental information to others in terms of velocities, positions or accelerations. This information is also called information state, which can be single or multiple, depending on what kind of task is required. Mathematically, the consensus can be reached by designing cooperative control protocols for each agent based on the local information exchanged from others. Thus all agents are converged to a common value in state space.

The applications for MAS include space-based combat, surveillance, terrain mapping, and underwater vehicle's mine hunting, etc. To enable these applications, various cooperative control capabilities need to be developed, including formation control, rendezvous, flocking, foraging, and air traffic control, etc. For MAS consensus, one of the most important challenges is the uncertainty caused by inter-vehicle communication, i.e., the information exchange among vehicles is unreliable. Today, agents in

multi-agent systems are mostly and popularly controlled through wireless networked systems due to its incomparable advantages in cost, operation range, flexility, and so on. However, wireless networks are not always as reliable as hardwired ones due to connection strength, bandwidth constraints, which can cause packet delays and data losses. Therefore, the networked control system stability has been studied extensively associated with communication channel constraints. These new interesting and challenging problems regarding MAS with wireless networked control are the motivation of the works presented in this thesis.

1.2 Multi-agent System Overview

In order to understand a multi-agent system, first, we need to know what an agent is. In [1], an agent is a computer system capable of autonomous action in some environment to achieve its delegated goals. It is shown from Fig. 1.1. An agent can act like a human, it has sensors to obtain information from the surrounding environment, and actuators to perform its action, Therefore an qualified agent is able to percept, decide and act based on the feedback from the environment and reprogrammed goals.

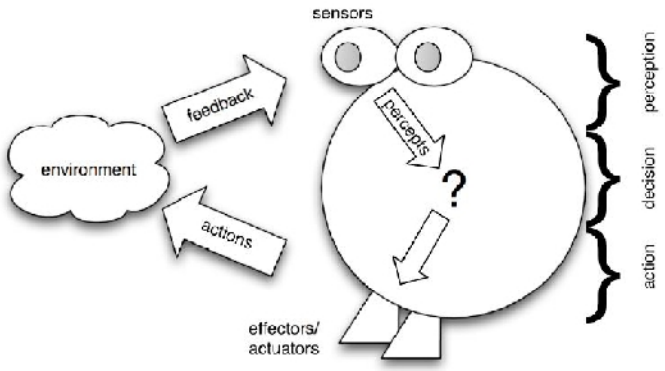


Figure 1.1: A Agent [1]

In [1] multi-agent system is defined as that one system consists a number of agents, which interact with one-another. In most cases, all agents are programmed by the same team and collaborated to complete the same task. To ensure a successful interact, each agent requires the ability to cooperate, coordinate, and negotiate with others, shown in Fig. 1.2. The field of cooperative control of multi-agent systems can trace back to the late $1980's$, when several researchers started to investigate issue in multiple robot systems. Before that, researches have been mainly concentrated on a single robot [3]. Thanks to the advent of communication technologies, especially the

introduction of inexpensive and reliable wireless communication systems, researches on distributed control of multi-agent systems have been dramatically grown in the $1990's$ [4]. By the early $2000's$, cooperative control of multiple agents have been mainly developed by using of unmanned aerial vehicles (UAVs) in United stats [5], spurring further researches on this field. In Fig. 1.3 shows well-known UAV global hawk formation flying, which is one of multi-agent system applications. Over the last decade, this research area has been blossomed, since a single complex system can be efficiently replaced by an interacting multi-agent system with simpler structures. In fact, a group of small non-holonomic robots such as, wheeled robots or UAVs with simple structures can achieve more complex tasks at a lower cost than a single complex robot owing to their modularity and flexibility [6].

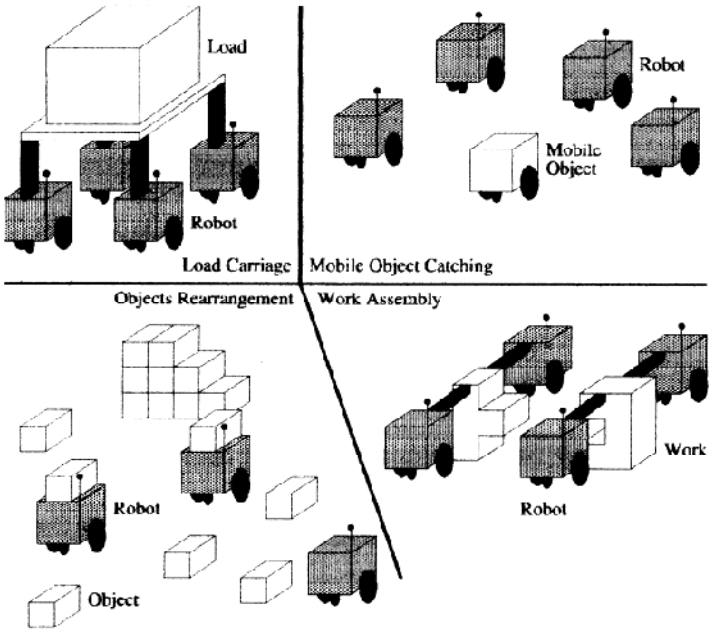


Figure 1.2: A Multi-agent System [2]

In [8], the cooperative control for multi-agent systems is categorized into two problems, one is formation control with applications to mobile robots, UAVs, satellites, aircraft, and automated highway systems; the other is called non-formation control, such as task assignment (constructing a building), playload transportation, and role assignment. Cooperative control of multi-agent systems poses significant theoretical and practical challenges. In [9], author has listed four challenges. First, the complex of architecture of MAS, involving several subsystems rather than a single system. Second, the limitation of communication bandwidth and connectivity among the system may cause the information exchanging unreliable. It is also difficult to show how

Figure 1.3: Global Hawk Formation Flying [7]

the communication channel to be established in order to decide what to communicate and when and with whom the communication takes place. Third, consensus between the team goals and individual goals need to be achieved. It is challenging to reach both goals without any compensation. Fourth, since MAS includes multiple agents, and computational resources of each individual agent will be always limited compared with a single agent. In this thesis, the research objective is to concentrated on first two challenges. For example, the first challenges is solved by adopting a directed algebraical graph, and the second challenge is managed by using Bernoulli process governed data losses.

1.3 Consensus Control Review

Multi-agent System consensus control can be noticeably divided into two phases; one is an early period, researchers focused on the strategies to ensure each member of system can be reached consensus without communication channel constraints, and followed by a recent phase, researcher focused consensus strategies with communication channel constraints, such as packets dropout and time delays. In this section, it summarizes consensus controlling techniques in two categories with the most elemental and recent research articles in this field.

1.3.1 Multi-agent Systems Consensus without Communication Constraints

Consensus problems have recently been studied with different controlling techniques. In [10] and [11], a algebraic graph based method is used to ensure the stability of controlling multi-vehicle formation. The algebra graph theory is widely applied for controlling of multi-agent systems. A graph consists of a set of nodes or vertices V together with a set of edges E. The V is symbolized identified agents, and E is symbolized the communication channels. If any agent can be reach directly or indirectly from others, then the system can be converged. Now the problem is how to prove the system can reach consensus. In [12] an Nyquist criterion that used the eigenvalues of the graph Laplacian matrix to determine the effect of the communication topology on formation stability. In [13] and [14], the multi-agent system is inspired by a topological structure of the network. However it was characterized by a digraph eventually. By means of transforming the Laplacian of the digraph into its Frobenius canonical form the system, which can decompose into one or several minimal-independent subsystems, and ensure each subsystem achieve consensus in result of entire system reach consensus.

In a higher order dynamics, such as [15]. It modeled the multi-agent system in a directed graph, where the agents are with dynamical order two. The control protocol depends on two parameters, position-cooperative and velocity-cooperation parameter, as well as the Laplacian associated with communication network. Finally using matrix analysis approached the necessary and sufficient conditions for inertial state and position and/or velocity inertia-consensus, were given for the second-order systems. In [16], it also introduced a second-order consensus protocol and derived necessary and sufficient conditions under uni-directional interaction topologies. In [17], this paper also modeled agents by identical linear nth order dynamical systems and the interconnection topology between the agents is modeled as a directed weighted graph. In [18], this paper addressed the control of a team of nonholonomic mobile robots navigating in a terrain with obstacles while maintaining a desired formation and changing formations when required, using graph theory.

In [19], a proposed approach shares many of the limitations of the Extended Kalman Flitter (EKF), such as a lack of guaranteed stability. Nevertheless, it can be expected that the wide successful usage of the EKF implies that the current approach will suffice for many problems of practical interest. The modified relative dynamics can

be viewed as linear system with uncertain parameters and an optimal guaranteed cost control law can be applied. In [20], distributed flocking algorithms for a multi-agent system are designed and analyzed on theoretical framework with considering of multiple obstacles. An Linear Matrix Inequality (LMI)-based designed to ensure reach convergence with necessary and sufficient condition. In [21], two model independent solutions to the problem of controlling wheel-based mobile platforms were proposed. These two algorithms were based on a so called virtual vehicle approach, where the motion of the reference point on the desired trajectory was governed by a differential equation containing error feedback. In [22], author considered undirected formations for centralized formations and directed formations for decentralized formations.

1.3.2 Summary

Based on literatures in this domain, the algebraic graph theory is a very mature approach the structure of the multi-agent system. each node is represented as an agent, the communication transmission is modeled as edges in the graph. This method is mostly used for a non-predefined trajectory, such as [11]. The other way to model the system is focused on single agent with predefined trajectory. If all agents follow their own trajectory, then the system reaches consensus, such as [21]. It used a called virtual vehicle approach, which is a geometric path following approach. In this thesis, a non-virtual leader is proposed, the system is flexible and followers reach consensus based on the information of the leader, which is more practical with wider applications. Therefore, an algebraic graph theory is used in this thesis.

Once the structure is established, the next step is to decide the system dynamics, most articles are focused on a single integrator dynamics due to the simplicity. Normally the higher order can represent more complicated system, such as [15]. In this paper, the controlling algorithm depends on both position and velocity cooperation. In this thesis, a double integrator dynamics is adopted. The most important step is to verify the stability of the system. The fundamental condition of consensus is to share information among system, and ensure each agent can reach the sharing information. Several techniques are used to verify the stability of the system, like in [12], an Nyquist criterion that used the eigenvalues of the graph Laplacian matrix to determine the effect of the communication topology on formation stability. Moreover in [13] and [14], the stability is proved by transforming the Laplacian of the digraph into its Frobenius canonical form the system.

1.3.3 Multi-agent Systems Consensus with Communication Constraints

Recently, the studies on multi-agent system have been considered of communication channel constraints among the system. Since the MAS control is still considered as a relatively new technology, a number of technical challenges need to be overcome before they are to be used effectively. In [23], one of the most important challenges is the uncertainty caused by inter-vehicle communication,i.e., the information exchange among vehicles may be unreliable. Therefore, the networked control system stability has been studied extensively associated with communication channel constraints. Most of relevant studies still primarily focus on a algebraic graph theory to present a structure of the system. However communication links are not stable or reliable anymore. In this phase, researchers need to find a method to represent the constraint of communication channels, and then prove the system is stable in the event of communication constraints. In [24], it proves that the agents over arbitrary switching topologies converge to a common steady state if the undirected or directed interaction topology is jointly connected. In [25], the multi-agent system had a discrete-time single-integrator dynamics, and the communication topology was modeled by a undirected and time-varying graph. This time-varying graph changes depended on a decentralized Model Predictive Control (MPC).

In [26] and [27], it pointed the integral connectivity was a key concept for solving the consensus problem under time-variant network, and dynamics of agents was described by a single and double-integrator respectively in these two papers. In [28], author studies the affection of both packet dropout and transmission delays induced by communication channels in the system. Both continuous-time and discrete-time cases were studied, and the proposed LMIs technique was based on Lyapunov-Razumikhin and Lyapunov-Krasovskii function method respectively. In [29], an H∞ controller synthesis was studied with time delay system approach. The goal was to utilize an H∞ norm to provide a pe-specified disturbance attenuation level, and alternatively to analyze robust stability of dynamical system. Again, a LMI technique was to examine the condition of stability in order to ensure the consensus of the multi-agent system.

In [30], the interaction topologies were studied with dynamically changing, which can represent some certain communication links were broken due to bandwidth constraints. Both discrete and continuous update schemes were proposed for information consensus. This note shows that information consensus under dynamically

changing interaction topologies could be achieved asymptotically if the union of the directed interaction graphs have a spanning tree frequently enough as the system evolves. In [31], each member of group updated its current state based on the current information received from neighboring agents. The neighboring agents could be varied from time to time depended on the strength of communication channel and designed topologies. Once more, the stability analysis is based upon a blend of graph-theoretic and system-theoretic tools. In [32], the agent's information was only updated from the closet agent. The system model was based on the Vicsek model, which was proposed by Vicsek at 1995. It was a simple but compelling discrete-time model of n autonomous agents. Finally, author approved that a graphic example of a switched linear system which was stable using Vicsek model.

In [33], authors used Euler–Lagrange systems for distributed coordinated tracking problems. A group of followers were modeled by full-actuated Euler-Lagrange equations, and track a dynamic leader whose vector of generalized coordinates was time varying under the constraints. However in [34], it was also used Euler–Lagrange systems, a adaptive controller that could achieve global full-state synchronization. For example, the difference between the agents positions and velocities asymptotically converges to zero. In [35], author considered a team of continuous-time of second order agents communicating via switching topology, which was modeled by Bernoulli random process. A necessary and sufficient condition of the solvability of the mean-square robust consensus problem was established to ensure the consensus of the multi-agent system.

1.3.4 Summary

Once the system is subjected to a communication constraints, the focus is to prove the system's stability in the even of communication constrains, such as data losses, packet delays. Therefore a static or original algebra graph theory cannot be applied directly. Most researcher implemented a time-variant switching graph, and then prove the system to reach consensus by using different methods, such as Model Predictive Control, H∞, and Lyapunov based methods, followed by a LMI technique which is to examine the condition of stability in order to ensure the consensus of the multi-agent system. The communication constrain can be represented by the Vicsek model, Euler–Lagrange systems or Bernoulli random process. Therefore, after reviewed all relevant articles, a switching algebra graph governed by Bernoulli process is used to model the structure of the system, since it has been widely used. Lyapunov based

methods and LMI techniques also are used to examine the condition of stability.

1.4 Applications Overview

In this section, it summarizes some of the main applications for the control of multi-agent systems.

1.4.1 Formation Control

With the development of advanced technologies, modern military systems have been becoming increasingly sophisticated with unmanned vehicles. Especially, at early $2000's$, unmanned aerial vehicles (UAVs) were significantly attracted military's interest because of low cost, easy maneuver, high stealthy and the most important is zero casualty. One of the simplest cooperative control problems is flight formation control over multiple UAVs by cooperating with other UAVs. Basically, in a flight formation control, each member needs to know at least the relative locations and speeds of nearby aircraft. The earliest research on this area was proposed by Parker in [36]. The control laws for cooperative agent teams utilized a combination of local and/or global knowledge to achieve a formation. It presented some general guidelines and principles for determining the appropriate level of global versus local control. According to [37], back to 2002, NASA also implemented a very courageous experiment in flight formation control by using F/A-18 fighters . It proved that unmanned aerial vehicles can reduce human error and increase accuracy during a formation flight, shown in Fig. 1.4.

Figure 1.4: F/A-18 Formation Flight [38]

1.4.2 Rendezvous and Cooperative Attack

Another application of multi-agent system for military is to rendezvous the troops and attack cooperatively. Fig. 1.5 shows a battle space management scenario and illustrates that each member of troop, such as tanks, aircrafts, control bases can receive and send commands to each other to form a multi-agent system. It highly increases the efficiency of operation.

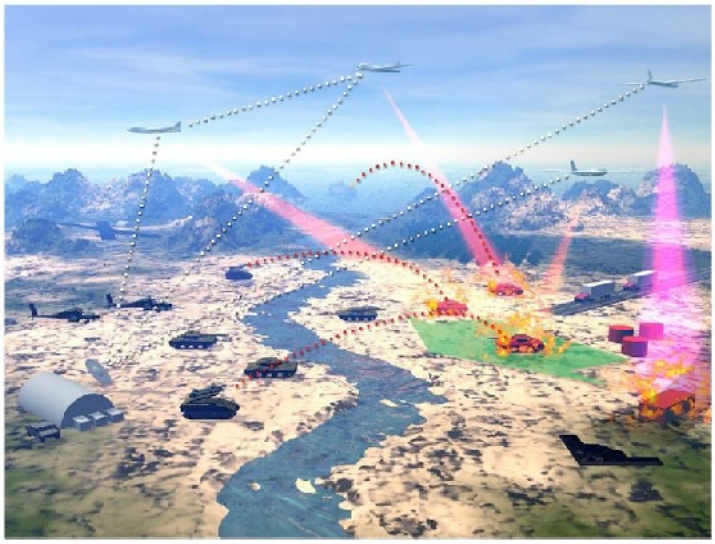


Figure 1.5: Multi-agent System Application in Modern Military Operation [39]

1.4.3 Data Collecting and Sampling

Since multi-agent systems have more advantages in massive tasks because of larger number of agents. In many areas, MAS are used to collect and sample data, such as biology, geography, and oceanology. In [40], it shows an autonomous ocean sampling network created by monterey bay aquarium research institute is an example of an environmental sampling network. This system consists of a collection of robotic vehicles that used for sampling. Another example is to patrol and map in an unknown environment by sending a group of vehicles and mapping rapidly based on information or data collected by each member. Fig. 1.6 shows a number of robots employ in the patrolling task, and each closed loop represents a trajectory of each robot.

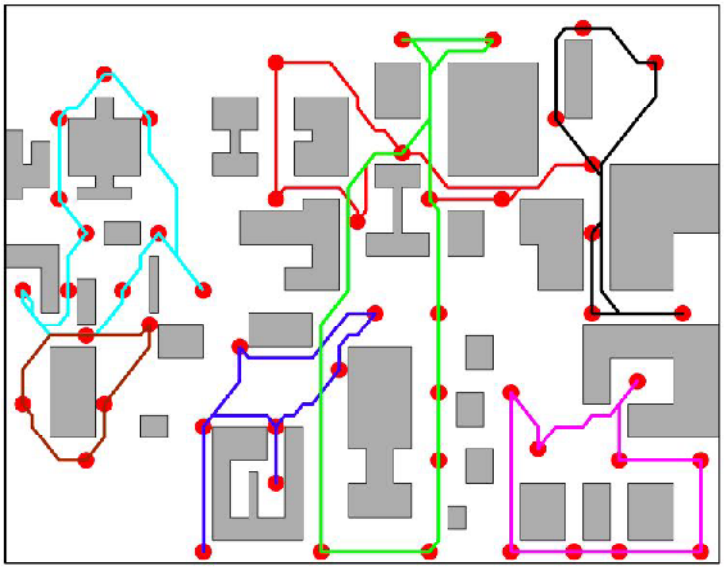


Figure 1.6: Multi-agent System Application in Mapping [41]

1.4.4 Intelligent Transportation System

The application of multi-agent system in transportation systems are also received considerable attentions over the last few decades. One of examples is an intelligent transportation system. The goal is to use modern communication and information technology to increase the efficiency of a transportation management system in order to optimize vehicle life, fuel efficiency, safety, and traffic. In this system, each vehicle can be treated as an agent, and they can communicate and share information between each other, shown in Fig. 1.7.

Another example is called an automatic highway system. A group of automatically driven cars are in a close spacing in the highway, which is a method of increasing the capacity of roads without adding any traffic lane. In 1997, California Partners for Advanced Transit and Highways (PATH) demonstrated an automatic highway system by allowing cars to be driven automatically down a highway at a very close spacing, shown in Fig. 1.8.

As density of air traffic continues to increase, air traffic control systems are another area where multi-agent system cooperative control can be applied ([44] and [45]). It is like the automatic highway system, the air traffic control can reduce human error during a heavy load of air traffic, and increase the operational efficiency.

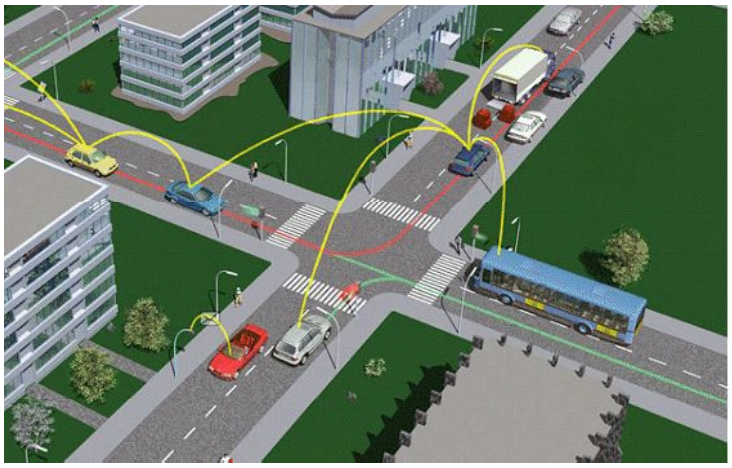


Figure 1.7: Multi-agent System Application in Intelligent Transportation System [42]

Figure 1.8: Multi-agent System Application in Automatic Highway System [43]

1.5 Thesis Contribution

The main contribution of this thesis is to develop a novel consensus controlling algorithm for a cooperation control problem of a multi-agent system with stochastic communication link failures among the system.

i) Specifically, the topology of MAS is modeled by a direct algebraic graph, and each transmission of communication link is assumed to be unreliable. Therefore a Bernoulli Process is implemented to represent the packet dropout during data transmission.

ii) A system dynamics is defined as a discrete time-invariant with second-order

system. The consensus problem is first converted into a corresponding error dynamics in order to form a leader following consensus.

iii) Lyapunov-based methodologies and Linear matrix inequality (LMI) techniques are applied to find the sufficient conditions of the error dynamics for the stabilization controller design.

iv) Moreover, it systematically summarizes the effect of data loss rate, communication weight, initial value, and sampling time on the efficiency of the controller.

v) Another important contribution is to experiment the consensus controller into Pioneer 3 wheeled robots by modifying kinematic model of the nonholonomic differentially driven mobile robots.

vi) Finally, these simulation results and experiments approved the feasibility and effectiveness of the proposed controller.

1.6 Thesis Outline

The thesis outline is structured as follows. In Chapter 2, a background of algebraic graph theory associated with multi-agent system is introduced. In Chapter 3 and 4, the design of the consensus control strategy for distributed cooperation control problems of multi-agent system with group random communication failure is demonstrated. In Chapter 5, the simulation works are proceeded to test the effectiveness and feasibility of the proposed algorithm by discussing effects by data loss, communication weight, initial value, and sampling time. In Chapter 6, simulation results on Pioneer 3 robots are shown by modifying a double-integrator into a single-integrator system. Moreover, simulation results show the actual trajectories of robots. A comparison study is also conducted in this chapter to prove the effectiveness of the designed controller. In Chapter 7, the experimental results regarding the main works represented in Chapter 3 are shown. Chapter 8 presents the conclusions and future works.

Chapter 2

Multi-agent System with Graph Theory

2.1 Algebraic Graphic Theory

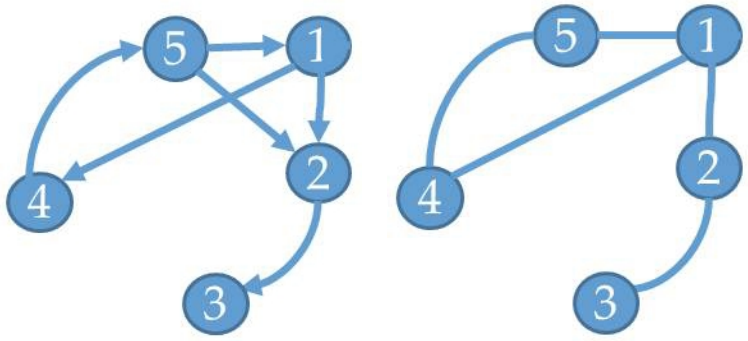
In order to completely understand the consensus control of multi-agent systems, algebraic graphic theory is one of the most fundamental concepts and approaches. In multi-agent systems, the information exchanging is normally modeled by directed or undirected graphs.

2.1.1 Directed and Undirected Graph

In [46] and [47], a directed graph also called digraph consists of a set of nodes or vertices V together with a set of edges E. The digraph is represented as $G(V, E)$, the V is symbolized identified agents, and E is symbolized the communication channels. For example, if (i, j) is an edge in E, then Agent i can receive information from Agent j. In contrast, for an undirected graph, the edge (i, j) represents Agent i and j can exchanged information in both directions. In other words, the undirected graph also can be considered as a combination of two directed graph edge with opposite directions. For example, in Fig. 2.2(a) , there are five vertices, and each of them represented as an agent, then $V = (1, 2, 3, 4, 5)$, and $E = (5, 4)(1, 5)(2, 5)(4, 1)(2, 1)(3, 2)$. For example, $(5, 4)$ means the formation can be only transferred from Agent 4 to 5. At (b), it is an undirected graph, then $V = (1, 2, 3, 4, 5)$, with $E = (5, 4)(5, 1)(4, 1)(2, 1)(2, 3)$. For example, $(5, 4)$ means the formation can be transferred from Agent 4 to 5 and vice versa. Therefore, an undirected graph is more reliable than a directed graph due to its unique double direction of transmission. In this thesis, the focus is on a system with an unreliable network; thus the directed graph is chosen to model the multi-agent system.

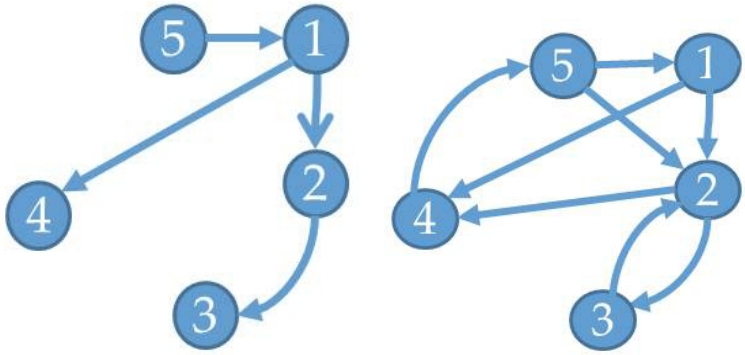
2.1.2 Adjacency Matrix

Firstly, some terminologies in the graphic theory need to be defined. In [48], spanning tree is a connected graph that contains no cycles. In a tree, every pair of points is connected by a unique path. That is, there is only one way to get from node i to



(a) (b)

Figure 2.1: (a) Directed Graph (b) Undirected Graph with 5 Agents



(a) (b)

Figure 2.2: (a) Directed Spanning Tree (b) Strongly Connected Directed Graph

j. Fig. 2.2(a) illustrates a directed tree. A directed graph is strongly connected if any vertex in G is reachable for from each other vertex, which is shown on Fig. 2.2(b) .

Degree in the graph theory is the number of vertices adjacent to a given vertex is called the degree of the vertex and is denoted $d(v)$. It can be obtained from the adjacency matrix of a graph by simply computing each row sum. For example, the degree of vertex 5 in Fig. 2.2(b) is 4. However in the case of the directed graph, the degree is defined as indegree and outdegree. The indegree is the number edges pointing to a node V (i.e. incoming edge), denoted $id(v)$. For example, the $id(4)$ in Fig. 2.2(b) is 2, because two incoming edges point toward to node 4. The outdegree is the opposite definition of indegree, the $od(4)$ in Fig. 2.2(b) is 1. In this thesis, all degrees are unified as the indegree. Every graph is associated with an adjacency

matrix to represent the relationship of connection among agents. It is a binary $n \times n$ matrix A, where n is the number of agents. The adjacency matrix $A_a = [a_{ij} \cdot \omega_{ij}]$ is defined as $a_{ii} = 0$, and $a_{ij} = 1$ if $(j, i) \in E$ where $i \neq j$. For example, at Fig. 2.1(a) , the adjacency matrix A_a is defined as

$$A_a = \begin{bmatrix} 0 & 0 & 0 & 0 & 1\omega_{15} \\ 1\omega_{21} & 0 & 0 & 0 & 1\omega_{25} \\ 0 & 1\omega_{32} & 0 & 0 & 0 \\ 1\omega_{41} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1\omega_{54} & 0 \end{bmatrix} \tag{2.1}$$

where w_{ij} called communication weight or weight can be independently added in each channel to define a corresponding information flow rate. It is represented as ω_{ij}, and which means there is a weight ω_{ij} added on the edge of Agent j to i. In [47], ω_{ij} is a percentage gain to control the information flow rate in the channel, 1 means 100% of information capacity, and 0 means the channel is disconnected. Therefore, ω_{ij} is selected from 0 to 1.

2.1.3 Laplancian Matrix

A Laplancian matrix L of a directed graph is an $n \times n$ matrix as well, $L = [l_{ij}] \in R^{n \times n}$ as $l_{ij} = \sum_{j \neq i} a_{ij} \cdot \omega_{ij}$ and $l_{ij} = -a_{ij} \cdot \omega_{ij}$ for all $i \neq j$. Note that of (j, i) is not connected, then $l_{ij} = -a_{ij} \cdot \omega_{ij} = 0$. According to the definition as shown from the above, Fig. 2.1(a) has Laplacian matrix L as following:

$$L = \begin{bmatrix} 1\omega_{15} & 0 & 0 & 0 & -1\omega_{15} \\ -1\omega_{21} & 1\omega_{15} + 1\omega_{25} & 0 & 0 & -1\omega_{25} \\ 0 & -1\omega_{32} & 1\omega_{32} & 0 & 0 \\ -1\omega_{41} & 0 & 0 & 1\omega_{41} & 0 \\ 0 & 0 & 0 & -1\omega_{54} & 1\omega_{54} \end{bmatrix} \tag{2.2}$$

If the ω_{ij} is one for all links, then Laplacian matrix L becomes

$$L = w_{ij} \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{2.3}$$

As known from [48], for an undirected graph, L is symmetric. However, for a directed graph, in both types of graphes, since row sums are 0, 0 is an eigenvalue of L with associated eigenvector $1 \triangleq [1, ...1]^T$, which is the $n \times 1$ column vector of ones, and l is diagonally domain and has non-negative diagonal entries. According to Gershorin"s disc theorem [49], for an undirected graph, all of the nonzero eigenvalues of L are positive, while for the undirected graph, the L has the positive real parts for the nonzero eigenvalues. Then the real eigenvalues of L satisfies $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \lambda_N$ [48]. In [12], for a directed graph, 0 is a simple eigenvalue of L if the directed graph is strongly connected. From [48], because the column vector 1_n of ones is an eigenvector associated with the zero eigenvalue, which implies that span$\{1_n\}$ is contained in the kernel of L. It follows that if zero is a simple eigenvalue of L, then $x(t) \longrightarrow \bar{x}1_n$, where $x(t)$ is information state, and $\bar{x}$ is a scalar constant, which implies that $|x_i(t) - x_j(t)| \longrightarrow 0$, as $t \longrightarrow \infty$, for all $i, j = 1, ...n.$, where $x_i(t)$ and $x_j(t)$ represent the Agent i and j's information states. If $|x_i(t) - x_j(t)| \longrightarrow 0$, then both information state in Agent i and j will be convergent. Therefore the consensus is reached, then ensuring zero is a simple eigenvalue of L is the condition of the consensus in term of the system's topology. The next section lists a few examples to prove this condition.

2.2 Consensus Condition in MAS Networks

Consensus control in multi-agent systems can be divided into centralized and decentralized approaches. In this thesis, all of consensus algorithms are based on the decentralized or distributed control method, since it only needs to deal with the local neighbor-to-neighbor interaction, and evolve in a parallel manner. Because of these facts of distributed control, it reduces the communication requirements, and improves the system's reliability. Compared to the centralized control, if one agent failed, the entire system will be dysfunctional. The consensus problem is to design an update law, which allows the information states of all agents in the network to converge into a common value. The information state can be selected as local positions of each vehicle, or velocity and direction of the motion. This problem is a canonical problem in distributed coordination, and it can be solved by using algebraic graph theory. In particularly, we assume that each agent is located at and that it can only measure the relative position of its neighbor, i.e it can only measure $x_i - x_j, \forall_j \in N_i$.

2.2.1 Single Integrator as An Example

If each agent has single integrator dynamics, i.e. $\dot{x}_i = u_i$,then the continuous consensus algorithm is given in [48].

$$\dot{\mathbf{x}}_i = -\sum_{j \in N_i} (\mathbf{x}_i - \mathbf{x}_j) \tag{2.4}$$

where N_i is the set of agents or robots adjacent to Robot i, and it can also be represented as adjacency matrix. since the network is assumed to be static dynamics, the adjacency matrix A_a is unchangeable. Then Eq.(2.4) can be rewritten as

$$\dot{\mathbf{x}}_i = -\sum_{j=1}^{n} a_{ij}(\mathbf{x}_i - \mathbf{x}_j). \tag{2.5}$$

Expanse the Eq.(2.5)

$$\dot{\mathbf{x}}_i = -deg(v_i)\mathbf{x}_i + \sum_{i=1}^{N} a_{ij}\mathbf{x}_j \tag{2.6}$$

where the $deg(v_i)$ is the indegree of node i in the directed graph, and a_{ij} is the adjacency. Because

$$L = D - A_a$$

where D is the indegree matrix. and A_a is the adjacency matrix. Eq.(2.6) can be written in the Laplacian matrix

$$\dot{\mathbf{x}} = -L\mathbf{x} \tag{2.7}$$

where $\mathbf{x} = [\mathbf{x}_i, ..., \mathbf{x}_N]^T$. By [48], it is known that the system's stability properties are entirely dependent on the eigenvalues of L matrix. But from the previous section, we know that as long as graph G is connected, and the L is positive semi-definite, then the $-L$ is the negative semi-definite with a single zero for one of the eigenvalues, and all others are real and negative. As such, the system is stable, and $\mathbf{x}$ will tend to be the null-space of L asymptotically, $\mathbf{x}_i \to \alpha$, as $t \to \infty$, where α is the final state value. Eq.(2.4)-(2.7) are assumed that there is no communication delay during the operation. When information is exchanged among vehicles through communication, time delays from both message transmission and processing after reception must be considered. Let δ_{ij} be denoted as the information time delay obtained from vehicle i to j. Then Eq.(2.5) can be modified as:

$$\dot{\mathbf{x}}_i = -\sum_{j=1}^{n} a_{ij}[\mathbf{x}_i(t-\delta_{ij}) - \mathbf{x}_j(t-\delta_{ij})] \tag{2.8}$$

If the time delay only affects the information state which is being obtained so that Eq.(2.8) is written as

$$\dot{\mathbf{x}}_i = -\sum_{j=1}^{n} a_{ij}[\mathbf{x}_i(t) - \mathbf{x}_j(t-\delta_{ij})]. \tag{2.9}$$

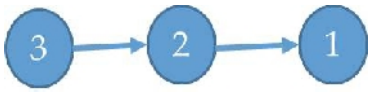


Figure 2.3: A Group of Three Agents

In this thesis, the time delay is not considered. Therefore δ_{ij} is set to be zero in all examples.

2.2.2 Simulation Results with Three Agents Consensus

Take Fig. 2.3 as an example, there are three agents represented as a connected spanning tree. The Laplacian matrix L is defined as

$$L = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.10}$$

In order to analyze the consensus stability, eigenvalues need to be calculated. They are 0, 1, and 1. Therefore there is a simple zero, and the rest are positive, which ensures the consensus can be reached. The following Fig. 2.4 also proves it.

Eq.(2.10) is the Laplacian matrix of Fig. 2.3, and the weight w is assumed to be 1. Since each distance of the link between any two agents is the difference, the weight can ensure that a specific final value can be reached, such as average consensus. This will be discussed in the next example. Fig. 2.4 shows the simulation of three agents or robots in the network of Fig. 2.3 conducting a rendezvous operation. It shows that Robot 3 is the only information sender to the rest of the two robots. Therefore the Robot 3 is not receiving any information and remains to be stationary, if the information state is considered as the position, while Robot 1 and 2 are converging to meet with Robot 3.

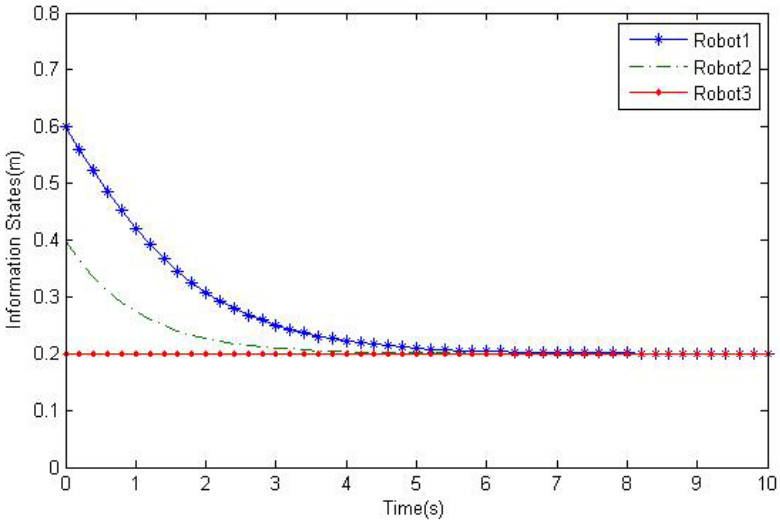


Figure 2.4: Simulation Result of a Group of Three Agents in A Connected Spanning Tree

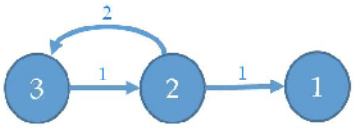


Figure 2.5: Weight in a Group of Three Agents

$$L = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -2 & 2 \end{bmatrix} \tag{2.11}$$

Fig. 2.5 is a weighted graph, and the Eq.(2.11) is the Laplacian matrix, whose eigenvalues are 0, 1, and 3. Fig. 2.6 is to show the simulation result at topology of Fig. 2.5. Since the communication link between Robot 2 and 3 are double than the rest of links, Robot 2 and 3 should be converged double faster than the previous example. From Fig. 2.4, the convergent time between Robot 2 and 3 is 4.5 seconds, while with a weight 2 added in the link, the time is reduced to 2.2 seconds, obtained from Fig. 2.6.

The weight, ω_{ij} is not a absolute but a percentage and relative number, which is to control the information flow rate. The highest number in the weight means 100% of information flow rate, then other numbers in the weight is proportionally decreased. Therefore, $\omega_{ij} \in (0, 1)$. Each communication channel is like a pipe, and communication weight is the valve installed in each pipeline. If the weight equals one, it means

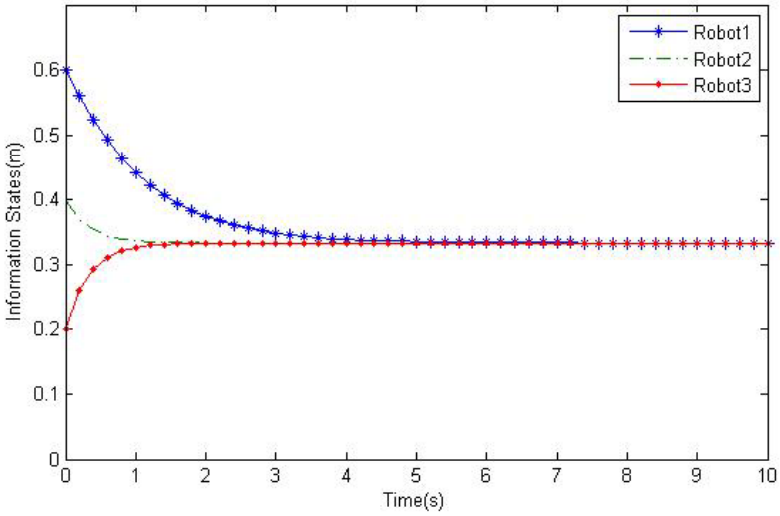


Figure 2.6: Simulation Result of a Group of Three Agents in Communication Weight

the valve is fully open, otherwise it is proportionally closed with the weight decreases, until it is fully closed with weight becomes zero. Therefore, Eq.(2.11) actually can be simplified as

$$L = \begin{bmatrix} 0.5 & -0.5 & 0 \\ 0 & 0.5 & -0.5 \\ 0 & -1 & 1 \end{bmatrix} \tag{2.12}$$

The simulation result based on Eq.(2.12) is exactly the same with Fig. 2.6, and shown below in Fig. 2.7.

2.2.3 Simulation Results with Four Agents Consensus

Now consider a unreachable convergence example. A group of four agents are shown in Fig. 2.8.

$$L = \begin{bmatrix} 1.5 & -1 & -0.5 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.13}$$

The eigenvalues of Eq.(2.13) are 0, 0; 1.5, and 2. Since it has two zeroes, it is shown that the consensus is unable to be reached. Then Fig. 2.9 can prove it.

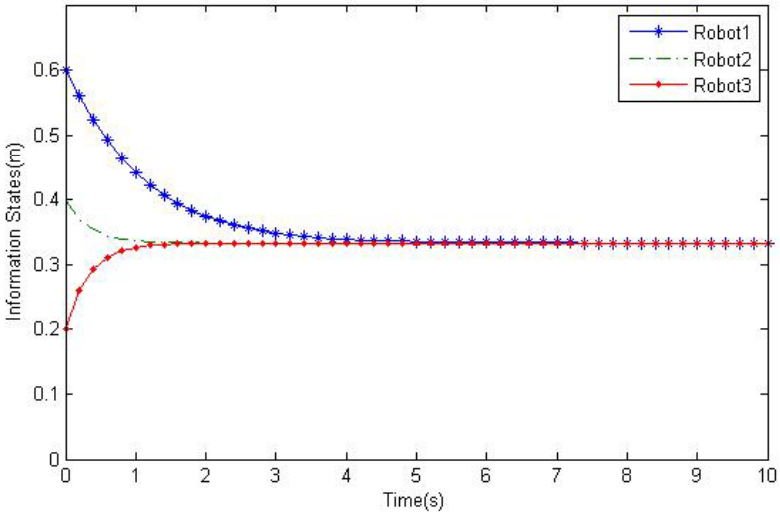


Figure 2.7: Simulation Result of a Group of Three Agents with Simplified Weights

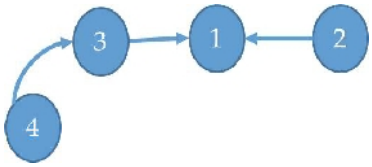


Figure 2.8: a Group of Four Agents

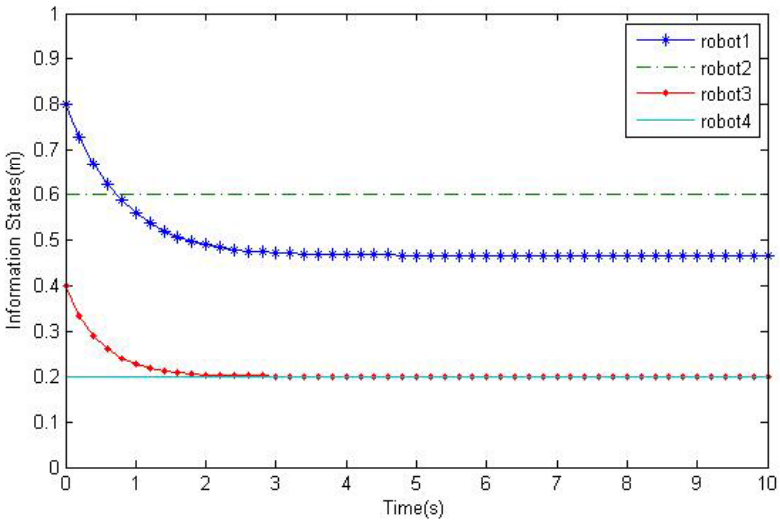


Figure 2.9: The Simulation Result of a Group of Four Agents without Consensus Archived

2.3 Summary

In this chapter, a basic background on algebraic graph is introduced, and shown that it can be used to model the topology of multi-agent systems. In order to reach consensus for all agents in the system, an important condition is to make sure the graph is connected. Therefore, the L matrix is positive semi-definite, then the $-L$ is the negative semi-definite with a single zero for one of the eigenvalues, and all others are real and negative. As such, the system is stable, and can be reached to consensus. Moreover, Fig. 2.6 and Fig. 2.7 proves that the weight, $\omega_{ij} \in (0, 1)$ is not a absolute but a percentage and relative number, which is to control the information flow rate. The highest number in the weight means 100% of information flow rate, then other numbers in the weight is proportionally decreased.

Chapter 3

Problem Formulation

In this chapter, a problem formation of multi-agents system is described by a combination of an algebra graph and a Bernoulli process. The algebra graph is to represent the topology of the system, and Bernoulli process is to represent the random data loss situation among the system, which is described as discrete-time with a double integrator dynamics.

3.1 System Descriptions

A single control system dynamics is expressed in a continuous-time system as follows

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

where $\mathbf{x_i}(t) \in R$ is the state vector, $u_i(t) \in R$ is the control input. t is the time. For example, the initial state is described as $\mathbf{x}(0)$, and $i \in 1, 2, ...n$ represents the identity of the agent. n is the total number of agents in the system. A_c and B_c are two known constant matrices with appropriate dimensions. For a simple double integrator system,

$$\begin{aligned} \dot{x}_{1,i}(t) = &\quad x_{2,i}(t) \\ \dot{x}_{2,i}(t) = &\quad u_i(t) \end{aligned} \tag{3.2}$$

A_c and B_c are,

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.3}$$

A state-feedback controller

$$u_i = -K_i\mathbf{x_i},\ i = 1, 2, ..., n \tag{3.4}$$

is designed to stabilize the system (3.1). However, in many practical control systems, such as computer-based control systems, the continuous time system is controlled by a digital controller, which accepts continuous-time signals as inputs and gives out continuous-time signals as outputs. Therefore, the continuous system has to be

converted into the discrete system by 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 the 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. Fig. 3.1 shows the block diagram of the digital control system.

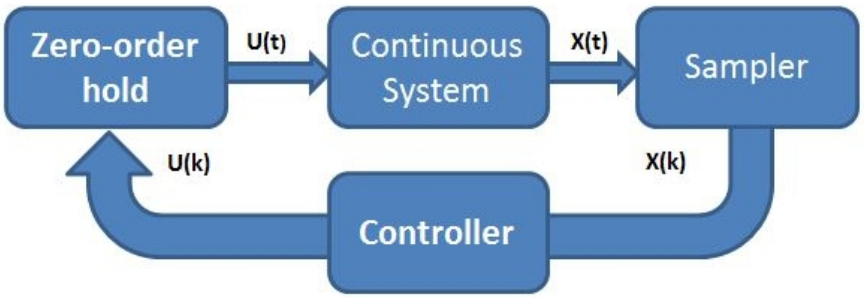


Figure 3.1: The Block Diagram of Sampled-Data System

Thus, Eq.(3.2) is converted to a discrete time system by a zero-order holder.

$$\mathbf{x}_i(k+1) = A_d\mathbf{x}_i(k) + B_d u_i(k) \tag{3.5}$$

$$u_i = -K\mathbf{x_i} \tag{3.6}$$

where $\mathbf{x}_i \in \boldsymbol{R}$, and $u_i \in \boldsymbol{R}$ denote the state and control input of Agent i respectively. The A_c and B_c are constant appropriate size of matrices. The A_d and B_d are varied according to the sample time period (Ts). For example, if the Ts is set as 0.1 second, then the A_d and B_d in discrete time system are

$$A = A_d = \begin{bmatrix} 0 & 0.1 \\ 0 & 1 \end{bmatrix}, B = B_d = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix}. \tag{3.7}$$

Thus, in multi-agent systems, the close-loop control system is expressed in discrete time system. Consider a group of n agents, the linear dynamic can be illustrated by Fig. 3.2. If the open-loop system is asymptotically stable, i.e. $\rho(A) < 1$, the $\rho(\cdot)$ is the spectral radius of a matrix. Based to Eq.(3.5), the protocol asymptotically solves the consensus problem, if and only if for any initial state, the agents agree upon a common state, i.e. $lim_{k\to\infty}||x_i(k) - x_j(k)|| = 0$ for any $i \neq j$. Moreover, if $lim_{k\to\infty}E(||x_i(k) - x_j(k)||)^2 = 0$ for any $i \neq j$, the system (3.5) will reach mean square consensus asymptotically. A discrete-time consensus controller is proposed as

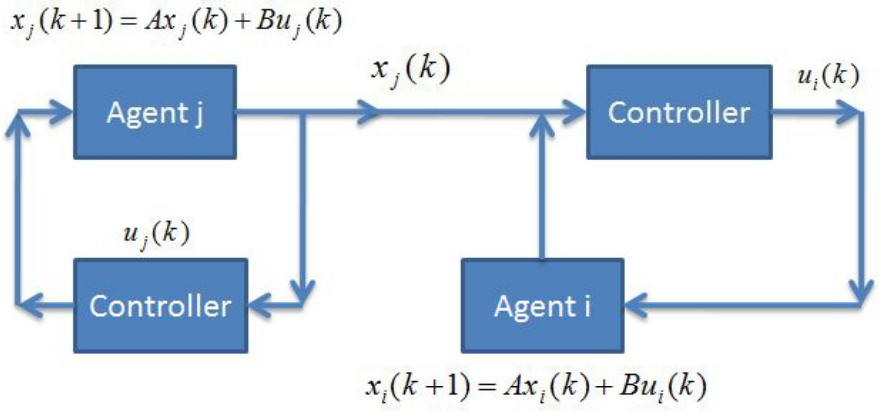


Figure 3.2: The Block Diagram of Sampled-Data System

$$u_i(k) = K\sum_{j=1}^{n} a_{ij}(k)w_{ij}(\mathbf{x}_j(k) - \mathbf{x}_i(k)), \tag{3.8}$$

where w_{ij} is the communication weight added on communication link (i, j). For simplification, set $w_{ij} = 1$, then the close-loop system is

$$\mathbf{x}_i(k+1) = A\mathbf{x}_i(k) + BK\sum_{j=1}^{n} a_{ij}(k)(\mathbf{x}_j(k) - \mathbf{x}_i(k)), \tag{3.9}$$

where k is the communication event. $a_{ij}(k) \in \{0, 1\}$ is the (i, j) entry of an adjacency matrix $A_a = [a_{ij}] \in R^{n\times n}$, (referred to Eq.(2.1)), at the discrete-time index k. The matrix A_a is dependant on the topology of the system.

3.2 Communication Channels and Data Losses

Fig. 3.2 illustrates how any two agents transfer information from one to another. This can be any communication link among the multi-agent systems, which is modeled by an algebra graph. When the system is not subjected to communication constrains, the information which is x can be freely and guaranteed to transmitted at anytime. Now, the challenge is to model the system in a switching graph. It means each link can be broken in any time step. If the system is discrete time system, and the data loss is shown in Fig. 3.3. Ts is the sampling time. information transferred from Agent i to j is subjected to an unreliable connection.

In [50], a Markov chain is adopted to model the packet-loss process in the communication channel. The Markov chain presents transition from one state to another between a finite number of possible states. The process is random and memoryless.

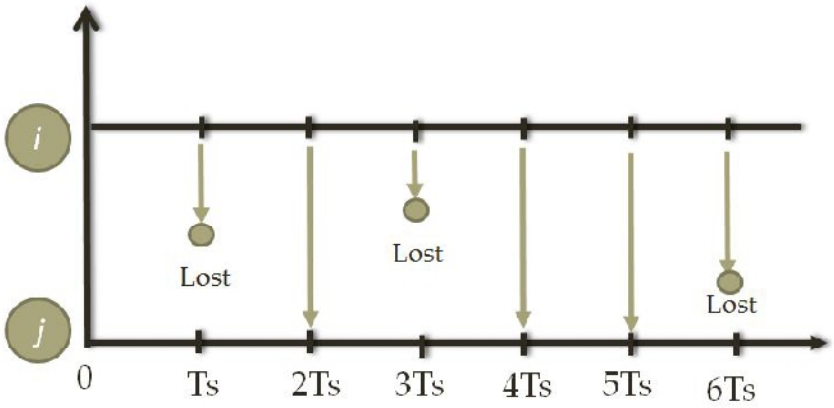


Figure 3.3: Data Flow Diagram of the Packet Loss

The next state depends only on the current state and not on the sequence of events that preceded it. Take Fig. 3.4 as example, there is three states, and the probability of transmission from state to state is specified. Therefore the probability of each state is corresponding to transmission probabilities.If State 1 receives 25% probability from State 0 and 2, respectively, then the probability of State 1 is 50%.

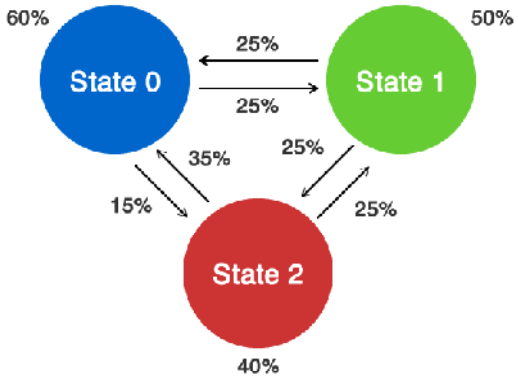


Figure 3.4: Data Flow Diagram of Markov Chain [51]

In order to ensure a successful data transmission, a packet delay also needs to be considered. Fig. 3.5 shows a data transmission subjected to both packet losses and delays. Now, the challenge is to find an optimal way to model communication channels with both packet losses and delays. A Bernoulli process is considered to model the packet losses in every time step. If there is not data receiving in current time step, it will trace backward to previous data. The process is random and memoryless, and guaranteed the information received is the latest successful data. If the transmission

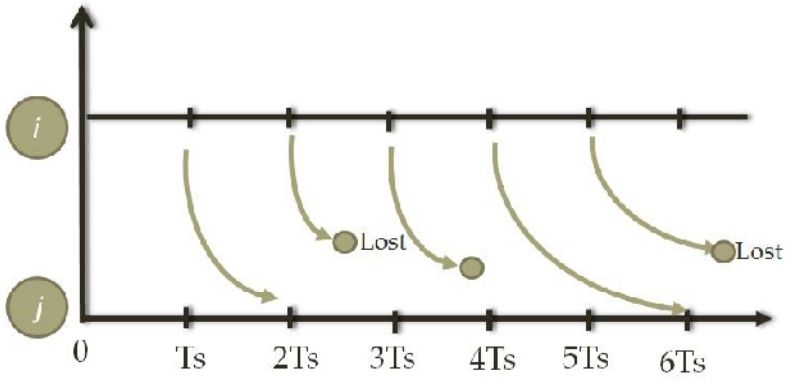


Figure 3.5: Data Flow Diagram of The Packet Loss and Delays

delay is over the certain sampling period, the process will drop the data and consider as a packet loss instead of a packet delay. Therefore, Bernoulli process combines data losses and delays into a single data loss case.

3.3 Modeling with Bernoulli Process

The Bernoulli Process is described as

$$\mathbf{x}_i(k) = \theta_{ij}(k)\mathbf{x}_i(k) + (1 - \theta_{ij}(k))\mathbf{x}_i(k-1) \tag{3.10}$$

where $\theta_{ij}(k) \in R$ is a stochastic variable satisfying interval Bernoulli distribution, and $\theta \in \{0, 1\}$. The i and j mean the signal is transferred from Agent j to Agent i. If Agent i cannot successfully receive the signal from Agent j at time k, the $\theta_{ij}(k) = 0$, otherwise $\theta_{ij}(k) = 1$. Moreover Eq.(3.10) is a recursive procedure. For example, if at $k = 1$, Agent i can receive the signal, while $k = 2, 3$ cannot receive data serially, then

$$\mathbf{x}_i(3) = \theta_{ij}(3)\mathbf{x}_i(3) + (1 - \theta_{ij}(3))\mathbf{x}_i(2) = \mathbf{x}_i(2) \tag{3.11}$$

where

$$\mathbf{x}_i(2) = \theta_{ij}(2)\mathbf{x}_i(2) + (1 - \theta_{ij}(2))\mathbf{x}_i(1) = \mathbf{x}_i(1) \tag{3.12}$$

then

$$\mathbf{x}_i(3) = \mathbf{x}_i(1).$$

Therefore, Agent i receives the most recently successfully transferred data. Substitute the Eq.(3.10) into $\mathbf{x}_j$ and $\mathbf{x}_i$ of Eq.(3.9), shown on Fig. 3.6. Then Eq.(3.9) becomes

$$\begin{aligned}\mathbf{x}_i(k+1) = \quad & A\mathbf{x}_i(k) + BK\sum_{j=1}^{n} a_{ij}(k)[\theta_{ji}(k)\mathbf{x}_j(k) + (1-\theta_{ji}(k))\mathbf{x}_j(k-1) \\ & -\theta_{ij}(k)\mathbf{x}_i(k) - (1-\theta_{ij}(k))\mathbf{x}_i(k-1)],\end{aligned} \tag{3.13}$$

where $\theta_{ji}(k)$ and $\theta_{ij}(k)$ are depended on success of the transmission from Agent i to j and from Agent j to i, respectively. If it is successful, then $\theta_{ji}(k)$ and $\theta_{ij}(k) = 1$. The communication reliability of each channel is theoretically independent in the system, which means $\theta_{ji}(k)$ or $\theta_{ij}(k)$ could be different in the same time k. In this thesis, all $\theta(k)$ is assumed to be uniform and synchronous. All $\theta_{ij}(k) = \theta_{ji}(k) = \theta(k)$ at the same time. Therefore, Eq.(3.13) is simplified as

$$\begin{aligned}\mathbf{x}_i(k+1) = \quad & A\mathbf{x}_i(k) + BK\sum_{j=1}^{n} a_{ij}(k)[\theta(k)\mathbf{x}_j(k) + (1-\theta(k))\mathbf{x}_j(k-1) \\ & -\theta(k)\mathbf{x}_i(k) - (1-\theta(k))\mathbf{x}_i(k-1)].\end{aligned} \tag{3.14}$$

Based on Fig. 3.2, Agent j sends signals to Agent i, and it is governed by Bernoulli process, which is shown in Fig. 3.2.

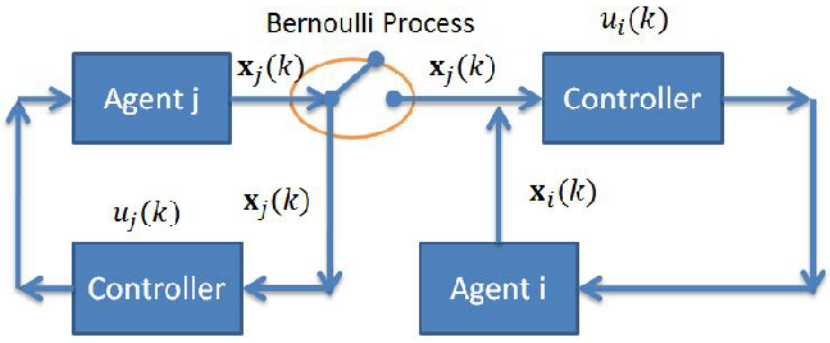


Figure 3.6: The Block Diagram of Bernoulli Process in Multi-agent Systems

So far, Eq.(3.13) is based on a two agent system, in order to find the general equation to define a n agent system. Let assume there is a number of n agents. Therefore finding the information state of each agent at time $k+1$ is represented as

$$
\begin{aligned}
\mathbf{x}_1(k+1) &= A\mathbf{x}_1(k) + BK\sum_{j=1}^{n} a_{1j}(k)[\theta(k)\mathbf{x}_j(k) + (1-\theta(k))\mathbf{x}_j(k-1) \\
&\quad -\theta_j(k)\mathbf{x}_1(k) - (1-\theta(k))\mathbf{x}_1(k-1)] \\
\mathbf{x}_2(k+1) &= A\mathbf{x}_2(k) + BK\sum_{j=1}^{n} a_{2j}(k)[\theta(k)\mathbf{x}_j(k) + (1-\theta(k))\mathbf{x}_j(k-1) \\
&\quad -\theta(k)\mathbf{x}_2(k) - (1-\theta(k))\mathbf{x}_2(k-1)] \\
\vdots \qquad &\quad \vdots \\
\mathbf{x}_n(k+1) &= A\mathbf{x}_n(k) + BK\sum_{j=1}^{n} a_{nj}(k)[\theta(k)\mathbf{x}_j(k) + (1-\theta(k))\mathbf{x}_j(k-1) \\
&\quad -\theta(k)\mathbf{x}_n(k) - (1-\theta(k))\mathbf{x}_n(k-1)].
\end{aligned}
\tag{3.15}
$$

From simplicity, $a_{ij} = a_{ij}(k)$, and $\theta = \theta(k)$. Thus, the final general form of multi-agent system is the following equation,

$$
\mathbf{x}(k+1) = (I_{n\times n} \otimes A)\cdot\mathbf{x}(k) + L_1 \otimes BK \cdot \mathbf{x}(k) + L_2 \otimes BK \cdot \mathbf{x}(k-1) \tag{3.16}
$$

where $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_n]^T$, $L_1 \in R^{n\times n}$ and $L_2 \in R^{n\times n}$

$$
L_1 = \begin{bmatrix}
-\sum_{j=1}^{n} a_{1j}\theta & a_{12}\theta & \cdots & a_{1n}\theta_n \\
a_{21}\theta_1 & -\sum_{j=1}^{n} a_{2j}\theta_j & \cdots & a_{2n}\theta_n \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1}\theta_1 & a_{n2}\theta_2 & \cdots & -\sum_{j=1}^{n} a_{nj}\theta
\end{bmatrix}
$$

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

3.4 Summary

In summary, a discrete time system with a double integrator dynamics is proposed. The structure is modeled with a switching algebra graph, and each link is governed by Bernoulli random process. In this thesis, the random process is assumed to uniform for any communication link. It means links are subjected to the link failure or the data loss at the same time step. Since the communication constraints are related to

packet losses and delays. Bernoulli process can convert a packet delay case into a packet loss situation. For example, if a packet delay is over than a sampling time, then the data transmission is considered as lost. Therefore, with Bernoulli process implementation, the communication constraint then is mainly focused on the data loss.

Chapter 4

Controller Design

In this chapter, a controller is designed based on a leader following consensus. In order to develop a leader following consensus algorithm, an error dynamics is created by setting one of agents as a leader. Lyapunov-based methodologies and Linear matrix inequality (LMI) techniques are then applied to find an appropriate controller gain by satisfying the sufficient conditions of the error dynamics. Therefore, the controller with the solved gain by LMI is guaranteed to drive the system to reach a consensus.

4.1 Error Dynamics in Leader Following Consensus

As mentioned before, in order to verify the stability of the system, $lim_{t\to\infty}(||\mathbf{x}_i(t) - \mathbf{x}_j(t)||)^2$ has to reach zero, and where $\mathbf{x}_i(k) - \mathbf{x}_j(k) = \mathbf{e}(k)$, the information state error between Agent i and j. Since Agent j sends the signal, and can be defined as the leader. Agent i needs to reach Agent j in the term of information states to achieve the consensus. Then the next step is to find the error dynamics. At the Eq.(4.1), Agent j is set to be the leader, and labeled as 1.

$$\begin{aligned}\mathbf{e}_i(k+1) = \;& x_i(k+1) - x_1(k+1) \\ = \;& A(x_i(k) - x_1(k)) + \sum_{j=1}^{n} a_{ij}[\theta x_j(k) + (1-\theta)x_j(k-1) \\ & -\theta x_i(k) - (1-\theta)x_i(k-m)] - \sum_{j=1}^{n} a_{1j}[\theta x_j(k) + (1-\theta)x_j(k-1) \\ & +\theta x_1(k) - (1-\theta)x_1(k-m)]\end{aligned} \tag{4.1}$$

where $i = 2, 3, \cdots, n$. Since $A(x_i(k) - x_1(k))$ is already in the error dynamic form, then rearrange Eq.(3.13)

$$\mathbf{x}_i(k+1) = Ax_i(k) + BK\bar{x}_i(k+1)$$

where

$$\bar{\mathbf{x}}_i(k+1) = \sum_{j=1}^{n} a_{ij}[\theta x_j(k) + (1-\theta_j)x_j(k-1) - \theta x_i(k) - (1-\theta)x_i(k-1)] \quad (4.2)$$

and

$$\sum_{j=1}^{n} a_{ij}\theta_j x_1(k) - \sum_{j=1}^{n} a_{ij}\theta_j x_1(k) \quad (4.3)$$

$$\sum_{j=1}^{n} a_{ij}(1-\theta_j)x_1(k-1) - \sum_{j=1}^{n} a_{ij}(1-\theta_j)x_1(k-1). \quad (4.4)$$

Add Eq.(4.3) and Eq.(4.4) into Eq.(4.2), then

$$\begin{aligned}\bar{\mathbf{x}}_i(k+1) = \; & \sum_{j=1}^{n} a_{ij}(\theta x_j(k) - \theta x_1(k)) + \sum_{j=1}^{n} a_{ij}[(1-\theta)x_j(k-1) - (1-\theta)x_1(k-1)] \\ & - \sum_{j=1}^{n} a_{ij}(x_i\theta - x_1\theta) - \sum_{j=1}^{n} a_{ij}[(1-\theta)(x_j(k-1) - (1-\theta)x_1(k-1)]\end{aligned} \quad (4.5)$$

and

$$-\bar{\mathbf{x}}_1(k+1) = -\sum_{j=1}^{n} a_{1j}(\theta_j x_j(k) - \theta_j x_1(k)) - \sum_{j=1}^{n} a_{1j}[(1-\theta_j)x_j(k-1) - (1-\theta_j)x_1(k-1)]. \quad (4.6)$$

By summing Eq.(4.5) and Eq.(4.6), then the general error dynamics becomes

$$\begin{aligned}\mathbf{e}_i(k+1) \; = \; & Ae_i(k) + BK\sum_{j=2}^{n}(a_{ij} - a_{1j})\theta e_j(k) + BK\sum_{j=2}^{n}(a_{ij} - a_{1j})(1-\theta)e_j(k-1) \\ & - BK\sum_{j=1}^{n} a_{ij}\theta e_i(k) - BK\sum_{j=1}^{n} a_{ij}(1-\theta)e_i(k-1)\end{aligned} \quad (4.7)$$

where

$$\begin{aligned} e_i(k+1) &= x_i(k+1) - x_1(k+1) \\ e_i(k) &= x_i(k) - x_1(k) \\ e_j(k) &= x_j(k) - x_1(k) \\ e_j(k-1) &= x_j(k-1) - x_1(k-1) \\ e_i(k-1) &= x_i(k-1) - x_1(k-1) \end{aligned}$$

with $i \neq j$, $i = 2, 3, ..., n$. Now, write Eq.(4.7) into a matrix form, then

$$
\begin{aligned}
e_2(k+1) &= Ae_2(k) + BK\sum_{j=2}^{n}(a_{2j}-a_{1j})\theta e_j(k) \\
&\quad +BK\sum_{j=2}^{n}(a_{2j}-a_{1j})(1-\theta)e_j(k-1) - BK\sum_{j=1}^{n}a_{2j}\theta e_2(k) \\
&\quad -BK\sum_{j=1}^{n}a_{2j}(1-\theta)e_2(k-1) \\
e_3(k+1) &= Ae_3(k) + BK\sum_{j=2}^{n}(a_{3j}-a_{1j})\theta_j e_j(k) \\
&\quad +BK\sum_{j=2}^{n}(a_{3j}-a_{1j})(1-\theta)e_j(k-1) - BK\sum_{j=1}^{n}a_{3j}\theta e_3(k) \\
&\quad -BK\sum_{j=1}^{n}a_{3j}(1-\theta)e_3(k-1) \\
&\vdots \qquad \vdots \\
e_n(k+1) &= Ae_n(k) + BK\sum_{j=2}^{n}(a_{nj}-a_{1j})\theta e_j(k) \\
&\quad +BK\sum_{j=2}^{n}(a_{nj}-a_{1j})(1-\theta)e_j(k-1) - BK\sum_{j=1}^{n}a_{nj}\theta e_n(k) \\
&\quad -BK\sum_{j=1}^{n}a_{nj}(1-\theta)e_n(k-1)
\end{aligned}
\tag{4.8}
$$

$$
\begin{aligned}
\mathbf{e}_i(k+1) &= A\mathbf{e}_i(k) + BK\sum_{j=2}^{n}(a_{ij}-a_{1j})\theta\mathbf{e}_j(k) + BK\sum_{j=2}^{n}(a_{ij}-a_{1j})(1-\theta)\mathbf{e}_j(k-1) \\
&\quad -BK\sum_{j=1}^{n}a_{ij}\theta\mathbf{e}_i(k) - BK\sum_{j=1}^{n}a_{ij}(1-\theta)\mathbf{e}_i(k-1)
\end{aligned}
\tag{4.9}
$$

Therefore, from Eq.(4.9), a general matrix form of error dynamics can be presented as

$$
\begin{aligned}
\mathbf{e}(k+1) &= (I_{n-1\times n-1}\otimes A)\cdot\mathbf{e}(k) + \tilde{L}_1\otimes BK\cdot e(k) + \tilde{L}_2\otimes BK\cdot\mathbf{e}(k-1) \\
&= (I_{n-1\times n-1}\otimes A + \tilde{L}_1\otimes BK)\cdot\mathbf{e}(k) + \tilde{L}_2\otimes BK\cdot\mathbf{e}(k-1) \\
&= (I\otimes A + \tilde{L}_1\otimes BK)\cdot\mathbf{e}(k) + \tilde{L}_2\otimes BK\cdot\mathbf{e}(k-1)
\end{aligned}
\tag{4.10}
$$

where $\mathbf{e} = [e_2, e_3, ..., e_n]^T$,$I \in R^{n-1\times n-1}$, $\tilde{L}_1 \in R^{n-1\times n-1}$ and $\tilde{L}_2 \in R^{n-1\times n-1}$. The system (3.16) asymptotically reaches consensus in mean square if and only if $lim_{t\to\infty}E(||\mathbf{e}(k+1)||^2) = 0$. $\otimes$ is called Kronecker product, which is an operation

on two matrices of arbitrary size resulting in a block matrix [52]. If A is an m × q matrix and B is a p × q matrix, then the Kronecker product A ⊗ B is the mp × nq block matrix.

$$\tilde{L}_1 = \begin{bmatrix} (a_{22}-a_{12})\theta - O_1 & (a_{23}-a_{13})\theta & \cdots & (a_{2n}-a_{1n})\theta \\ (a_{32}-a_{12})\theta & (a_{33}-a_{13})\theta - O_2 & \cdots & (a_{3n}-a_{1n})\theta \\ \vdots & \vdots & \ddots & \vdots \\ (a_{n2}-a_{12})\theta & (a_{n3}-a_{13})\theta_3 & \cdots & (a_{nn}-a_{1n})\theta_n - O_3 \end{bmatrix}$$

where $n \geq 2$,thus, $O_1 = \sum_{j=1}^{n} a_{2j}\theta$, $O_2 = \sum_{j=1}^{n} a_{3j}\theta$, $O_3 = \sum_{j=1}^{n} a_{nj}\theta$, and $\tilde{L}_1 = [\tilde{l}_{ij}]_{(n-1)\times(n-1)}$, where $\tilde{l}_{ij} = (a_{(i+1)(j+1)} - a_{1(j+1)})\theta_{(j+1)}$, when $i \neq j$ and $\tilde{l}_{ii} = (a_{(i+1)(i+1)} - a_{1(i+1)})\theta - \sum_{j=1}^{n} a_{(i+1)j}\theta$.

$$\tilde{L}_2 = \begin{bmatrix} (a_{22}-a_{12})S_1 - Q_1 & (a_{23}-a_{13})S_2 & \cdots & (a_{2n}-a_{1n})S_n \\ (a_{32}-a_{12})S_1 & (a_{33}-a_{13})S_2 - Q_2 & \cdots & (a_{3n}-a_{1n})S_n \\ \vdots & \vdots & \ddots & \vdots \\ (a_{n2}-a_{12})S_1 & (a_{n3}-a_{13})S_2 & \cdots & (a_{nn}-a_{1n})S_n - Q_3 \end{bmatrix}$$

where $n \geq 2$,thus, $Q_1 = \sum_{j=1}^{n} a_{2j}S_j$, $S = (1-\theta)$, $Q_2 = \sum_{j=1}^{n} a_{3j}S_j$, $Q_3 = \sum_{j=1}^{n} a_{nj}S_j$, and $\tilde{L}_2 = [\tilde{l}_{ij}]_{(n-1)\times(n-1)}$, where $\tilde{l}_{ij} = (a_{(i+1)(j+1)} - a_{1(j+1)})(1-\theta)$, when $i \neq j$ and $\tilde{l}_{ii} = (a_{(i+1)(i+1)} - a_{1(i+1)})(1-\theta) - \sum_{j=1}^{n} a_{(i+1)j}(1-\theta)$.

4.2 Controller Gain Design

Theorem 1. *For the uncertain networked control system* (3.16)*, if there exist positive-definite matrices,* $\hat{H}_0 \in R^{2(n-1)\times 2(n-1)} > 0$, $\hat{H}_1 \in R^{2(n-1)\times 2(n-1)} > 0$ *matrices* $K \in R^{1\times n}$, $P_1 = \epsilon P_0$, $P_0^{-1} = X \in R^{2(n-1)\times 2(n-1)}$, $X = diag[X_m, X_m]$, $KX_m = Y_m \in R^{1\times n}$, *such that both following matrix inequalities Eq.(4.11) and Eq.(4.12) hold,*

$$\begin{bmatrix} \Xi_{11} & 0 & \Xi_{13} & \Xi_{14} \\ * & -\hat{H}_0 & \Xi_{23} & \Xi_{24} \\ * & * & -(1-r)(\frac{1}{\epsilon})X & 0 \\ * & * & * & -rX \end{bmatrix} < 0, \tag{4.11}$$

where

$$\begin{aligned}
\Xi_{11} = &\quad r\hat{H}_0 + (1-r)\hat{H}_1 - X \\
\Xi_{13} = &\quad (I \otimes X_m A^T)(1-r) \\
\Xi_{14} = &\quad (I \otimes X_m A^T)r \\
\Xi_{23} = &\quad (\tilde{L}_{20}^T \otimes Y_m^T B^T)(1-r) \\
\Xi_{24} = &\quad (\tilde{L}_{20}^T \otimes Y_m^T B^T)r(\tfrac{1}{\epsilon})
\end{aligned}$$

with

$$\tilde{L}_{20} = \begin{bmatrix}
(a_{22} - a_{12}) - \sum_{j=1}^{n} a_{2j} & (a_{23} - a_{13}) & \cdots & (a_{2n} - a_{1n}) \\
(a_{32} - a_{12}) & (a_{33} - a_{13}) - \sum_{j=1}^{n} a_{3j} & \cdots & (a_{3n} - a_{1n}) \\
\vdots & \vdots & \ddots & \vdots \\
(a_{n2} - a_{12}) & (a_{n3} - a_{13}) & \cdots & (a_{nn} - a_{1n}) - \sum_{j=1}^{n} a_{nj}
\end{bmatrix}$$

and

$$\begin{bmatrix}
\Gamma_{11} & 0 & \Gamma_{13} & \Gamma_{14} \\
* & -\hat{H}_0 & 0 & 0 \\
* & * & -(1-r)(\frac{1}{\epsilon})X & 0 \\
* & * & * & -rX
\end{bmatrix} < 0, \tag{4.12}$$

where

$$\begin{aligned}
\Gamma_{11} = &\quad r\hat{H}_0 + (1-r)\hat{H}_1 - X \\
\Gamma_{13} = &\quad (I \otimes X_m A^T)(1-r) + (\tilde{L}_{11}^T \otimes Y_m^T B^T)(1-r) \\
\Gamma_{14} = &\quad (I \otimes X_m A^T)r + (\tilde{L}_{11}^T \otimes Y_m^T B^T)r
\end{aligned}$$

with

$$\tilde{L}_{11} = \begin{bmatrix}
(a_{22} - a_{12}) - \sum_{j=1}^{n} a_{2j} & (a_{23} - a_{13}) & \cdots & (a_{2n} - a_{1n}) \\
(a_{32} - a_{12}) & (a_{33} - a_{13}) - \sum_{j=1}^{n} a_{3j} & \cdots & (a_{3n} - a_{1n}) \\
\vdots & \vdots & \ddots & \vdots \\
(a_{n2} - a_{12}) & (a_{n3} - a_{13}) & \cdots & (a_{nn} - a_{1n}) - \sum_{j=1}^{n} a_{nj}
\end{bmatrix}$$

with the controller gain $K = Y_m X_m^{-1}$, the system (3.16) is then mean square stable.

Proof :
Choose a Lyapunov functional candidate for it follows

$$V_i = e_i^T \bar{P}_i e_i + e_i^T(k-1)\bar{Q}_i e_i(k-1) + \sum_{j=1, j\neq i}^{n} e_j^T(k-1)\bar{R}_j e_j(k-1) \tag{4.13}$$

then,

$$V(k) = \mathbf{e}^T(k)P_s\mathbf{e}(k) + \mathbf{e}^T(k-1)H_s\mathbf{e}(k-1) \tag{4.14}$$

where $P = diag[\bar{P}_1, ..., \bar{P}_n]$, $H = diag[\bar{Q}_1 + \sum_{j=1}^{n}\bar{R}_j, ..., \bar{Q}_n + \sum_{j=1}^{n}\bar{R}_j]$. $\bar{P}$, $\bar{Q}$, and $\bar{R}$ are positive-defined matrices with appropriate dimensions. The packet loss probability is defined as $Pr(\theta(k) = 0) = r$, $\theta(k) = s$, and $s \in \{0, 1\}$.

$$\begin{aligned}
& E\{V(k+1)|V(k)\} - V(k) \\
= \;& E\{\mathbf{e}^T(k+1)P_s\mathbf{e}(k+1) + \mathbf{e}^T(k)H_s\mathbf{e}(k)\} - (\mathbf{e}^T(k)P_s\mathbf{e}(k) + \mathbf{e}^T(k-1)H_s\mathbf{e}(k-1)) \\
= \;& E\{[(I\otimes A + \tilde{L}_{1s}\otimes BK)\cdot\mathbf{e}(k) + \tilde{L}_{2s}\otimes BK\cdot\mathbf{e}(k-1)]^T(rP_0 + (1-r)P_1) \\
& \cdot[(I\otimes A + \tilde{L}_{1s}\otimes BK)\cdot\mathbf{e}(k) + \tilde{L}_{2s}\otimes BK\cdot\mathbf{e}(k-1)]\} \\
& + E\{\mathbf{e}^T(k)(rH_0 + (1-r)H_1)\mathbf{e}(k)\} - \mathbf{e}^T(k)P_0\mathbf{e}(k) - \mathbf{e}^T(k-1)H_0\mathbf{e}(k-1) \\
= \;& [(I\otimes A + \tilde{L}_{1s}\otimes BK)e(k) + (\tilde{L}_{2s}\otimes BK)\mathbf{e}(k-1)]^T(rP_0 + (1-r)P_1) \\
& \cdot[(I\otimes A + \tilde{L}_{1s}\otimes BK)\mathbf{e}(k) + \tilde{L}_{2s}\otimes BK\cdot\mathbf{e}(k-1)] \\
& + \mathbf{e}^T(k)(rH_0 + (1-r)H_1)\mathbf{e}(k) - \mathbf{e}^T(k)P_0\mathbf{e}(k) - \mathbf{e}^T(k-1)H_0\mathbf{e}(k-1)
\end{aligned} \tag{4.15}$$

Set Eq.(4.15) as follows

$$\begin{aligned}
& E\{V(k+1)|V(k)\} - V(k) \\
& = \underbrace{[D]^T(rP_0 + (1-rP_1)[D]}_{\alpha} \\
& \quad + \underbrace{\mathbf{e}^T(k)(rH_0 + (1-r)H_1)\mathbf{e}(k) - \mathbf{e}^T(k)P_0\mathbf{e}(k) - \mathbf{e}^T(k-1)H_0\mathbf{e}(k-1)}_{\beta}
\end{aligned} \tag{4.16}$$

where $D = (I\otimes A + \tilde{L}_{1s}\otimes BK)\mathbf{e}(k) + (\tilde{L}_{2s}\otimes BK)\mathbf{e}(k-1)$. If the $E\{V(k+1)|V(k)\} - V(k) < 0$, then $lim_{t\to\infty}E(||\mathbf{e}(k+1)||^2) = 0$, the system (3.9) is mean square stable, e.g. $e_i(k+1)$ tends to zero asymptotically, which means Agent i in the group is able to track the leader's trajectory.
where

$$
\begin{aligned}
\alpha = \; & [(I \otimes A + \tilde{L}_{1s} \otimes BK)\mathbf{e}(k) + \tilde{L}_{2s} \otimes BK\mathbf{e}(k-1)]^T (rP_0 + (1-r)P_1) \\
& \cdot[(I \otimes A + \tilde{L}_{1s} \otimes BK)\mathbf{e}(k) + \tilde{L}_{2s} \otimes BK\mathbf{e}(k-1)] \\
= \; & \begin{bmatrix} e(k) \\ e(k-1) \end{bmatrix}^T \begin{bmatrix} (I \otimes A + \tilde{L}_{1s} \otimes BK) & \tilde{L}_{2s} \otimes BK \end{bmatrix}^T (rP_0 + (1-r)P_1) \\
& \cdot \begin{bmatrix} (I \otimes A + \tilde{L}_{1s} \otimes BK) & \tilde{L}_{2s} \otimes BK \end{bmatrix} \begin{bmatrix} e(k) \\ e(k-1) \end{bmatrix} \\
= \; & \bar{\mathbf{e}}^T M^T (rP_0 + (1-r)P_1) M\bar{\mathbf{e}} \\
= \; & \bar{\mathbf{e}}^T [M^T (rP_0)M + M^T (1-r)P_1 M]\bar{\mathbf{e}} \\
= \; & \bar{\mathbf{e}}^T \Pi_1 \bar{\mathbf{e}}
\end{aligned}
$$

where

$$
\begin{aligned}
M = \; & \begin{bmatrix} (I \otimes A + \tilde{L}_{1s} \otimes BK) & \tilde{L}_{2s} \otimes BK \end{bmatrix} \\
\Pi_1 = \; & M^T (rP_0)M + M^T (1-r)P_1 M \\
\bar{\mathbf{e}} = \; & \begin{bmatrix} \mathbf{e}(k) & \mathbf{e}(k-1) \end{bmatrix}^T
\end{aligned}
$$

and

$$
\begin{aligned}
\beta = \; & \mathbf{e}^T(k)(rH_0 + (1-r)H_1)\mathbf{e}(k) - \mathbf{e}^T(k)P_0\mathbf{e}(k) - \mathbf{e}^T(k-1)H_0\mathbf{e}(k-1) \\
= \; & \begin{bmatrix} \mathbf{e}(k) \\ \mathbf{e}(k-1) \end{bmatrix}^T \begin{bmatrix} (rH_0 + (1-r)H_1) - P_0 & 0 \\ 0 & -H_0 \end{bmatrix} \begin{bmatrix} \mathbf{e}(k) \\ \mathbf{e}(k-1) \end{bmatrix} \\
= \; & \bar{\mathbf{e}}^T \begin{bmatrix} (rH_0 + (1-r)H_1) - P_0 & 0 \\ 0 & -H_0 \end{bmatrix} \bar{\mathbf{e}} \\
= \; & \bar{\mathbf{e}}^T \Pi_2 \bar{\mathbf{e}}
\end{aligned}
$$

where

$$
\Pi_2 = \begin{bmatrix} (rH_0 + (1-r)H_1) - P_0 & 0 \\ 0 & -H_0 \end{bmatrix} \tag{4.17}
$$

$$
\Pi = (\Pi_1 + \Pi_2) \tag{4.18}
$$

If $E\{V(k+1)|V(k)\} - V(k) < 0$ holds, then

$$
\alpha + \beta = \bar{\mathbf{e}}^T (\Pi_1 + \Pi_2)\bar{\mathbf{e}} < 0 \tag{4.19}
$$

By applying Schur Complement to Eq.(4.18)

$$\begin{aligned}\Pi = &\ M^T(rP_0)M + M^T(1-r)P_1M + \Pi_2 \\ = &\ M^T(rP_0)(rP_0)^{-1}(rP_0)M + M^T(1-r)P_1M + \Pi_2 \\ = &\ \begin{bmatrix} M^T(1-r)P_1M + \Pi_2 & M^T(rP_0) \\ (rP_0)M & -rP_0 \end{bmatrix} \\ < &\ 0\end{aligned} \tag{4.20}$$

then by using Schur Complement again to $M^T(1-r)P_1M + \Pi_2$,

$$M^T(1-r)P_1((1-r)P_1)^{-1}(1-r)P_1M + \Pi_2 = \begin{bmatrix} \Pi_2 & M^T(1-r)P_1 \\ (1-r)P_1M & -(1-r)P_1 \end{bmatrix} \tag{4.21}$$

Then put Eq.(4.21) into Eq.(4.20),

$$\Pi = \begin{bmatrix} \Pi_2 & M^T(1-r)P_1 & M^T(rP_0) \\ (1-r)P_1M & -(1-r)P_1 & 0 \\ (rP_0)M & 0 & -rP_0 \end{bmatrix} < 0 \tag{4.22}$$

Combine Eq.(4.17) with Eq.(4.22), thus

$$\Pi = \begin{bmatrix} \Pi_{11} & 0 & \Pi_{13} & \Pi_{14} \\ * & -H_0 & \Pi_{23} & \Pi_{24} \\ * & * & -(1-r)P_1 & 0 \\ * & * & * & -rP_0 \end{bmatrix} < 0 \tag{4.23}$$

where

$$\begin{aligned}\Pi_{11} = &\ rH_0 + (1-r)H_1 - P_0 \\ \Pi_{13} = &\ (I \otimes A + \tilde{L}_{1s} \otimes BK)^T(1-r)P_1 \\ \Pi_{14} = &\ (I \otimes A + \tilde{L}_{1s} \otimes BK)^T(rP_0) \\ \Pi_{23} = &\ (\tilde{L}_{2s} \otimes BK)^T(1-r)P_1 \\ \Pi_{24} = &\ (\tilde{L}_{2s} \otimes BK)^T(rP_0)\end{aligned}$$

Now, pre- and post-multiply the both side of Eq.(4.23) by $diag(P_0^{-1}, P_0^{-1}, P_1^{-1}, P_0^{-1})$,

$$\Xi = \begin{bmatrix} P_0^{-1} & & & \\ & P_0^{-1} & & \\ & & P_1^{-1} & \\ & & & P_0^{-1} \end{bmatrix} \Pi \begin{bmatrix} P_0^{-1} & & & \\ & P_0^{-1} & & \\ & & P_1^{-1} & \\ & & & P_0^{-1} \end{bmatrix}$$

$$\Xi = \begin{bmatrix} \Xi_{11} & 0 & \Xi_{13} & \Xi_{14} \\ * & -P_0^{-1}H_0P_0^{-1} & \Xi_{23} & \Xi_{24} \\ * & * & -(1-r)P_1^{-1} & 0 \\ * & * & * & -rP_0^{-1} \end{bmatrix}$$

where

$$\begin{aligned} \Xi_{11} = &\ rP_0^{-1}H_0P_0^{-1} + (1-r)P_0^{-1}H_1P_0^{-1} - P_0^{-1} \\ \Xi_{13} = &\ P_0^{-1}(I \otimes A + \tilde{L}_1 \otimes BK)^T(1-r) \\ \Xi_{14} = &\ P_0^{-1}(I \otimes A + \tilde{L}_1 \otimes BK)^T r \\ \Xi_{23} = &\ P_0^{-1}(\tilde{L}_2 \otimes BK)^T(1-r) \\ \Xi_{24} = &\ P_0^{-1}(\tilde{L}_2 \otimes BK)^T r(P_0P_1^{-1}) \end{aligned}$$

Let $P_0 = diag[P_m,\, P_m,\, ...]_{2(n-1)\times 2(n-1)}$,$P_m^{-1} = X_m \in R^{2\times 2}$,$X = diag[X_m,\, X_m,\, ...]_{2(n-1)\times 2(n-1)}$,$KX_m = Y_m \in R^{1\times 2}$,$\hat{H}_0 = P_0^{-1}H_0P_0^{-1} > 0$,$\hat{H}_1 = P_0^{-1}H_1P_0^{-1} > 0$, assume $P_1 = \epsilon P_0$, and $\epsilon > 0$, $P_1^{-1} = (\frac{1}{\epsilon})X$

$$\Xi = \begin{bmatrix} \Xi_{11} & 0 & \Xi_{13} & \Xi_{14} \\ * & -\hat{H}_0 & \Xi_{23} & \Xi_{24} \\ * & * & -(1-r)(\frac{1}{\epsilon})X & 0 \\ * & * & * & -rX \end{bmatrix} \tag{4.24}$$

where

$$\begin{aligned} \Xi_{11} = &\ r\hat{H}_0 + (1-r)\hat{H}_1 - X \\ \Xi_{13} = &\ (I \otimes X_mA^T)(1-r) + (\tilde{L}_{1s}^T \otimes Y_m^TB^T)(1-r) \\ \Xi_{14} = &\ (I \otimes X_mA^T)r + (\tilde{L}_{1s}^T \otimes Y_m^TB^T)r \\ \Xi_{23} = &\ (\tilde{L}_{2s}^T \otimes Y_m^TB^T)(1-r) \\ \Xi_{24} = &\ (\tilde{L}_{2s}^T \otimes Y_m^TB^T)r(\tfrac{1}{\epsilon}) \end{aligned}$$

Since the $s \in \{0, 1\}$, Eq.(4.23) can be considered into two cases,

Case 1: s = 0, Packet Lost

Eq.(4.23) becomes

$$\Pi = \begin{bmatrix} \Pi_{11} & 0 & \Pi_{13} & \Pi_{14} \\ * & -H_0 & \Pi_{23} & \Pi_{24} \\ * & * & -(1-r)P_1 & 0 \\ * & * & * & -rP_0 \end{bmatrix} < 0 \tag{4.25}$$

where

$$\begin{aligned}
\Pi_{11} = &\quad rH_0 + (1-r)H_1 - P_0 \\
\Pi_{13} = &\quad (I \otimes A)^T (1-r) P_1 \\
\Pi_{14} = &\quad (I \otimes A)^T (rP_0) \\
\Pi_{23} = &\quad (\tilde{L}_{20} \otimes BK)^T (1-r) P_1 \\
\Pi_{24} = &\quad (\tilde{L}_{20} \otimes BK)^T (rP_0)
\end{aligned}$$

with

$$\tilde{L}_{20} = \begin{bmatrix}
(a_{22} - a_{12}) - \sum_{j=1}^{n} a_{2j} & (a_{23} - a_{13}) & \cdots & (a_{2n} - a_{1n}) \\
(a_{32} - a_{12}) & (a_{33} - a_{13}) - \sum_{j=1}^{n} a_{3j} & \cdots & (a_{3n} - a_{1n}) \\
\vdots & \vdots & \ddots & \vdots \\
(a_{n2} - a_{12}) & (a_{n3} - a_{13}) & \cdots & (a_{nn} - a_{1n}) - \sum_{j=1}^{n} a_{nj}
\end{bmatrix}$$

Case 2: s = 1, Packet Received

Eq.(4.23) becomes

$$\Pi = \begin{bmatrix}
\Pi_{11} & 0 & \Pi_{13} & \Pi_{14} \\
* & -H_0 & 0 & 0 \\
* & * & -(1-r)P_1 & 0 \\
* & * & * & -rP_0
\end{bmatrix} < 0 \tag{4.26}$$

where

$$\begin{aligned}
\Pi_{11} = &\quad rH_0 + (1-r)H_1 - P_0 \\
\Pi_{13} = &\quad (I \otimes A + \tilde{L}_{11} \otimes BK)^T (1-r) P_1 \\
\Pi_{14} = &\quad (I \otimes A + \tilde{L}_{11} \otimes BK)^T (rP_0)
\end{aligned}$$

with

$$\tilde{L}_{11} = \begin{bmatrix} (a_{22} - a_{12}) - \sum_{j=1}^{n} a_{2j} & (a_{23} - a_{13}) & \cdots & (a_{2n} - a_{1n}) \\ (a_{32} - a_{12}) & (a_{33} - a_{13}) - \sum_{j=1}^{n} a_{3j} & \cdots & (a_{3n} - a_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ (a_{n2} - a_{12}) & (a_{n3} - a_{13}) & \cdots & (a_{nn} - a_{1n}) - \sum_{j=1}^{n} a_{nj} \end{bmatrix}$$

If the both inequalities Eq.(4.25) and Eq.(4.26) hold,, then $E\{V(k+1)|V(k)\} - V(k) < 0$, which means system (3.9) is mean square stable.

4.3 Selection of Parameter in Theorem 1

In Theorem 1, the inequality Eq.(4.24) is derived by using Lyapunov function. In order to ensure this inequality holds, Linear matrix inequality technique is implemented based on pre- and post-multiply the both side of Eq.(4.23) by $diag(P_0^{-1}, P_0^{-1}, P_1^{-1}, P_0^{-1})$. If the LMI is feasible, then the inequality holds. H, P_1, P_0, X_m, Y_m are assumed to be known and existing positive-define matrices, A, B matrices are dependent on the sampling time while the continues-time system is converted into discrete-time system. Moreover, r is the loss rate, and ϵ is an arbitrary number which is the fraction of P_1 to P_0. In this thesis, $\epsilon = 0.5$. If all selected parameters make LMI feasible, which is a sufficient condition, then Theorem 1 is valid.

4.4 Summary

Chapter 4 is the main theory contribution. A double integrator dynamics with discrete-time system associated with a proposed controller is defined for a multi-agent system. A algebra graph is to represent the topology of the system, and Bernoulli process is to represent the random data loss situation among the system. In order to ensure the followers Agent $2, ..., n$ to reach the consensus with the leader, Agent 1, an appropriate controller gain is obtained by implementing Lyapunov-based methodologies and Linear Matrix Inequality (LMI) techniques.

Chapter 5

Simulation Results and Parametric Studies

The simulation results are simulated in Matlab, and analyzed in several different scenarios to verify the effectiveness of the controller designed in Theorem 1. Matlab is used to solve the LMI problems as described in Chapter 4 to find a sufficient condition in order to obtain a controller gain K. The following sections discuss the effect on efficiency of consensus in four cases; 1) effect of data loss rate(r); 2) effect of communication weight ω_{ij}; 3) effect of initial values; 4) effect of sampling time; and finally, a five agent example is studied as well to verify the effectiveness of controller in a large number of agent system.

5.1 Effect of Data Loss Rates

In this section, the simulation results are focused on studying consensus ability and consensus (convergence) time on different examples with eight data loss rates from 5% to 96%. Since a data loss rate is very small in a stable condition, six out of eight examples are studied within 50% of data loss rate.

5.1.1 Simulation Configuration

The topology of multi-agent system is modeled by a directed graph with three vertices, means three agents, shown in Fig. 5.1. Agent 1 is considered as a leader, which is only capable of sending signals to the rest of agents, and Agent 2 and 3 are followers. They can receive the signals from Agent 1. Beside the communication between the leader and followers, among followers, there is also a communication channel. Therefore, followers can not only receive signals from the leader but also from the neighboring agent. In case of the failure of communication link between Agent 1 and 2, Agent 2 can still receive information from Agent 1 indirectly by passing through Agent 3. Thus this topology maximizes the formidability during the operation. Moreover, the communication weight $\omega_{ij} = 1$ for all channels, and sampling time $Ts = 0.1$ s throughout this section. In order to each consensus, one of conditions is to ensure that zero is a simple eigenvalue of Laplacian matrix L (refer to Chapter 2).

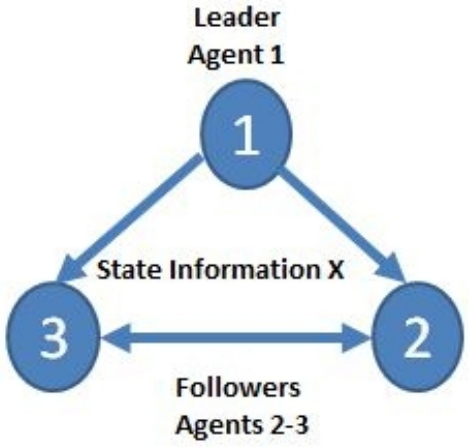


Figure 5.1: A Group of Three Agents in A Directed Graph Topology

Based on Fig. 5.1. the adjacency matrix A_a and Laplacian matrix L are respectively:

$$A_a = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{5.1}$$

The eigenvalues of L are $0, 3, 5$, which satisfies the condition that zero is a simple eigenvalue of Laplacian matrix L . Therefore this topology of the system can be consentable.

5.1.2 Simulation Results

In this section, eleven simulation results were investigated and studied in order to show how the data loss rate affect the consensus ability and time with the designed controller.

Example 1: r=5%

Firstly, a 5% data loss rate is studied. The controller gain (K) is obtained by LMI with the data loss rate bound set as 0.05. With a group of three agents, set $\hat{H}_0 = diag[H_{01}, H_{02}]^{4\times4}, \hat{H}_1 = diag[H_{11}, H_{12}]^{4\times4}$ then the control gain is designed from this case based on the sufficient condition in Theorem 1 (Referred to Chapter 4). Since this gain is only satisfied a sufficient condition of LMI, it might not be a optimal gain but it does guarantee the consensus of the system when the packet loss rate is no more than 5%.

$$K = Y_m X_m^{-1} = [\ 0.2713 \quad 2.0399\ ]. \tag{5.2}$$

Based on this gain, Fig. 5.2 is the result of example 1. Information states x_1 and x_2 were both converged with agent 1. If each agent represents a mobile robot, and then x_1 is the position, and x_2 will be the speed. In this figure, it does not show the actual moving trajectory of agents but information state consensus. Fig. 5.2 (a) is one dimensional, meaning all agents are located in one straight line, and only able to move in this line backward or forward. However, for a actual trajectory in a X-Y plane, it will be shown on in Chapter 6. In which, the system will be modeled by the kinematic model of Pioneer 3 mobile robots. Fig. 5.2 (b) Agent 1 started at 0 with initial speed of 1 m/s, while Agent 2 and 3 start at position 5 and 10, which means they are from 5 and 10 m away from the origin 0 in the one dimensional plane, with initial speed of 0 m/s. At 1 second, Agent 2 is speeded up to 0.29 m/s toward to Agent 1. Since the speed direction of Agent 1 is defined as positive, and then the -0.39 m/s means Agent 3 is moving in an opposite direction of Agent 1 while it is approaching. Fig. 5.3(a) shows mean square error of Agent 2 and 3, respected to Agent 1, and the mean square error is defined in Eq.(5.3).

$$Error = \sqrt{(x_{1i} - x_{11})^2 + ((x_{2i} - x_{21})Ts)^2}, \tag{5.3}$$

where x_{1i} and x_{2i} are the current position and speed of Agent i, the i is 1,2,3,...,n. Based on Eq.(5.3), the error reaches zero if position and speed of Agent 2 and 3 are both converged to Agent 1. Fig. 5.3(b) shows the distribution of data loss rate among the system. The Bernoulli variable is represented as θ. If $\theta = 1$, it means the packet is received by followers at this particular time step. If $\theta = 0$, it means the packet is lost. In order to determine the consensus time, arbitrary bound is introduced and set as $b = 0.05$. If the *Error* is under 0.05, then the system is considered as consensus achievement. Therefore, from Fig. 5.3, at 38 second, the system is converged completely. Fig. 5.4 shows controller input of gain (K) in respect of time. The higher number of input in magnitude means the lager effort of controller acting on agents. Once the system reaches consensus, the input will become zero. The controller input is defined in Eq.(5.4).

$$u = -K\mathbf{x}, \tag{5.4}$$

where K is the gain from Eq.(5.2), and $\mathbf{x}$ is $[x_1, x_2]^T$.

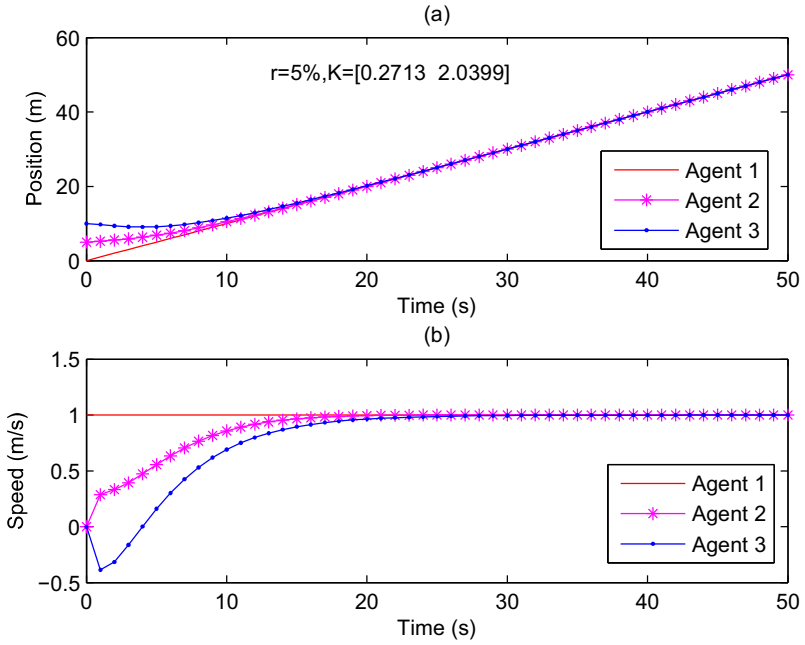


Figure 5.2: Simulation Results of Position and Speed in One Dimensional Plane with r=5%

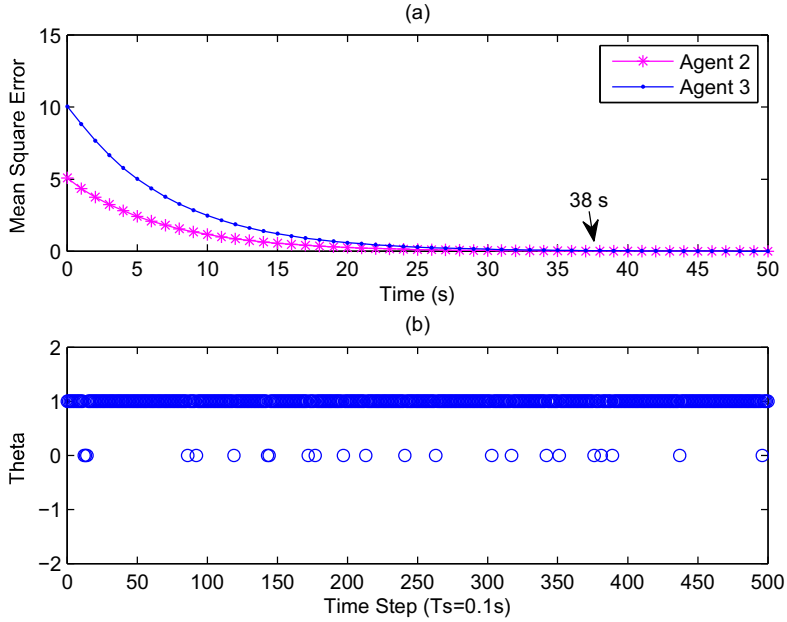


Figure 5.3: Mean Square Error and Loss Rate Distribution with r=5%

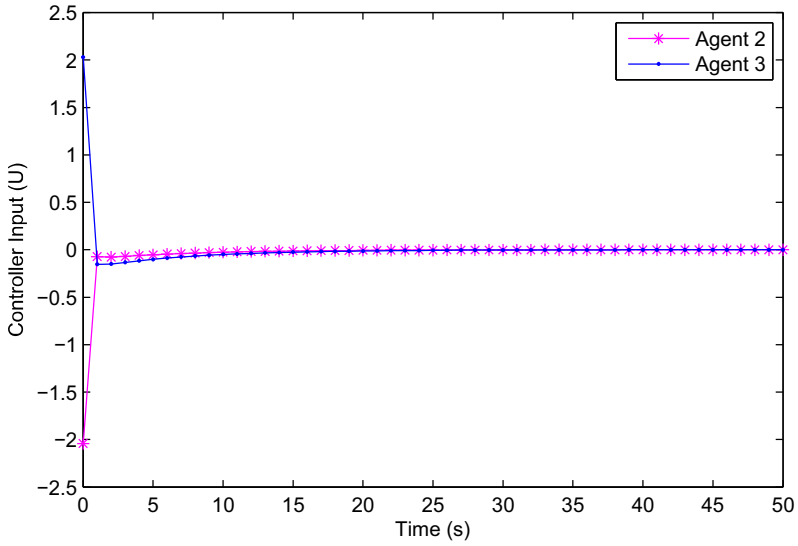


Figure 5.4: Controller Input (u) with r=5%

Example 2: r=10%

In the second example, the data loss rate was set as 10%. Therefore the controller gain (K) was obtained by LMI with the data loss rate bound set as 0.1. Then

$$K = [0.1150\ 1.0506]. \tag{5.5}$$

The rest of conditions are the same with Example 1. Fig. 5.5(b) shows Agent 2 and 3 are raised up to 0.4 and 0.1 m/s at 1 second, respectively. Fig. 5.6(a) shows the mean square error, in which the system reaches consensus at 44 second. Fig. 5.7 again shows the controller input into Agent 2 and 3.

Example 3: r=20%

In this example, the data loss rate is set to be 20%. Therefore the controller gain (K) was obtained by LMI with the data loss rate bound set as 0.2. Then

$$K = [0.1036\ 0.6603]. \tag{5.6}$$

Fig. 5.8 shows the simulation result in terms of the position and the speed of agents. Fig. 5.9 shows the mean square error and data loss distribution at $r = 20\%$. Fig. 5.10 shows the controller input into Agent 2 and 3. Compared to Fig. 5.7 while the loss rate is at 10%, the controller effort in Fig. 5.10 is much intensive from the first 10

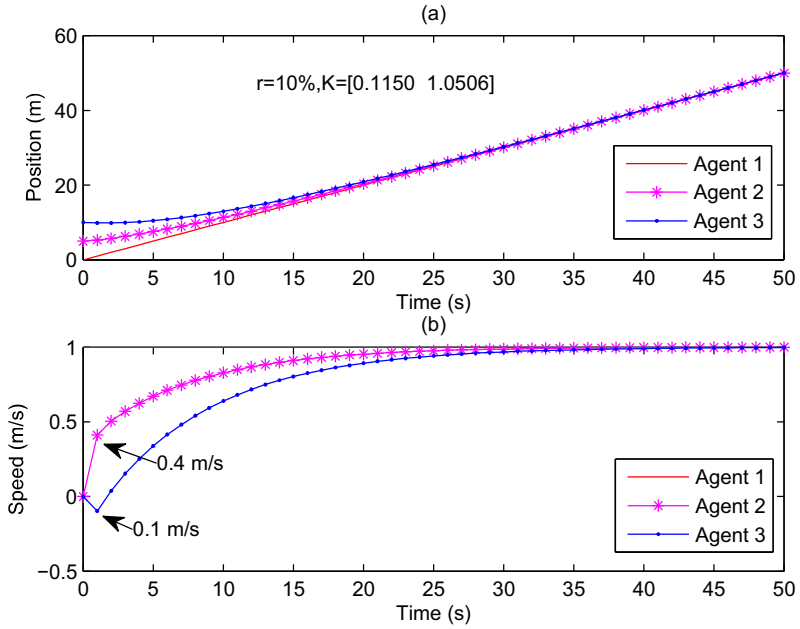


Figure 5.5: Simulation Results of Position and Speed in One Dimensional Plane with r=10%

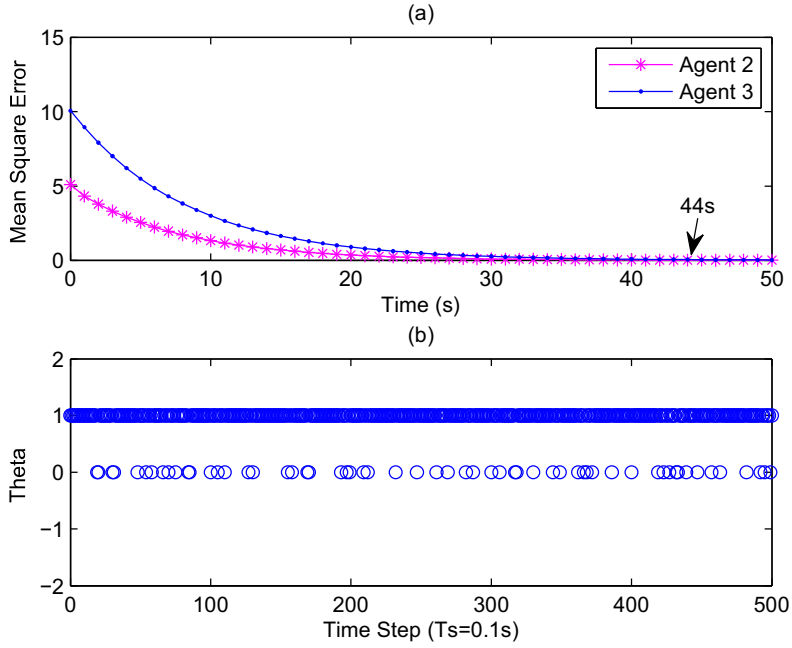


Figure 5.6: Mean Square Error and Loss Rate Distribution with r=10%

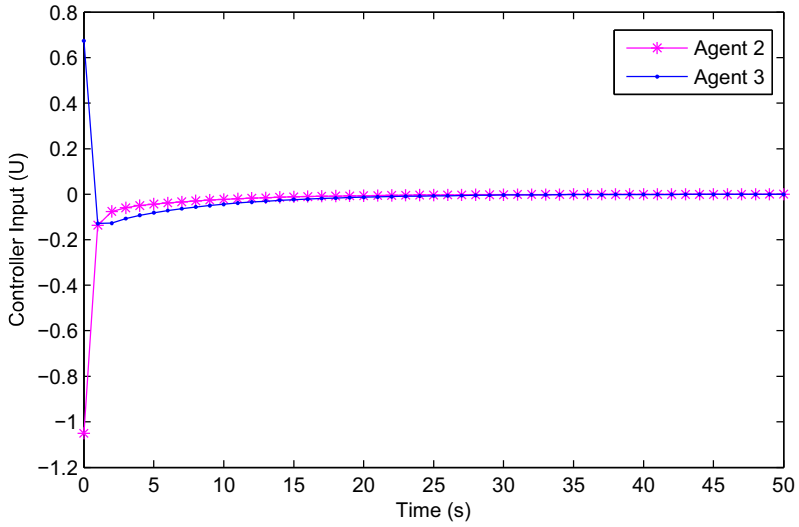


Figure 5.7: Controller Input (u) with r=10%

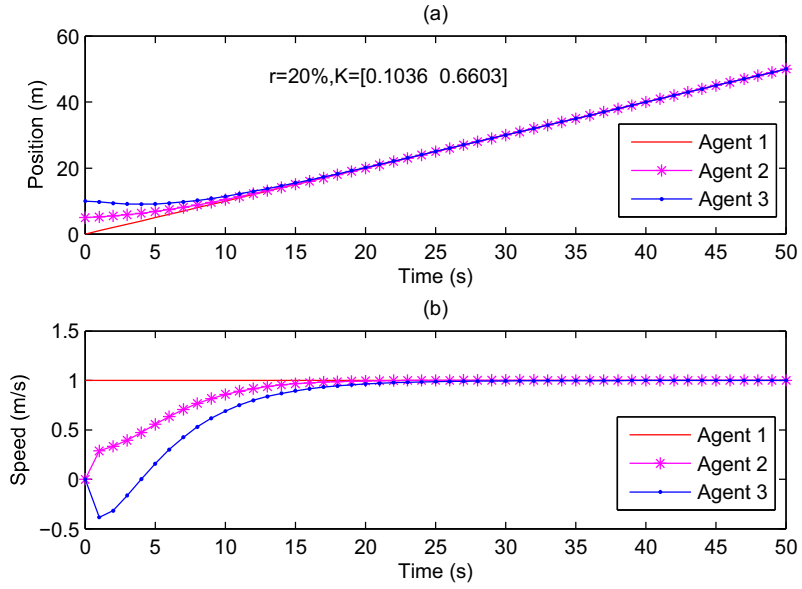


Figure 5.8: Simulation Results of Position and Speed in One Dimensional Plane with r=20%

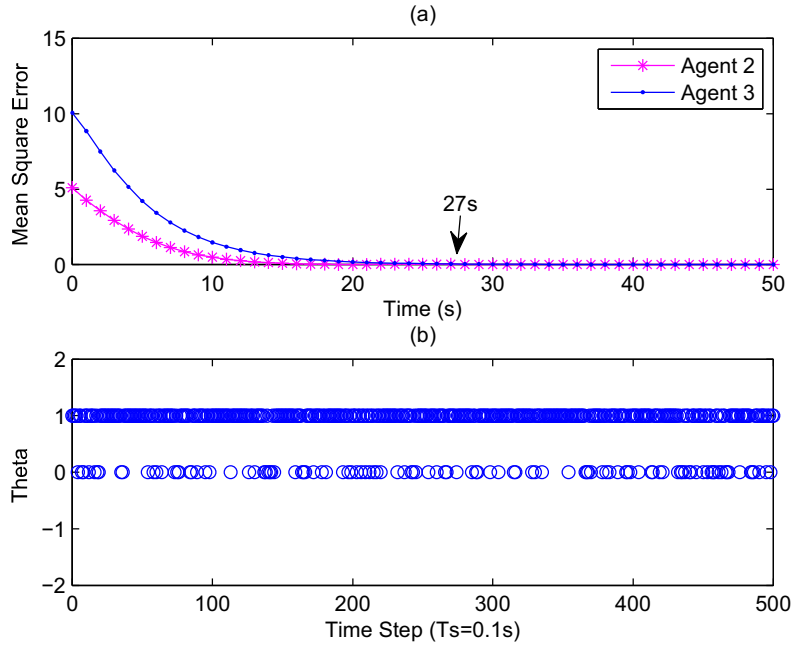


Figure 5.9: Mean Square Error and Loss Rate Distribution with r=20%

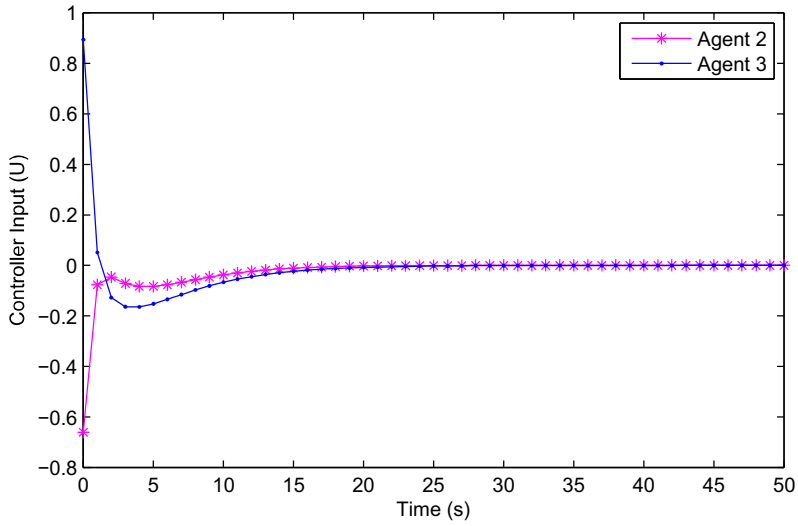


Figure 5.10: Controller Input (u) with r=20%

seconds. Therefore, with stronger effort of the controller, at 27 second, the system achieves consensus. It is faster than Example 2 by 17 seconds.

Example 4: r=30%

With increase of the loss rate, the controller gain is obtained as:

$$K = [0.0877\ 0.4223]. \tag{5.7}$$

It is easy to see that with higher loss rate, the controller gain is decreasing. At Example 1, when the loss rate is 5%, the K is [0.2713 2.0399]. Because a higher loss rate raises the uncertainty of the system, the controller gain has to be reduced in order to reach consensus.

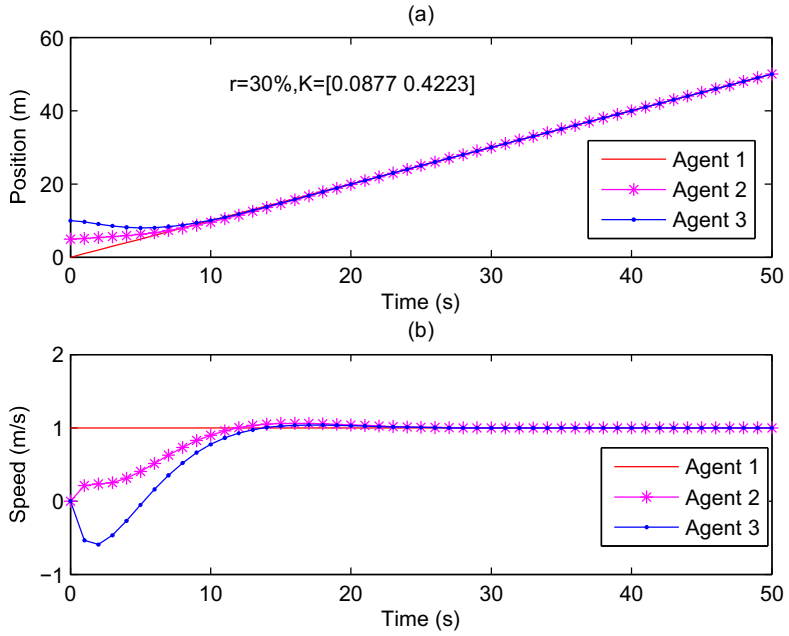


Figure 5.11: Simulation Results of Position and Speed in One Dimensional Plane with r=30%

Fig. 5.11 shows the simulation results after the loss rate becomes 30%. Fig. 5.12 shows the mean square error and data loss distribution. Since each gain has a different controller input, Fig. 5.13 shows the controller with gain of [0.0877 0.4223]. From the first 10 seconds, the slop of curve is very steep, and without any noticeable overshoot after. It means the correcting effort is much stronger than previous examples.

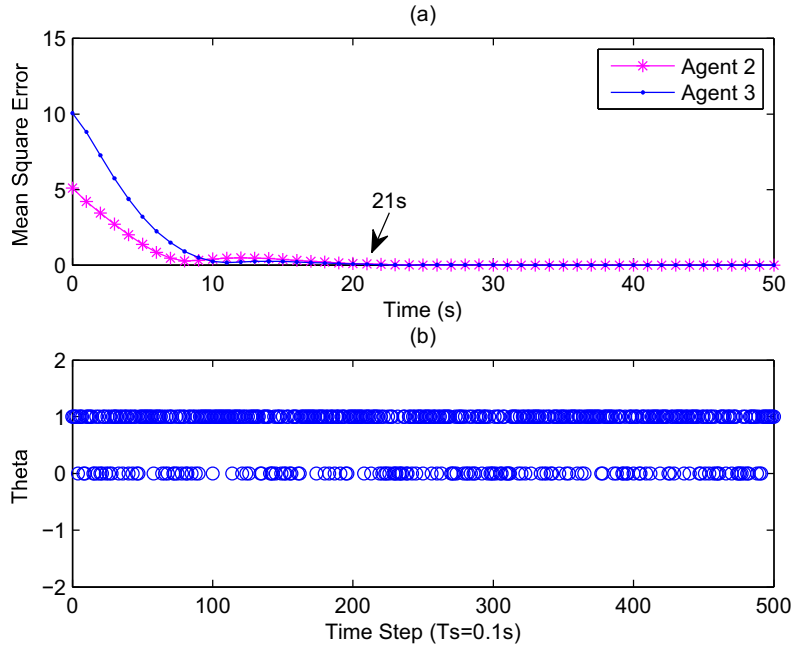


Figure 5.12: Mean Square Error and Loss Rate Distribution with r=30%

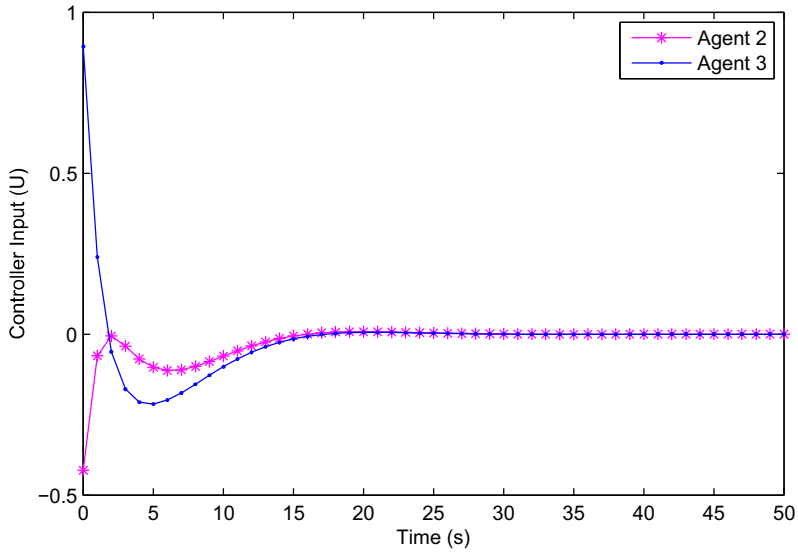


Figure 5.13: Controller Input (u) with r=30%

Therefore, with exact same other conditions, the system reaches consensus in just 21 seconds.

Example 5: r=40%

The gain becomes $K = [0.0673\ 0.2728]$, once the rate loss increases to 40%. The simulation results are shown in Fig. 5.14, and the mean square error with loss rate distribution are shown in Fig. 5.15. Fig. 5.16 shows the controller input on Agent 2 and 3. The curve of this figure shows there is a overshoot after 15 second. However it barely can see a similar overshoot in Example 4. Thus the consensus time is 33 seconds which is longer than 21 seconds in Example 4.

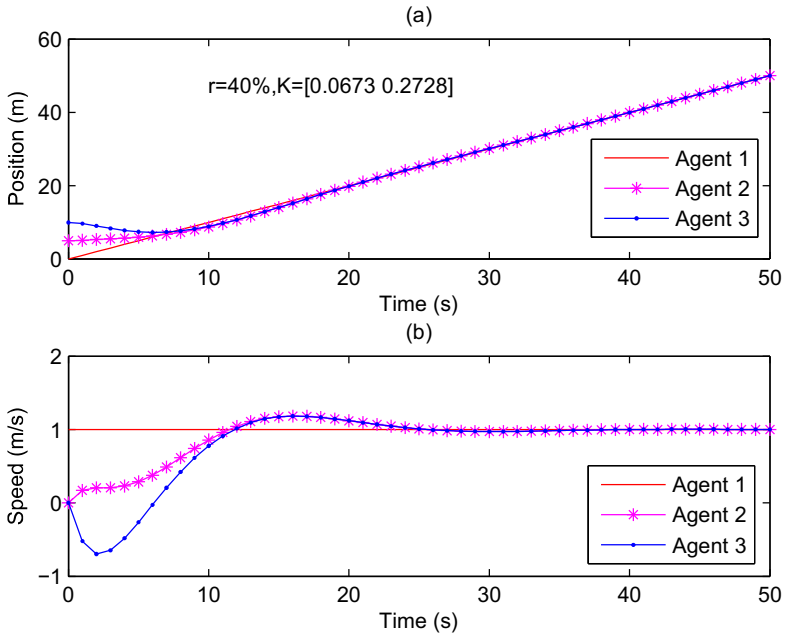


Figure 5.14: Simulation Results of Position and Speed in One Dimensional Plane with r=40%

Example 6: r=50%

With the data loss rate bound set as 0.5, the controller gain is

$$K = [0.0050\ 0.0494]. \tag{5.8}$$

In previous examples, the time range is 50 seconds. However, the time range in this example is 70 seconds due to the higher loss rate. From both Fig. 5.17 and Fig. 5.18,

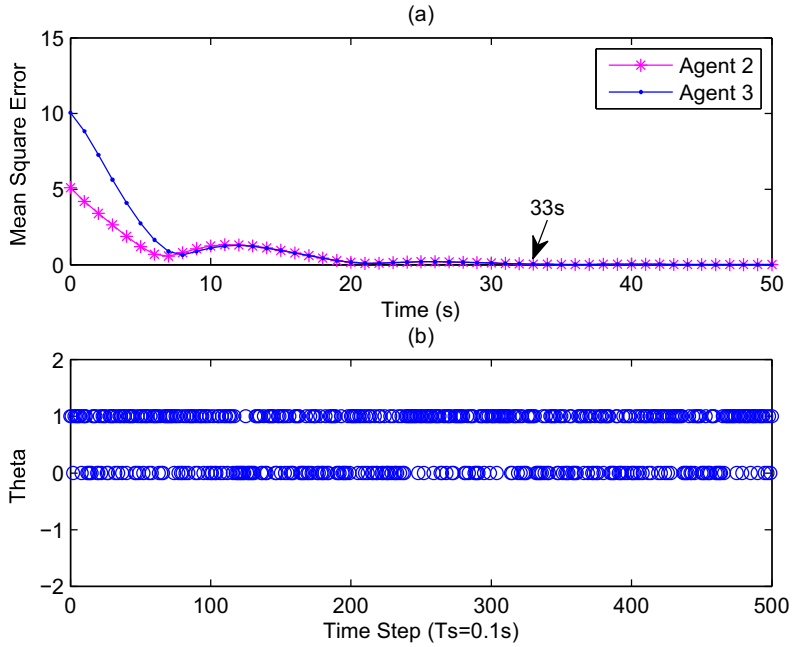


Figure 5.15: Mean Square Error and Loss Rate Distribution with r=40%

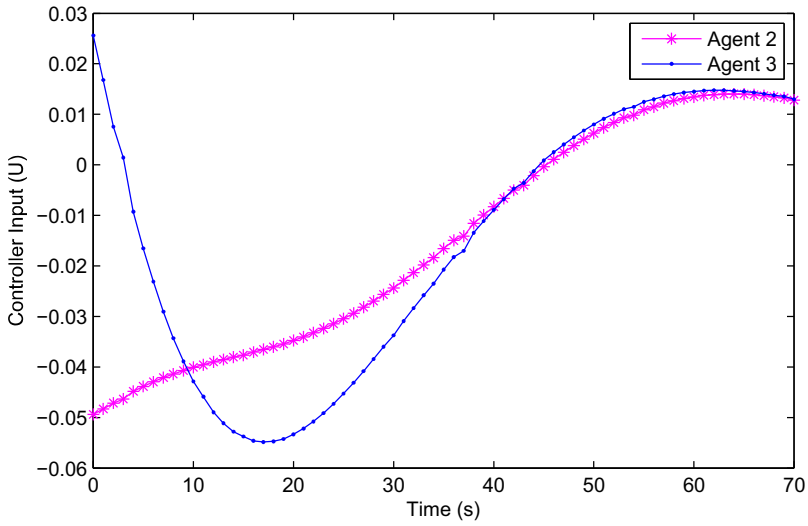


Figure 5.16: Controller Input (u) with r=40%

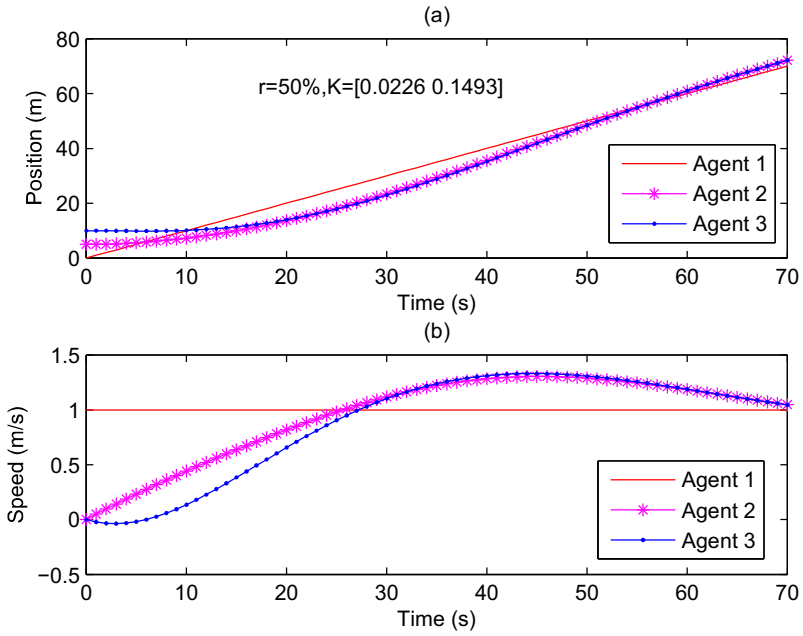


Figure 5.17: Simulation Results of Position and Speed in One Dimensional Plane with r=50%

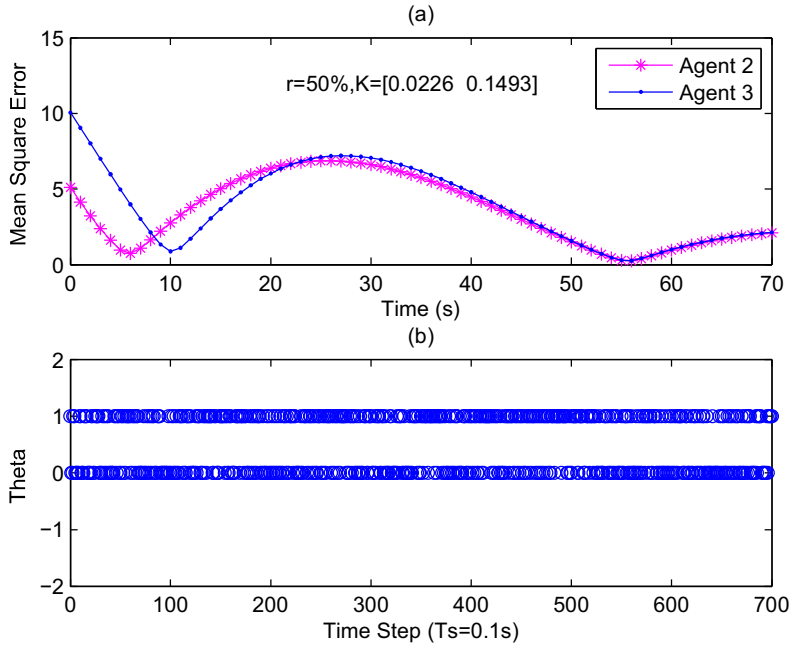


Figure 5.18: Mean Square Error and Loss Rate Distribution with r=50%

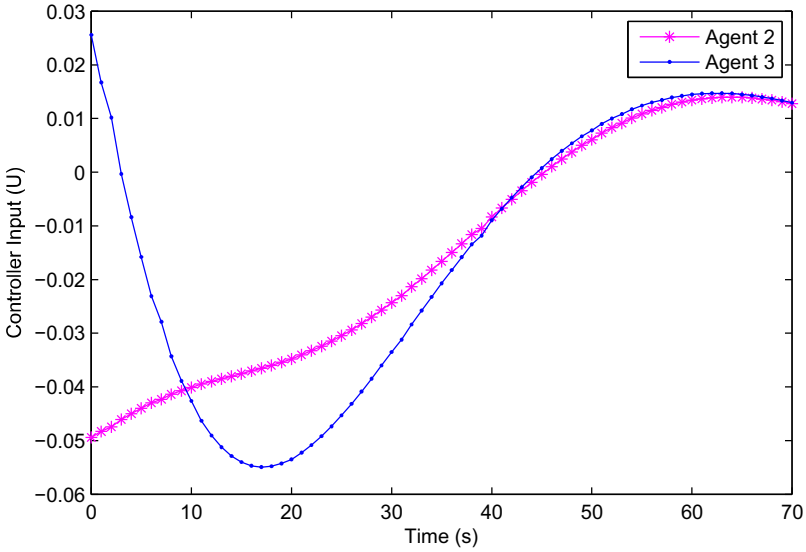


Figure 5.19: Controller Input (u) with r=50%

even with an expanded time range, the system is still not consented at 70 seconds. In Fig. 5.18(a), both of Agent 2 and 3 are convergent at 5 and 10 second, and then divergent after those points. At 55 seconds, the error reaches to zero again. It is like a ball bounces a few time before it can be settled down. Because the higher loss rate increases the uncertainty of the system, as a result, the controller gain decreases. A weaker controller causes a larger oscillation while controlling the system. Therefore the consensus time is extended. It happens once the loss rate reaches 40%. Fig. 5.19 shows the controller input. Since the controller gain is very small compared with previous examples, at 70 seconds, the controller input is still correcting the system in order to reach consensus.

Example 7: r=90%

After studying examples within 50% in loss rate, in order to analyze the effectiveness of designed controller, the data loss rate reaches 90%. The controller gain is $K = [0.0002\ 0.0080]$, which is extremely small. The time range maintains as 70 seconds, in Fig. 5.20, Agent 2 and 3 far beyond to reach consensus with Agent 1 within this time frame. It also can be seen from Fig. 5.21(a), the mean square errors of Agent 2 and 3 converges in the first 5 and 10 seconds respectively. However both of errors bounces back to higher values of error. At 70 seconds, the error reaches 40, which is just like the phenomena happens in Example 6. Fig. 5.21(b) shows 10% of θ equals

one, which means that only 10% of packet can be successfully received by Agent 2 and 3. From Fig. 5.21, the controller inputs for both agents are very limited. They start at -0.005 and -0.008 respectively, and slowly increase to -0.0105 and -0.011 at 70 second. In order to study more in the case of higher loss rate, the time range is expanded to 1000 seconds instead of 70 seconds.

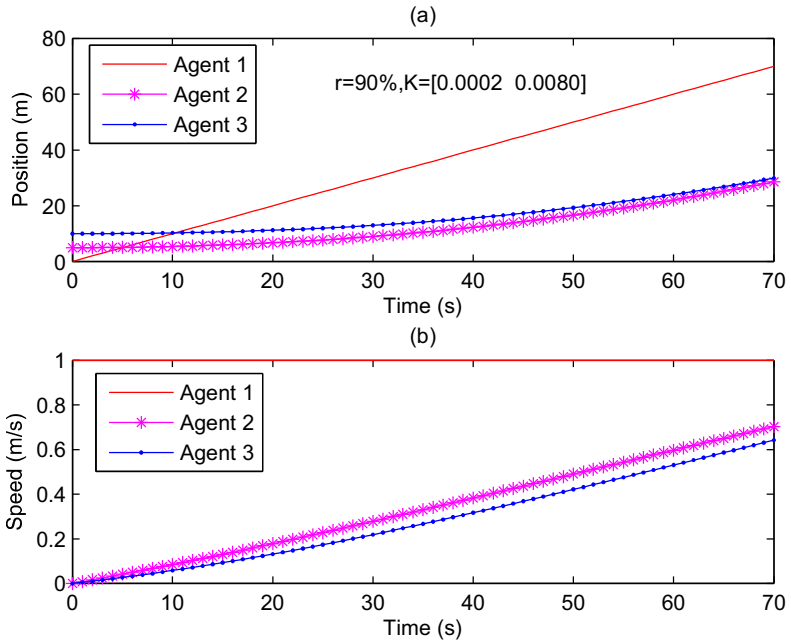


Figure 5.20: Simulation Results of Position and Speed in One Dimensional Plane with r=90%

Fig. 5.23 shows the result of the system within a 1000 seconds of time frame. The speed of Agent 2 and 3 reaches 1 m/s while the time approaches to 1000 second. In Fig. 5.24, it is clear to see that the mean error of the system bounces six time within 1000 seconds. at 1000 second, the error is about 2.

Example 8: r=96%

Now, in order to determine the maximum data loss rate bound, the data loss rate reaches 96%. The controller gain is $K = [-0.0001\ 0.0037]$, which is extremely small. Fig. 5.25 and Fig. 5.26 show the system cannot reach consensus anymore. Therefore, the maximum data loss rate of the proposed controller is about 95%.

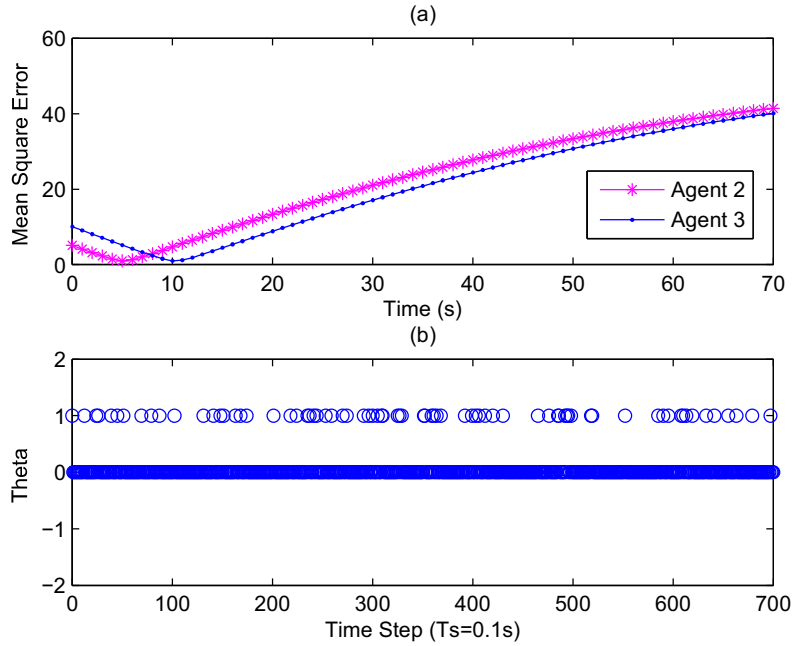


Figure 5.21: Mean Square Error and Loss Rate Distribution with r=90%

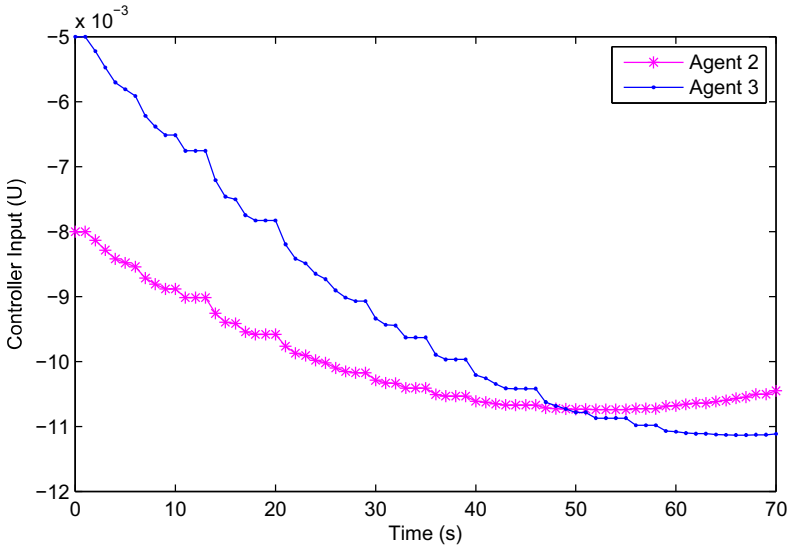


Figure 5.22: Controller Input (u) with r=90%

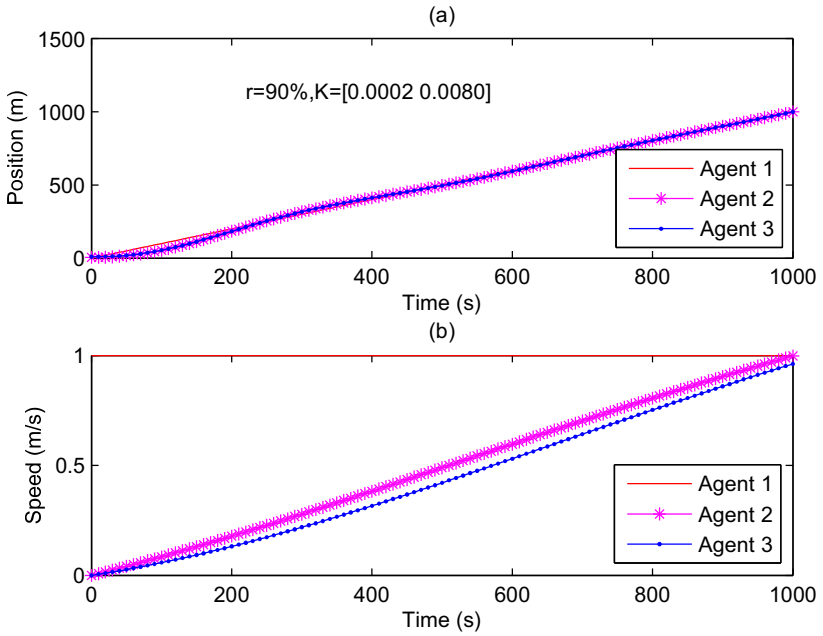


Figure 5.23: Simulation Results of Position and Speed in One Dimensional Plane with r=90% in 1000 Seconds

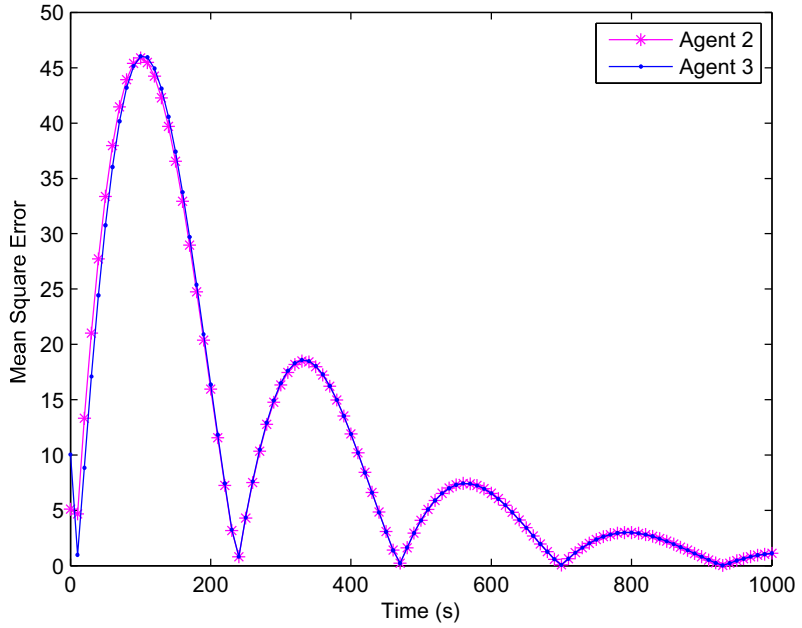


Figure 5.24: Mean Square Error with r=90% in 1000 Seconds

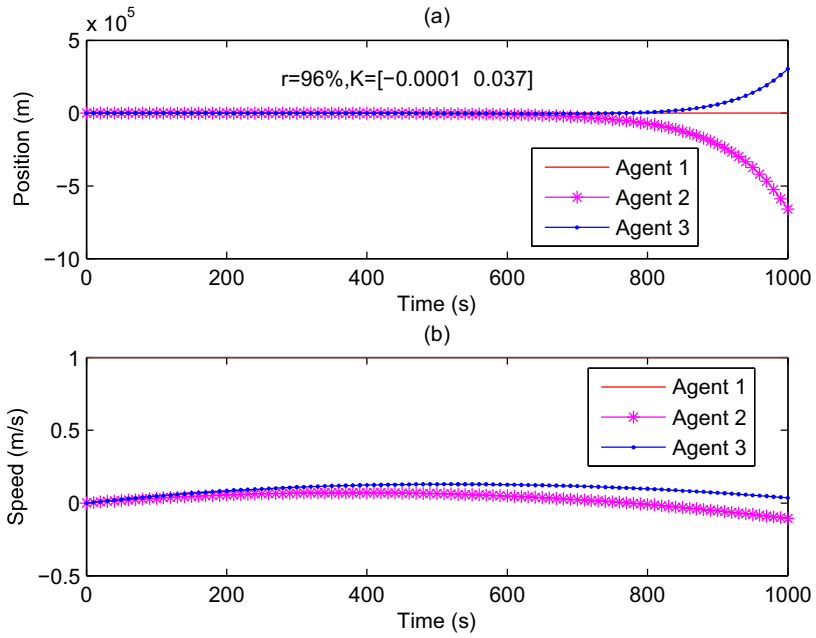


Figure 5.25: Simulation Results of Position and Speed in One Dimensional Plane with r=96%

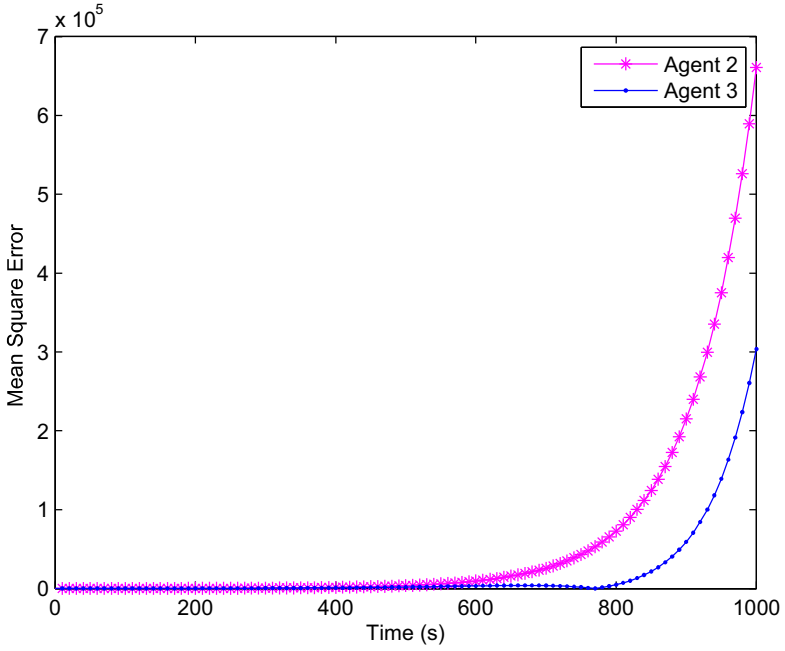


Figure 5.26: Mean Square Error with r=96%

Table 5.1: Effects of data loss rates (r)

Example	Loss Rate(r%)	Controller Gain(K)	Consensus Time(s)
(1)	5%	[0.2713 2.0399]	38
(2)	10%	[0.1150 1.0506]	44
(3)	20%	[0.1036 0.6603]	27
(4)	30%	[0.0877 0.4223]	21
(5)	40%	[0.0673 0.2728]	33
(6)	50%	[0.0226 0.1493]	> 70
(7)	90%	[0.0002 0.0080]	> 1000
(8)	96%	[-0.0001 0.0037]	∞

5.1.3 Summary on the Effect of Data Loss Rate

In summary, the simulation results for a group of three agent system are explored and discussed with different data loss rates. Table 5.1 shows all results of examples with corresponding time. Each gain obtained from LMI is only satisfied for a sufficient condition, it means this gain might not the optimal gain but it does guarantee the system can be reached consensus at corresponding data loss rate. For example, with 20% of data loss rate, a gain K = [0.0877 0.4223] ensures the system to reach the consensus. Consensus time in Table 5.1 only shows that whether the controller drive the system to reach consensus or not. The maximum data loss rate of effective controller is 95%, because at 96%, the controller is not able to converge the system anymore. Based on the result, the proposed controller is more efficient within 50% of loss rate.

5.2 Effect of Communication Weights

In Section 5.1, the communication weight ω_{ij} was assumed to be 1 for all channels. In this section, the effect of weight on consensus time is studied.

5.2.1 Simulation Configuration

As same as Section 4.1, the topology of multi-agent system was modeled by a directed graph with three vertices, as shown Fig. 5.27. The sampling time is still 0.1 second which is the same with section 4.1, and the data loss rate is $r = 20\%$ for all examples. In order to analyze the effect of weight, the communication link between Agent 2 and 3 is removed. Therefore Agent 2 and 3 can independently and exclusively receive information from Agent 1. Refer to Chapter 2, the $\omega_{ij} \in (0, 1)$ is a proportional weight, which is to increase the information flow rate. Each communication channel

is like a pipe, and communication weight is the valve installed in each pipeline. If the weight equals one, it means the valve is fully open, otherwise it is proportionally closed with the weight decreases, until it is fully closed with weight becomes zero. Four examples are studied on effect of communication weight of consensus ability and time. The first example, set ω_{21}=ω_{31}=1, which is the standard example with both channels are fully open. The second example, set $\omega_{21} = 1$, while $\omega_{31} = 0.1$, which is ten times smaller than Example 1. Therefore, the effect of communication weight on the channel between Agent 1 and Agent 3 can be summarized while comparing with Example 1. The third example, set $\omega_{21} = 0.1$ instead, and maintain $\omega_{31} = 1$. Therefore the result from Example 2 can be verified in the channel between Agent 1 and Agent 2. The last example, set $\omega_{21} = 0.5$ and $\omega_{31} = 1$. The purpose of Example 4 is to explore if the consensus ability or time is proportional to the weight ω_{ij}.

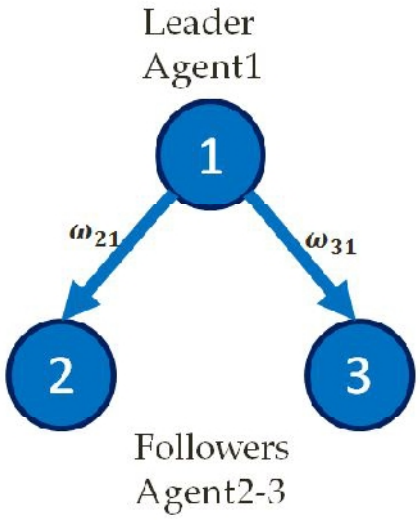


Figure 5.27: A Group of Three Agents in A Directed Graph Topology with Communication Weight

Because the weight on each channel is changed, the adjacency matrix A_a and Laplacian matrix L in this section are defined again based on Eq.(2.1) and Eq.(2.2). Therefore, each example needs to be obtained a new matrix A_a and L based on Eq.(6.8.

$$A_a = \begin{bmatrix} 0 & 0 & 0 \\ \omega_{21}1 & 0 & 0 \\ \omega_{31}1 & 0 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ -\omega_{21}1 & \omega_{21}1 & 0 \\ -\omega_{31}1 & 0 & \omega_{31}1 \end{bmatrix}. \tag{5.9}$$

5.2.2 Simulation Results

Example 1: ω_{21}=1, ω_{31}=1

In this example, ω_{21}=$\omega_{31} = 1$. As a result, matrix A_a and L from Eq.(6.8 become

$$A_a = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \tag{5.10}$$

The eigenvalues of L are $0, 1, 1$,respectively, which satisfies the condition of controllability in Chapter 2.

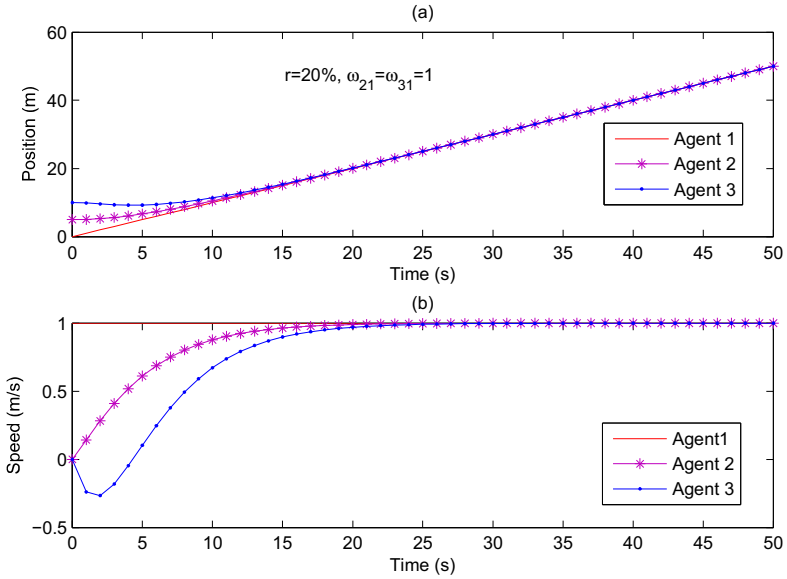


Figure 5.28: Simulation Results of Position and Speed in One Dimensional Plane with $\omega_{21} = \omega_{31} = 1$, r=20%

Since the data loss rate is set as 20% for all examples in this section, the data loss distribution is not shown anymore. The figure of controller input is also not shown again because controller input is only changed with the controller gain, and the value of K is based on the loss rate. Fig. 5.25 shows the results of simulation while the data loss rate is 20%, and $\omega_{21} = \omega_{31} = 1$. From Fig. 5.29, the consensus time for Agent 2 and Agent 3 is in 20 and 24 second, respectively.

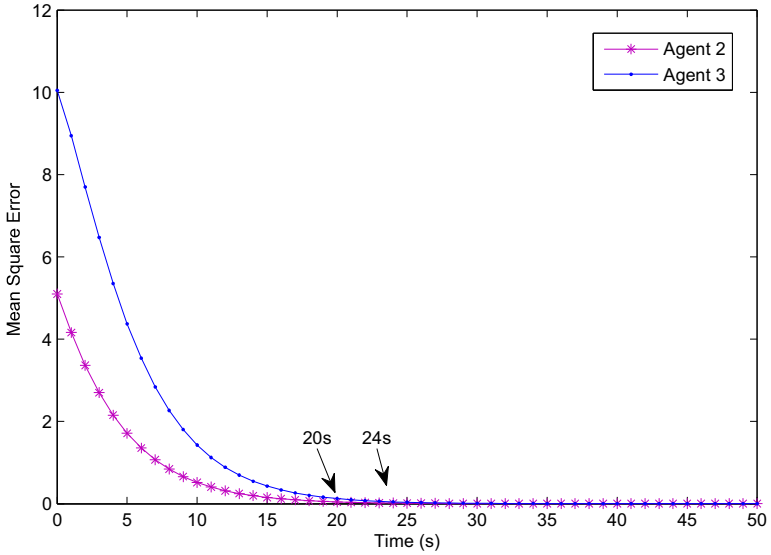


Figure 5.29: Mean Square Error with $\omega_{21} = \omega_{31} = 1$, r=20%

Example 2: ω_{21}=1, ω_{31}=0.1

In the second example, set ω_{21} bigger than ω_{31} with $\omega_{21} = 1$ and $\omega_{31} = 0.1$. Agent 2 can receive information from Agent 1 with 100% flow rate, while Agent 3 can only receive with a flow rate of 10% compared with Example 1. Since the weigh is changed in the system, a new matrix A_a and L in this example become based onEq.(6.8)

$$A_a = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0.1 & 0 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -0.1 & 0 & 0.1 \end{bmatrix}. \tag{5.11}$$

The eigenvalues of L are this example are $0, 0.1, 1$. Within a zero in the eigenvalue, the system can be consented.

Fig. 5.31 shows Agent 2 can converge with Agent 1 in 20 seconds. It is the same with Example 1 since the ω_{21} maintains one. However, the consensus time for Agent 3 is over 50 seconds. For this example, it verified the communication weight is to increase the information flow rate in the channel between Agent 1 and 3. In order to determine the influence of the weight further, another example is studied with $\omega_{21} = 0.1; \omega_{31} = 1$.

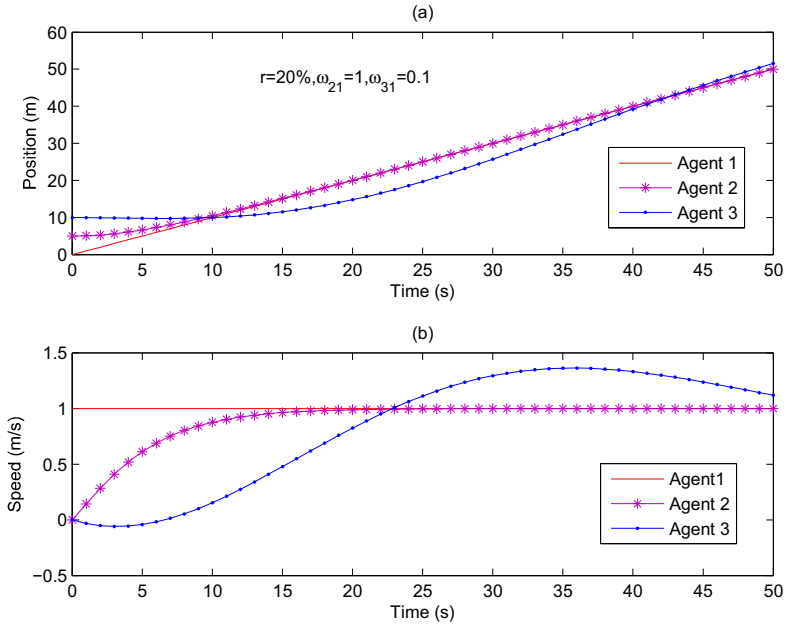


Figure 5.30: Simulation Results of Position and Speed in One Dimensional Plane with $\omega_{21} = 1; \omega_{31} = 0.1$, r=20%

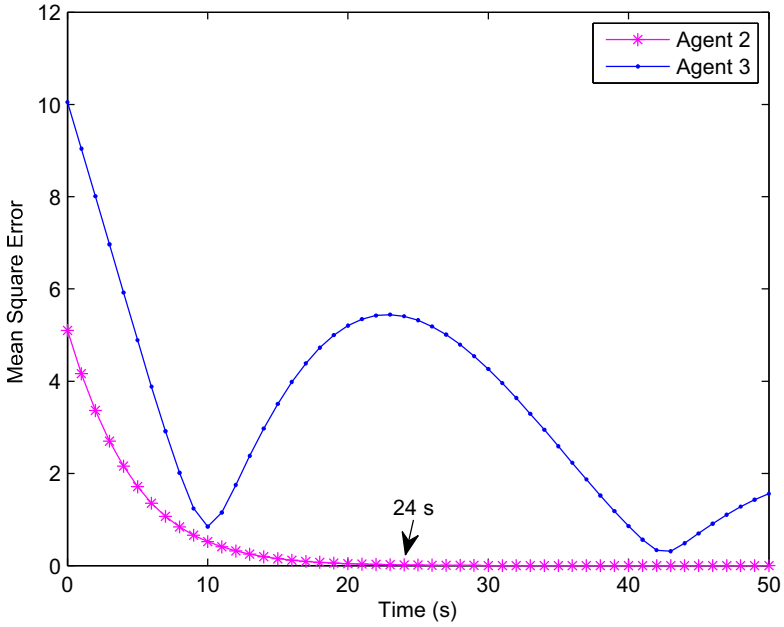


Figure 5.31: Mean Square Error with $\omega_{21} = 1; \omega_{31} = 0.1$, r=20%

Example 3: ω_{21}**=0.1,** ω_{31}**=1**

In this time, if Agent 3 receives information from Agent 1 at 100%, while Agent 2 only receives at 10%. Thus $\omega_{21} = 0.1$ and $\omega_{31} = 1$ and the adjacency matrix A_a and Laplacian matrix L become

$$A_a = \begin{bmatrix} 0 & 0 & 0 \\ 0.1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ -0.1 & 0.1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{5.12}$$

The eigenvalues of L are $0, 0.1, 1$, which satisfies the condition to be consented.

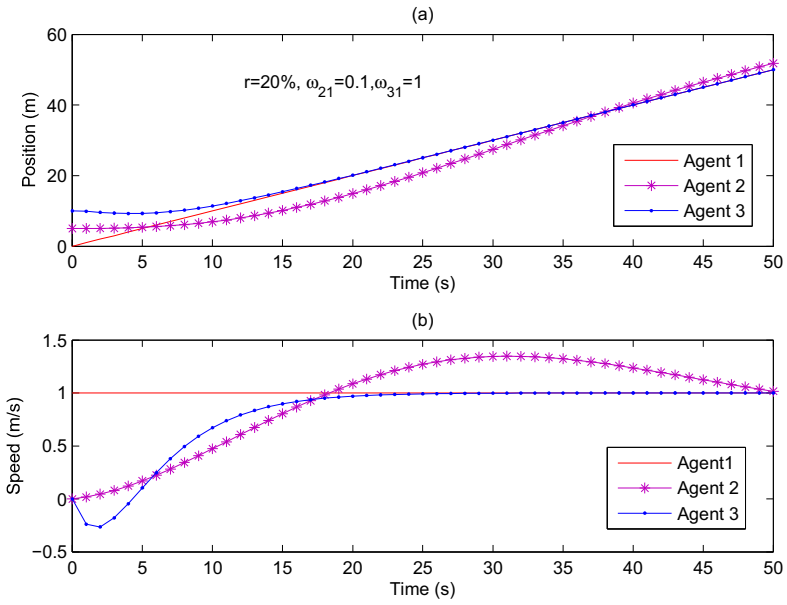


Figure 5.32: Simulation Results of Position and Speed in One Dimensional Plane with $\omega_{21} = 0.1; \omega_{31} = 1$, r=20%

Fig. 5.33 shows that Agent 3 in this example converges with Agent 1 in 24 seconds. it is increased slightly based on Example For Agent 2, the consensus time is definitely over 50 seconds. At this point, we can say that Example 1, 2, and 3 verified the communication weight is to increase the information flow rate in the weighed channel.

Example 4: ω_{21}**=0.5,** ω_{31}**=1**

From pervious examples, they approve that the that the communication weight is a information flow distribution measurement. The higher weight gives a higher information flow rate. However, it is reasonable to ask if there a proportional relationship

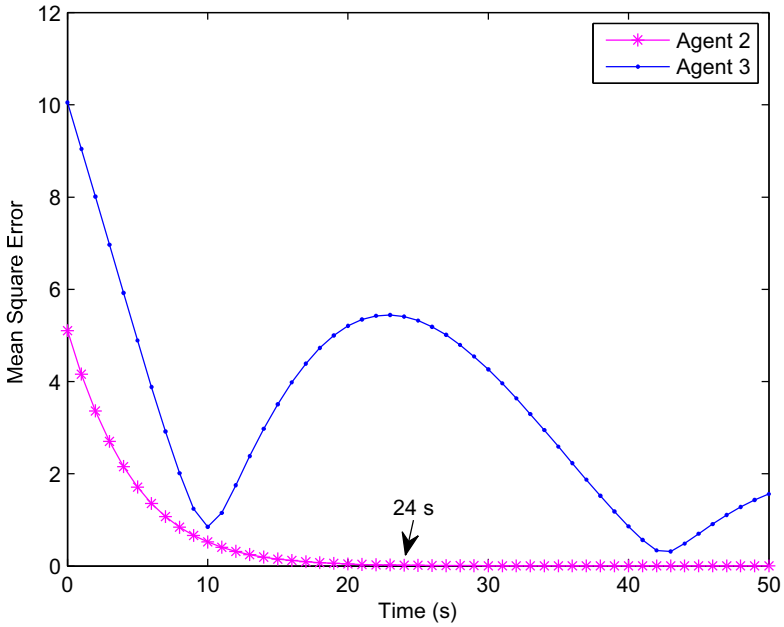


Figure 5.33: Mean Square Error of with $\omega_{21} = 0.1; \omega_{31} = 1$, r=20%

between the weight and the consensus time. For example, increase the ω_{21} by double, does it mean that the consensus time for Agent 2 is reduced by half. Let set ω_{21}=0.5, and ω_{31}=1. Compared to Example 1, the weight between Agent 1 and 2 is decreased to 0.5 from 1, and while the weight between Agent 1 and 3 maintains the same. Since the ω_{31} is the same to Example 1. Therefore the consensus of Agent 3 is expected to be 24 seconds. In Fig. 5.35, it is approved that the consensus time of Agent 3 is 24.

5.2.3 Summary on the Effect of Communication weight

Table 5.2: Effects of Communication Weight (ω_{ij}) with $r = 20\%$

Example	Weight(ω_{21})	Weight(ω_{31})	Consensus Time of Agent2(s)	Consensus Time of Agent3(s)
(1)	1	1	20	24
(2)	1	0.1	20	> 50
(3)	0.1	1	> 50	24
(4)	0.5	1	25	24

In summary, the simulation results are explored and discussed for four examples with different communication weights. Table 5.2 shows all results of consensus time of Agent 2 and 3, respectively. The simulation results show the weight is a proportional

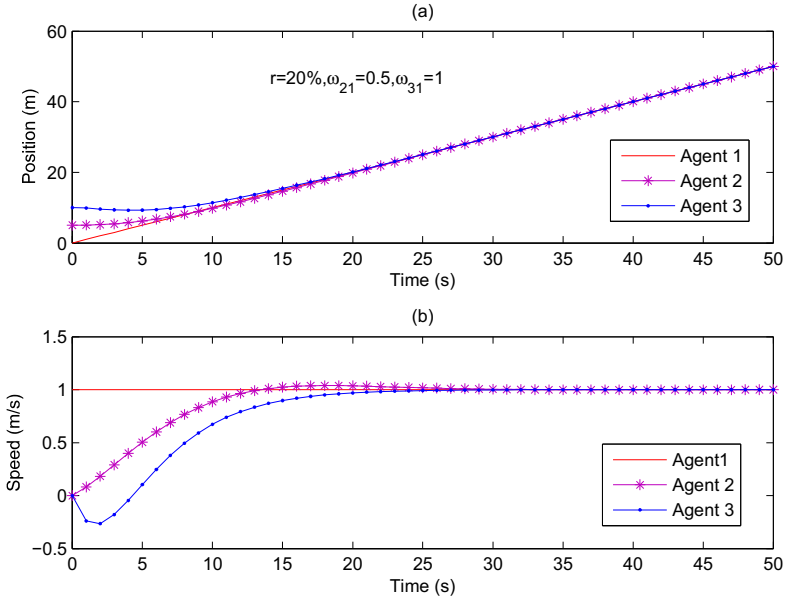


Figure 5.34: Simulation Results of Position and Speed in One Dimensional Plane with $\omega_{21} = 0.5; \omega_{31} = 1$, r=20%

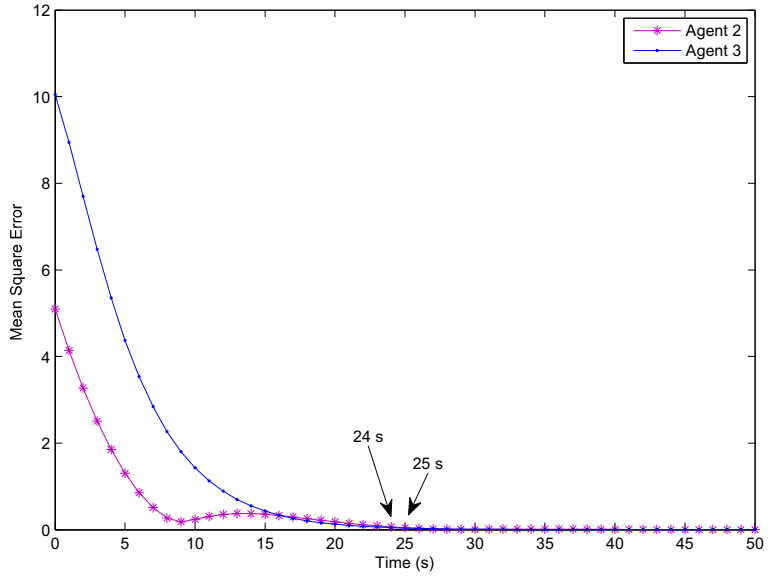


Figure 5.35: Mean Square Error with $\omega_{21} = 0.5; \omega_{31} = 1$, r=20%

gain of distributing the information flow rate among the system. The higher weight distribution can shorten the consensus time until the weight reaches to 1. In Example 4, ω_{21}=1 by decreasing half from Example 2. However the consensus time is not proportionally increase in scale.

5.3 Effect of Initial Values

In Section 4.1 and 4.2, all previous examples have the same initial values, such as Agent 2 is 5 and Agent 3 is 10. In order to prove the effectiveness of the proposed controller, effect of initial value for consensus time are studied in this section with four examples. $x_{11}(k)$ means the first state information value (position) of Agent 1. $x_{11}(1)$ is the initial value in the term of the position for Agent 1. $x_{21}(1)$ means the initial vale of Agent 2. Each example has a unique initial value ($x_{11}(1)$,$x_{21}(1)$). For example, $(5, 10)$ means the initial value of Agent 2 is 5, and 10 for Agent 3. Moreover, $(10, 20)$, $(20, 40)$, $(80, 160)$ are studied as well to determine the effect of initial value on consensus ability.

5.3.1 Simulation Configuration

The configuration is the exactly the same in Section 4.2. Three agent system with all weight equaling one, data loss rate in all examples is 20%, and sampling time is 0.1 second. Therefore, the adjacency matrix A_a and Laplacian matrix L are unaffected.

5.3.2 Simulation Results

Example 1: $x_{11}(0)$**=5,** $x_{21}(0)$**=10**

The initial values of Agent 2 and 3 are 5 and 10, respectively.
Fig. 5.36 and Fig. 5.37 show that consensus time of the system is 27 seconds.

Example 2: $x_{11}(0)$**=10,** $x_{21}(0)$**=20**

In this example, the initial values of Agent 2 and 3 are increased to 10 and 20,respectively.
Fig. 5.38 and Fig. 5.41 show that consensus time of the system is 30 s.

Example 3: $x_{11}(0)$**=20,** $x_{21}(0)$**=40**

The initial values of agent 2 and 3 are increased to 20 and 40 respectively.

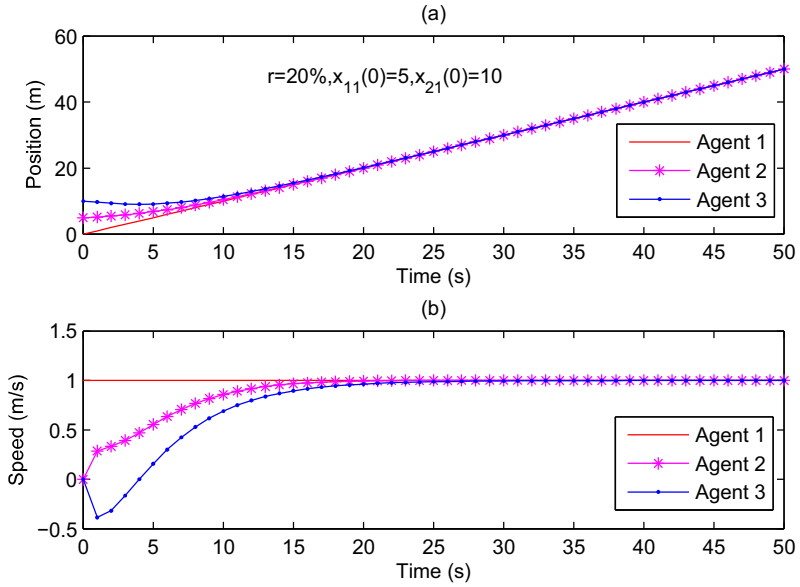


Figure 5.36: Simulation Results of Position and Speed in One Dimensional Plane with $x_{11}(0)$=5, $x_{21}(0)$=10, r=20%

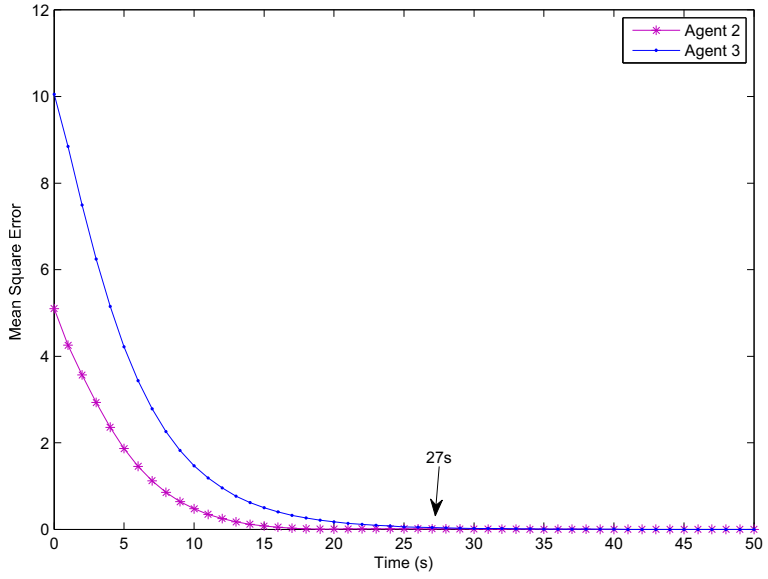


Figure 5.37: Mean Square Error with $x_{11}(0)$=5, $x_{21}(0)$=10, r=20%

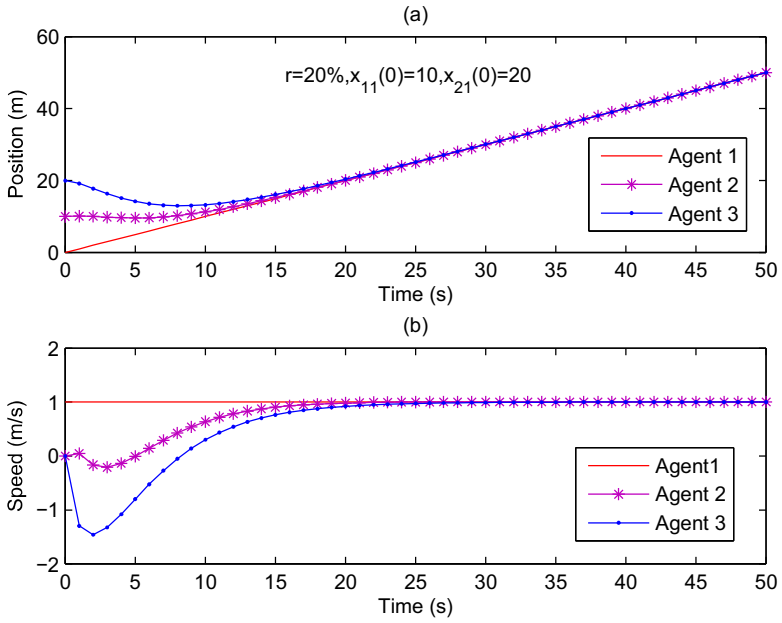


Figure 5.38: Simulation Results of Position and Speed in One Dimensional Plane with $x_{11}(0)$=10, $x_{21}(0)$=20, r=20%

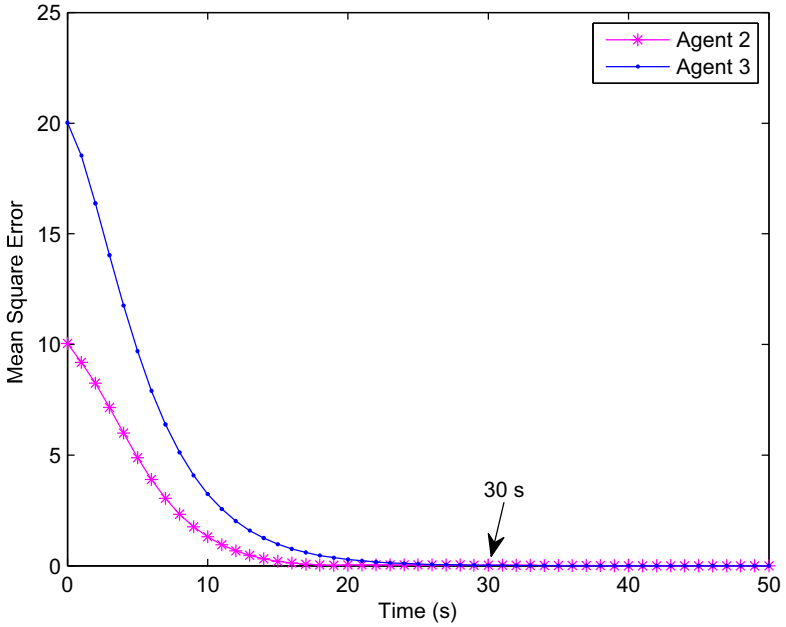


Figure 5.39: Mean Square Error with $x_{11}(0)$=10, $x_{21}(0)$=20, r=20%

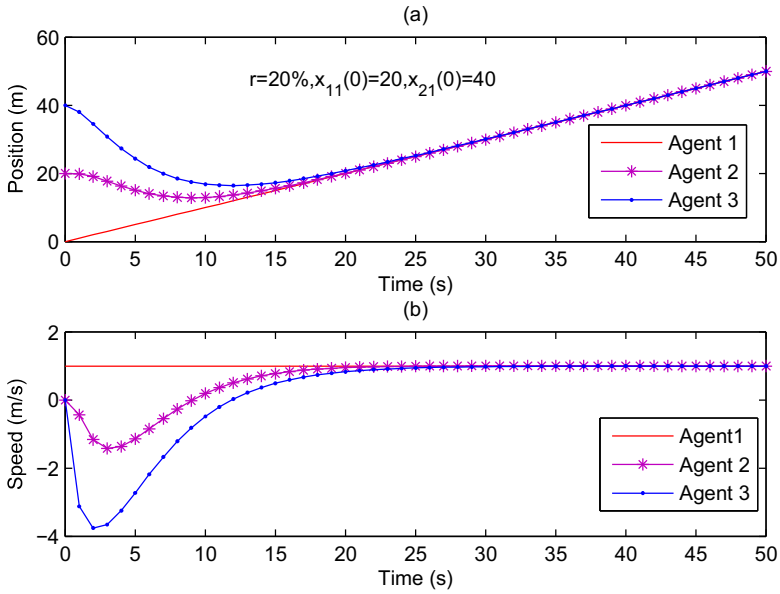


Figure 5.40: Simulation Results of Position and Speed in One Dimensional Plane with $x_{11}(0)$=20, $x_{21}(0)$=40, r=20%

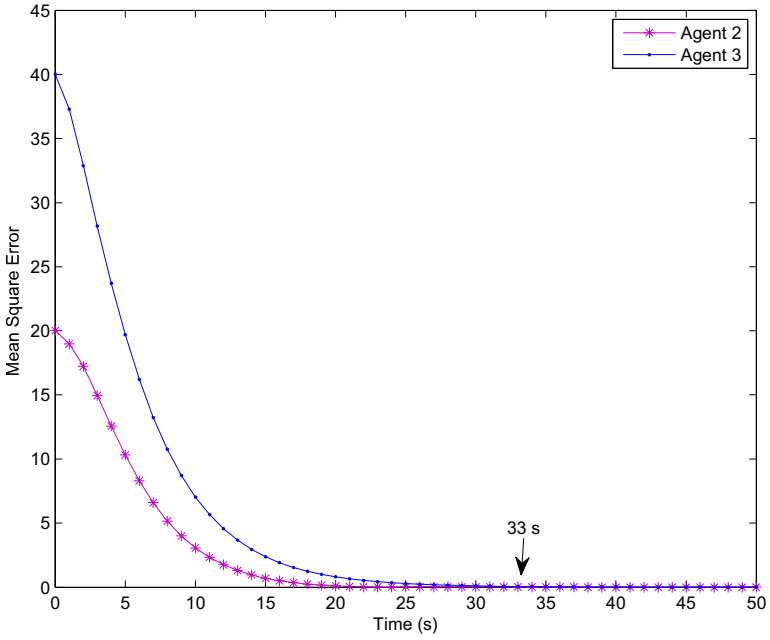


Figure 5.41: Mean Square Error with $x_{11}(0)$=20, $x_{21}(0)$=40, r=20%

Fig. 5.40 and Fig. 5.41 show that consensus time of the system is 34 seconds. From Example 1 to Example 3, the initial values are increased fourfold, however the consensus time is only changed from 27s to 34s with 7 increment. From Fig. 5.40, Agent 3, 40 meters away from Agent 1, accelerates to nearly 4 m/s from 0 in just 2 seconds. Because the controller proportionally increases the acceleration of agents while the initial value becomes bigger, the initial values does not significantly affect the consensus time. In order to prove this conclusion, another example with initial value of (80,160) is studied.

Example 4 $x_{11}(0)$**=80,** $x_{21}(0)$**=160**

In the last example, the initial values of Agent 2 and 3 are increased to 80 and 160 respectively.

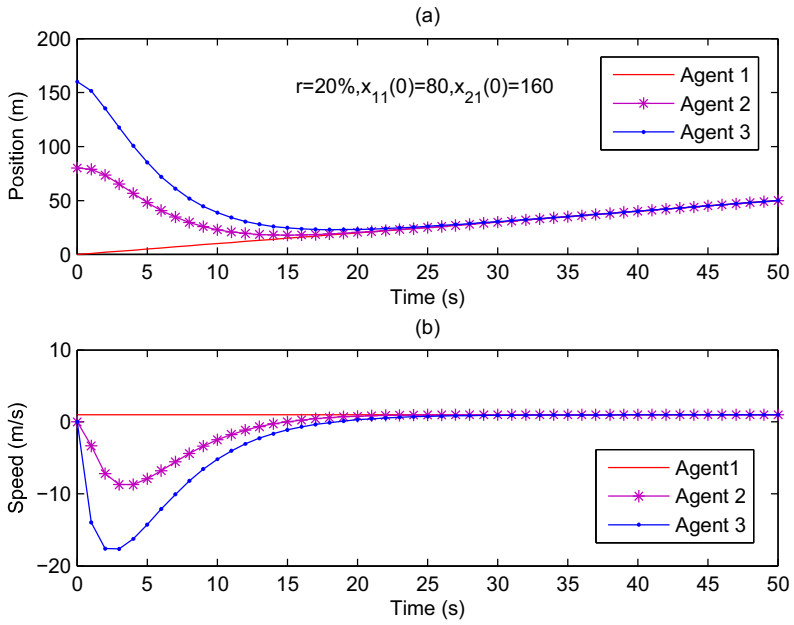


Figure 5.42: Simulation Results of Position and Speed in One Dimensional Plane with $x_{11}(0)$=80, $x_{21}(0)$=160, r=20%

Fig. 5.42 and Fig. 5.43 show that consensus time of the system is 41 seconds. Therefore, it approved again the initial value does not significantly effect the consensus time, because the acceleration is also increased correspondingly.

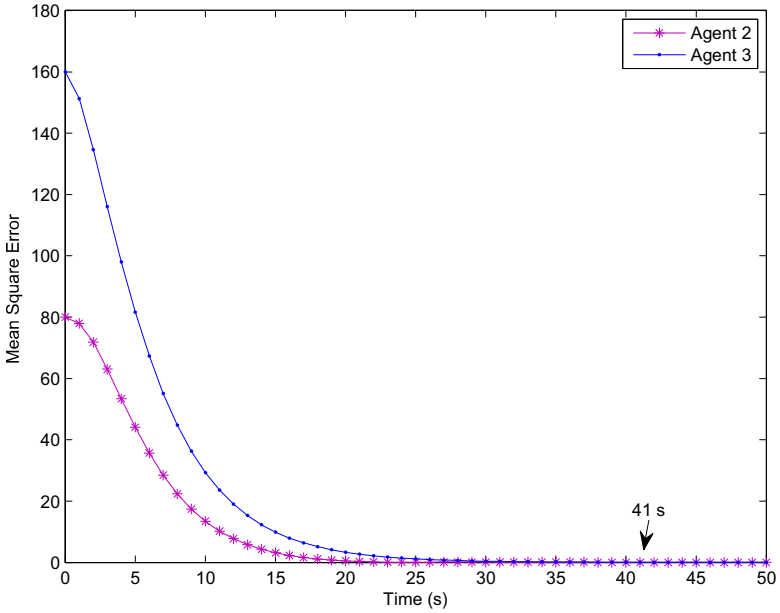


Figure 5.43: Mean Square Error with $x_{11}(0)$=80, $x_{21}(0)$=160, r=20%

5.3.3 Summary on the Effect of Initial Value

Table 5.3: Effects of Initial Value ($x_i(0)$) with $r = 20\%$

Example	No. of Agents	$x_{11}(0)$,$x_{21}(0)$(m)	Consensus Time(s)
(1)	3	5,10	27
(2)	3	10,20	30
(3)	3	20,40	33
(4)	3	80,160	41

In summary, the simulation results are explored and discussed for four examples with different initial values. Table 5.3 shows all results of consensus time of the system with corresponding initial values. The simulation results show the higher initial value increases the consensus time. However, the initial value from $(5, 10)$ to $(80, 160)$ only increases the consensus time from 27 to 41 seconds. The initial value is risen by 16 times, the consensus time is only increased by less than 2 times. Therefore, the initial value does not significantly affect the consensus time.

5.4 Effect of Sampling Time

Sampling time is the time between successive measurements of a physical quantity. [53] discussed the effect of sampling time for a PID controller. It concludes that if the sampling time is too big, it has a negative impact on controller performance. However, sampling faster does not necessarily provide better performance. In this section, five examples are studied with the sampling time (Ts) equaling 0.01,0.1,0.5,1, and 10 seconds in order to determine the effect of the sampling time and explore the most optimal sampling time for the proposed controller.

5.4.1 Simulation Configuration

The configuration is exactly the same as in Section 4.3. Three agent system with all weights equal to one; data loss rates in all examples are 20%; and initial values are 5 and 10 for Agent 2 and 3 as well. Since the sampling time does not change the adjacency matrix A_a and as result, Laplacian matrix L is unaffected. However, the controller gain (K) is obtained by LMI, which is based on a double integrator system with corresponding sampling time. Therefore, each sampling time causes a new gain (K).

5.4.2 Simulation Results

Example 1: Ts= 0.01s

With a sampling time of Ts=0.01 second and a data loss rate of r=20%. The system is operation time is 50 seconds, the controller gain (K) is

$$K = [0.1079\ 6.6506]. \tag{5.13}$$

Fig. 5.44(b) shows the speed of both of Agent 2 and 3 are increased to 0.8 and 0.9 m/s in just 1 second. After that, the speed of followers are barely increased in next 50 seconds. Since the speed is close enough to 1 m/s in first second, the trajectory of Agent 2 and 3 from Fig. 5.44(a) is controlled to move parallel with Agent 1. Fig. 5.45(a) shows the mean error, and Fig. 5.45(b) is the distribution of data transmission.

Example 2: Ts= 0.05s

In this example, sampling time is increased to 0.05 second. The controller gain (K) was obtained just as the same as Section 4.1 by LMI as

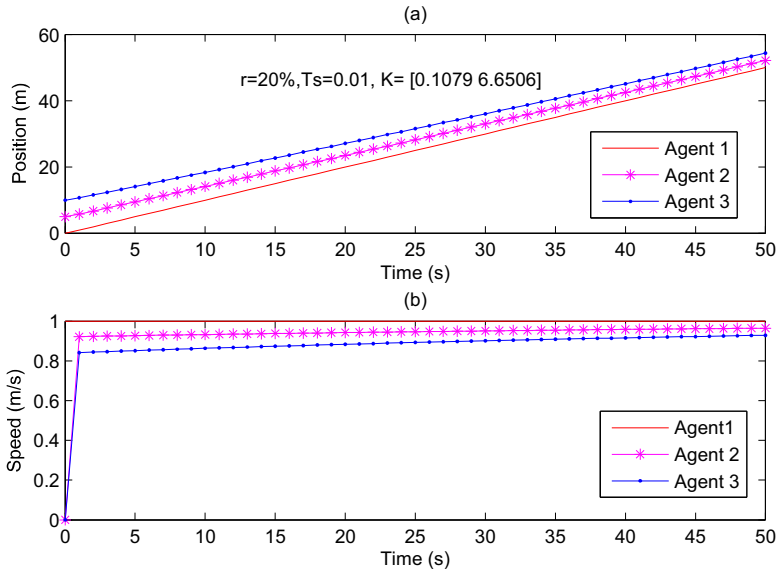


Figure 5.44: Simulation Results of Position and Speed in One Dimensional Plane with $Ts = 0.01$, r=20%

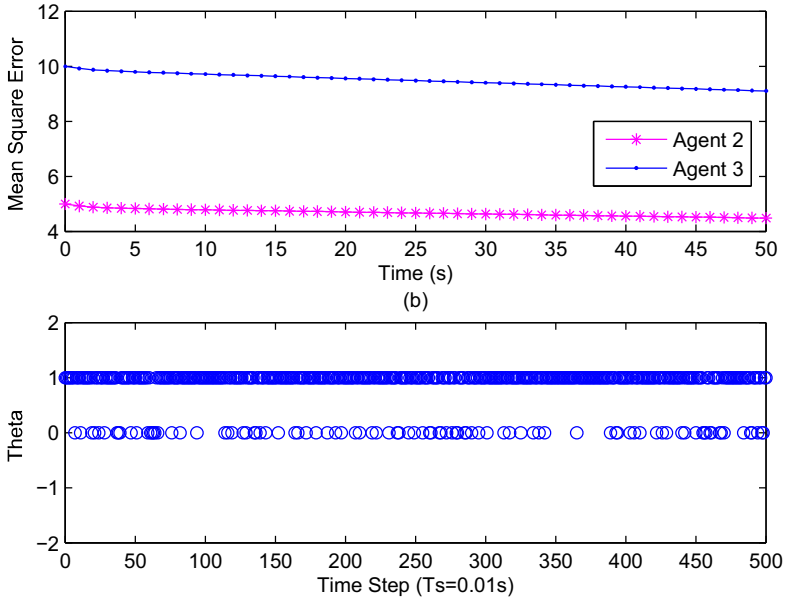


Figure 5.45: Mean Square Error with $Ts = 0.01$, r=20%

$$K = [0.0481\ 1.328]. \tag{5.14}$$

It is noticeable that with a higher sampling time, the controller gain becomes smaller.

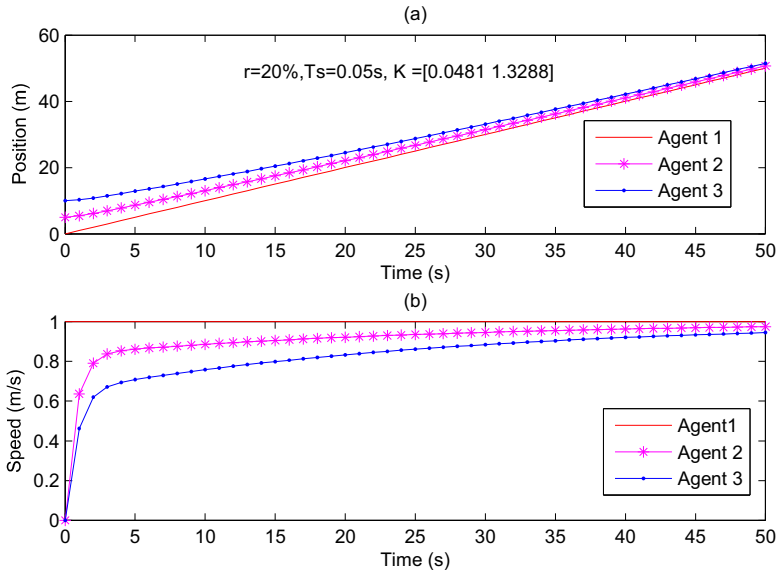


Figure 5.46: Simulation Results of Position and Speed in One Dimensional Plane with $Ts = 0.05$, r=20%

At 1 second, Agent 2 and 3 just reach 0.46 and 0.64 m/s in Fig. 5.46. Since the sampling time increases, the controlling input in the same period is not as strong as Example 1. The controller is affected by both of position and speed. However Fig. 5.47 shows the system is converged much faster than Example 1 in term of position and speed. It concludes that a smaller sampling does not guarantee a faster consensus.

Example 3: Ts**= 0.1s**

Once the sampling increases to 0.1 second, the controller gain is the same as Section 4.2 and 4.3, which is $K = [0.1036,\ 0.6603]$.

Fig. 5.46 shows Agent 2 and 3 accelerate more smoothly. there is no such a sharp jump like it is shown from Fig. 5.44 and Fig. 5.46. Fig. 5.47 shows the system is converged in 27 seconds. From previous three examples, a smaller sampling time can only increase the responding time, but it does not reduce the consensus time. It seems the consensus time becomes smaller with a higher sampling time. Therefore,

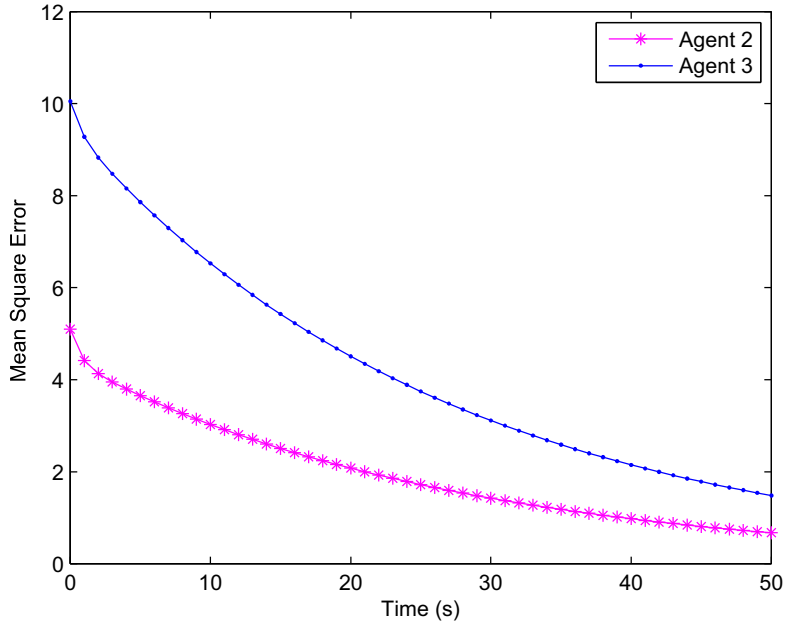


Figure 5.47: Mean Square Error with $Ts = 0.05$, r=20%

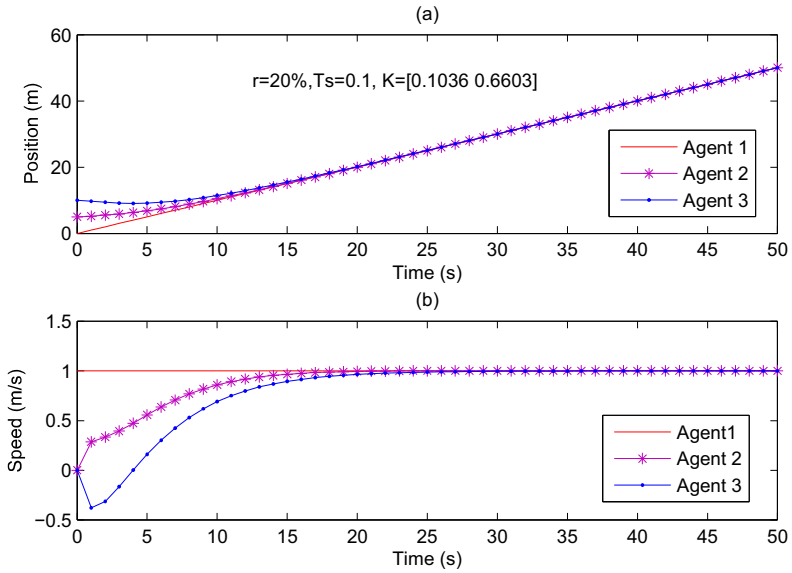


Figure 5.48: Simulation Results of Position and Speed in One Dimensional Plane with $Ts = 0.1$, r=20%

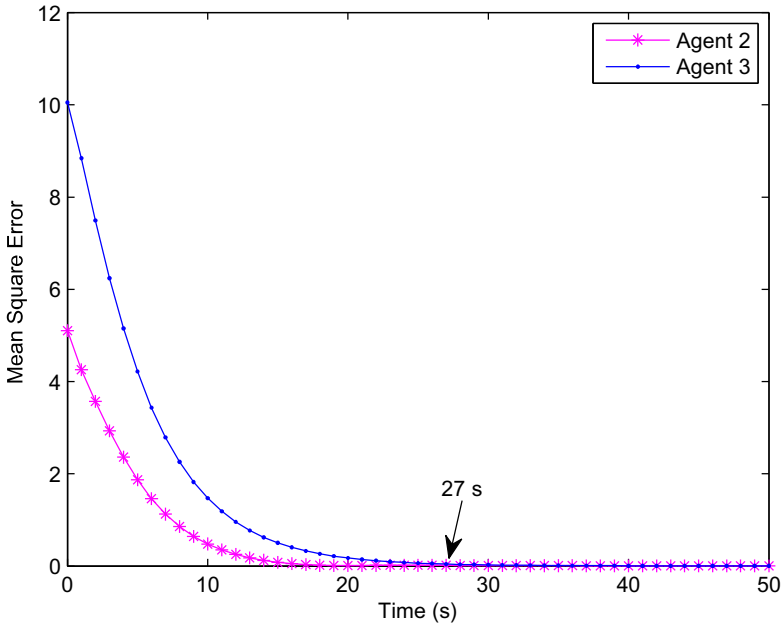


Figure 5.49: Mean Square Error with $Ts = 0.1$, r=20%

Example 4 is studied and to explore the consensus ability with a 1 second of sampling time.

Example 4: Ts**= 1s**

The corresponding controller gain (K) is obtained by LMI as

$$K = [0.0002\ 0.0407]. \tag{5.15}$$

The gain is dramatically decreased from $K = [0.1036,\ 0.6603]$.

Fig. 5.50 shows that at 43 second, Agent 3 just reaches 0.8 m/s. It is 24 seconds in Example 3 and the accelerations of Agent 2 and 3 are even smaller than all previous examples. Fig. 5.51 shows at 50 second, the system is not converged at all. Therefore, a time range is increased to 500 seconds, which is shown in Fig. 5.52. The system is still not consented completely at 500 second. Therefore, from all four examples, there is a optimal range of the sampling time. If the sampling time is not big, which can be treated as a time delay, the controller is too slow to respond, and eventually the system goes divergency.

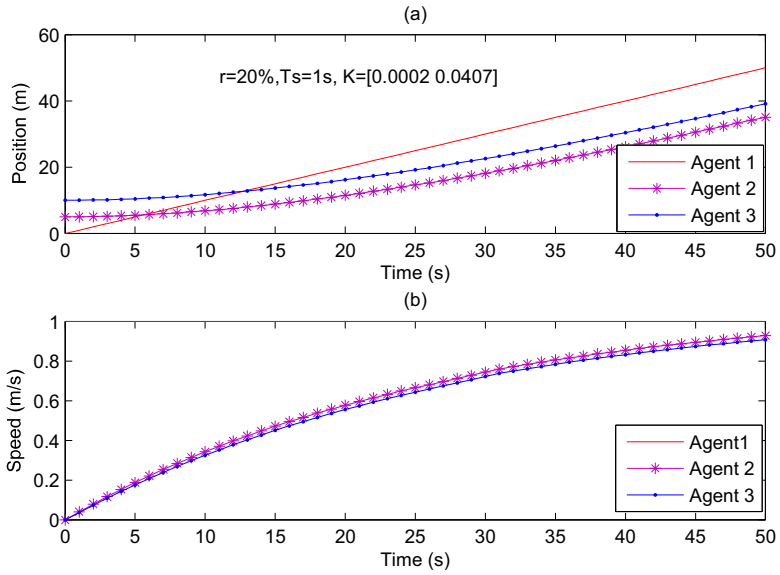


Figure 5.50: Simulation Results of Position and Speed in One Dimensional Plane with $Ts = 1$, r=20%

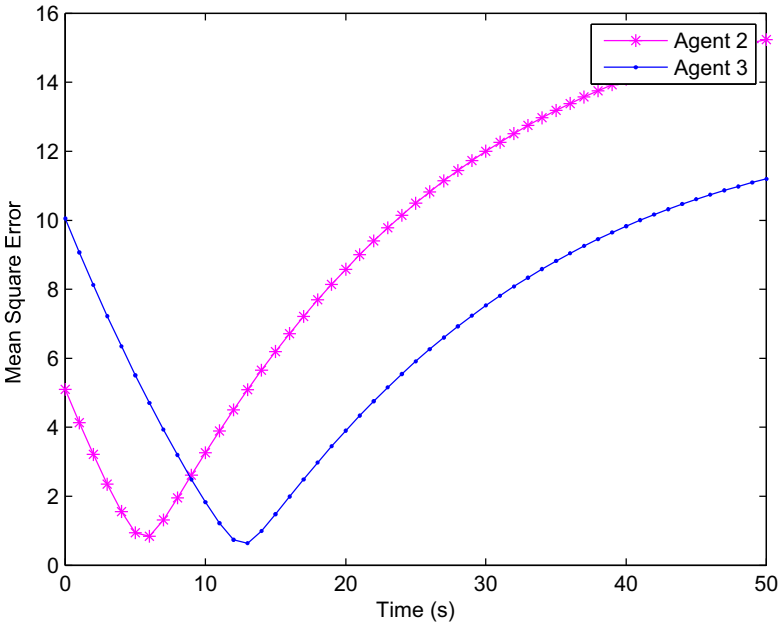


Figure 5.51: Mean Square Error with $Ts = 1$, r=20%

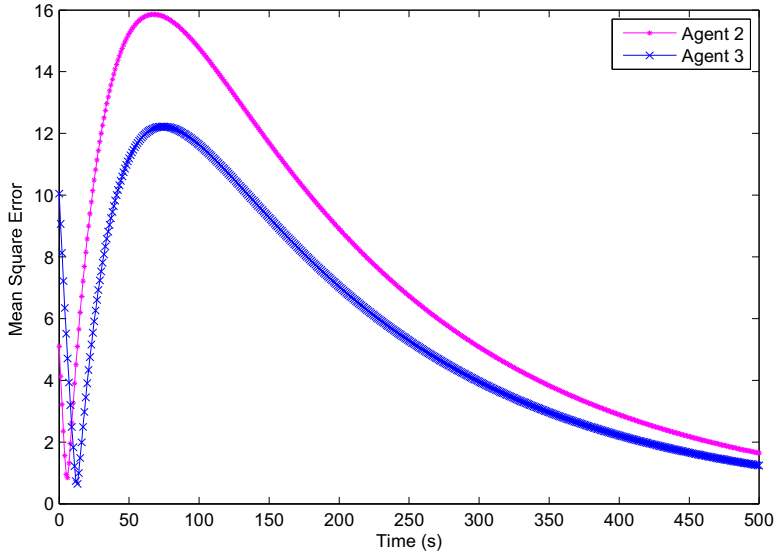


Figure 5.52: Mean Square Error with $Ts = 1$, r=20% in 500s

Table 5.4: Effects of Sampling Time (Ts) with $r = 20\%$

Example	No. of Agents	Sampling Time(s)	Controller Gain (K)	Consensus Time(s)
(1)	3	0.01	[0.1079 6.6506]	>> 50
(2)	3	0.05	[0.0481 1.3288]	> 50
(3)	3	0.1	[0.1036 0.6603]	27
(4)	3	1	[0.0002 0.0407]	> 500

5.4.3 Summary on Effect of the Sampling Time

In summary, the simulation results are explored and discussed for four examples with different sampling times. Table 5.4 shows all results of the system. Each sampling time is required with a new controller gain in order to satisfy the LMI in Theorem 1. Therefore, the result from Table 5.4 shows that the consensus ability is not affected by sampling time, but also affected by corresponding new controller gains. The goal of controller is to drive the system to reach a consensus in terms of the speed and the position. Therefore, the speed and the position are coupled, and the best controller gain is to balance the effect on both of speeds and positions. In Example 1, the gain obtained from a 0.01 second of sampling shifts the controller input into the speed instead of the position. Thus the consensus time does not become smaller as expected. With increase of sampling time, the gain is trended to become optimal for both speed and position, such as Example 3. However, once the sampling time is too

big, like Example 4, it takes longer time for system to reach consensus since the gain is so small. Therefore, for the proposed algorithm, sampling time 0.1 second is in an ideal range.

5.5 Summary on Parameters

Table 5.5: Effects of Increasing a Parameter Independently

Case	Parameter	Gain (K)	Consensus Time (s)
(1)	Loss Rate	Decrease	More Effective under 50%
(2)	Communication Weight	Unchanged	Decrease
(3)	Initial Value	Unchanged	Slightly Increase
(4)	Sampling Time	Decrease	Most Effective with 0.01 s

The parametric studies are conducted in four cases: 1) effect of data loss rate (r); 2) effect of communication weight ω_{ij}; 3) effect of initial values; and 4) effect of sampling time. Table 5.5 summaries the tendency of the effect by four parameters. Based on the results of Case 1, within $r = 50\%$, it proves that the effectiveness of the proposed controller design. The maximum of lata loss rate for the controller is determined in Example 7 with 95%.

The simulation results in Case 2 show the communication weight $\omega_{ij} \in (0, 1)$ is a proportional gain which is to distribute the information flow rate among the system. The higher weight distribution shortens the consensus time until the weight reaches the maximum value of 1. However, the consensus time is not proportionately decreased in scale with the corresponding weight. For instance, in Example 1 and 4, the weight on Agent 2 is decreased from 1 to 0.5, but the consensus time is only increased from 20 to 25 seconds.

For Case 3, the simulation results show that the larger initial value extends the consensus time. However, the initial value only slightly affects the consensus ability. The initial value is increased by 16 times from $(5, 10)$ to $(80, 160)$, the consensus time is raised from 27 to 41 seconds.

In Case 4, since each sampling time is required with a new controller gain in order to satisfy the LMI in Theorem 1, the results show that the consensus ability is not only influenced by sampling time, but also affected by corresponding new controller gains. The speed and the position are coupled, and the best controller gain is to balance the

effect on both of speeds and positions. Therefore the sampling time cannot be too big or too small in order to reach a consensus in both of the speed and the position. When the sampling time is around 0.1 second, it is in an ideal sampling range.

Based on the parametric studies, the consensus time is mainly depended on the controller gain. In Case 1, the controller gain is varied correspondingly by data loss rate, in result of affecting the consensus time. From Case 4, the controller gain is to adjust both positions and speeds. Thus the best controller gain is to balance the effect on both of speeds and positions.

5.6 Case with Five Agents

The last section of this chapter is to explore a case with a system of five agents. The topology of five agents is shown on the Fig. 5.53. Agent 1 is still the leader, and only capable if sending signals, while the rest of group members, Agent $2-5$ are follower which can receive signals from the leader directly or indirectly through neighbors. The sampling time is 0.1 second, and The communication weight is 1 for all channels. The initial positions for Agent 1, 2,3,4, and 5 are 0,5,10,15, and 20, respectively. The speeds of Agent 1 starts at 1 m/s, and the rest of agents start from 0 m/s. Since the major contribution of this thesis is to design a controller to seek a convergency of a multi-agent system in the even of data loss, in this section, two examples are studied, one is with 20% of data loss rate, and the other one is with 50% of data loss rate. Both examples verify the effectiveness of the controller in a system with a larger number of agents.

Example 1: r=20%

With a group of five agents, set $\hat{H}_0 = diag[H_{01}, H_{02}, H_{03}, H_{04}]_{8\times 8}$, $\hat{H}_1 = diag[H_{11}, H_{12}, H_{13}, H_{14}]_{8\times 8}$,. The control gain with a data loss rate bound 0.2 is from this particular case based on the sufficient condition in Theorem 1.

$$K = Y_m X_m^{-1} = [\ 0.1038 \quad 1.0473\ ]. \tag{5.16}$$

Fig. 5.54 shows the system of five agents is also able to be converged in terms of the position and the speed with the controller. Fig. 5.55(a) shows at 28 seconds the system is reached consensus. Fig. 5.56 shows controller input of gain (K) in respect of

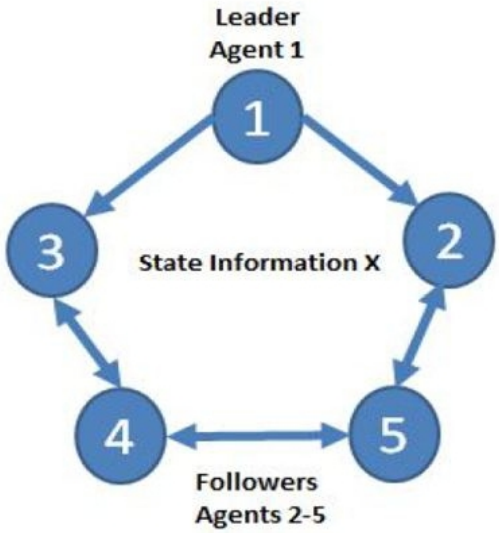


Figure 5.53: A Group of Five Agents in A Directed Graph Topology

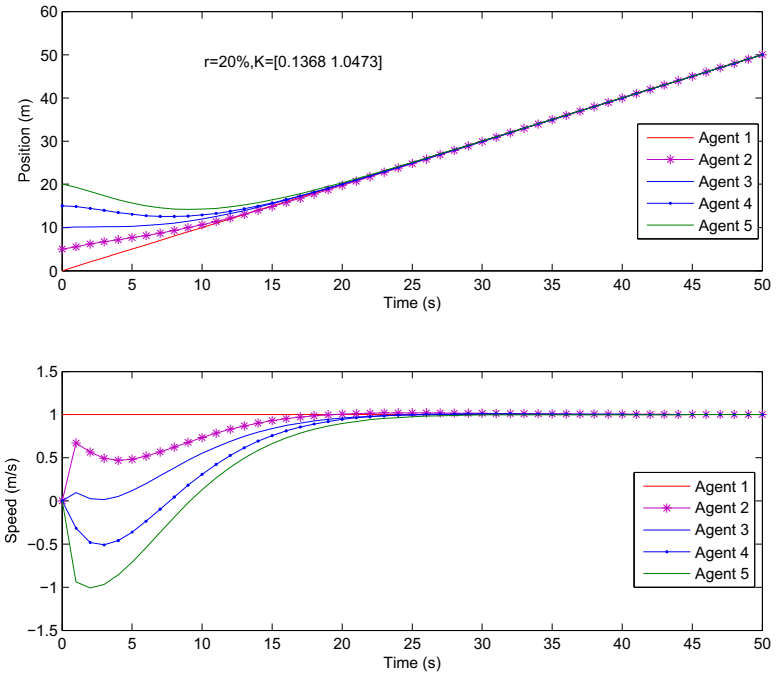


Figure 5.54: Simulation Results of Position and Speed in One Dimensional Plane with r=20% in Five Agent Case

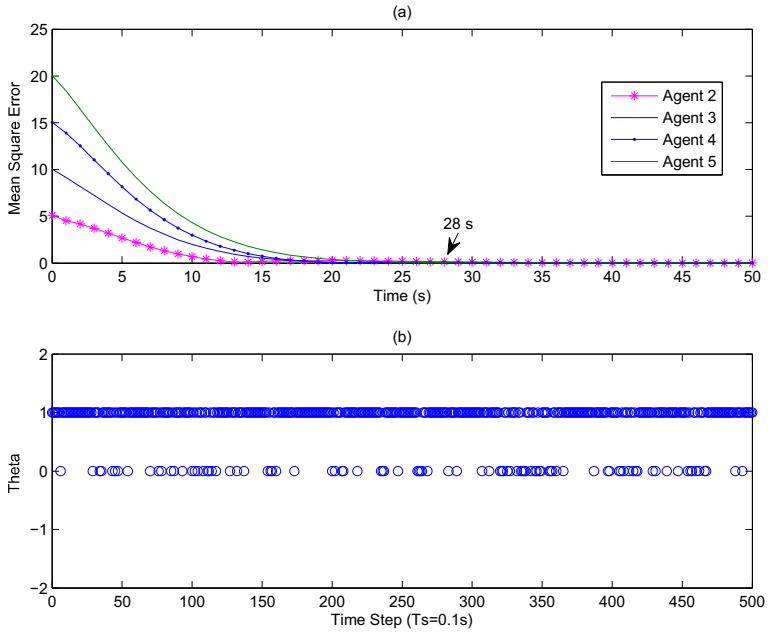


Figure 5.55: Mean Square Error and Loss Rate Distribution with r=20% in Five Agent Case

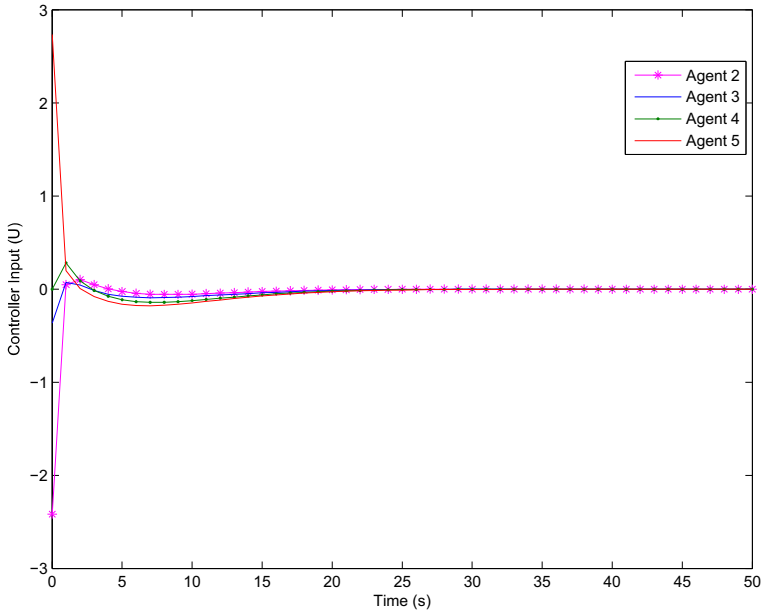


Figure 5.56: Controller Input (u) with r=20% in Five Agent Case

the time. The higher number of input in magnitude means the lager effort of controller acting on agents. Once the system reaches consensus, the input will become zero.

Example 2: r=50%

The control gain with a data loss rate bound 0.5 is from this example, and based on the sufficient condition in Theorem 1.

$$K = Y_m X_m^{-1} = [\ 0.0166 \quad 0.0283\ ]. \tag{5.17}$$

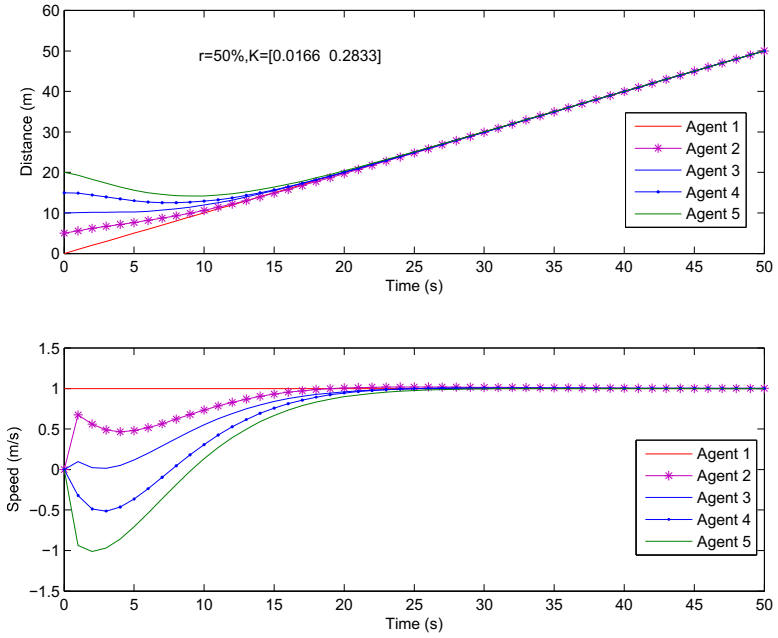


Figure 5.57: Simulation Results of Position and Speed in One Dimensional Plane with r=50% in Five Agent Case

Fig. 5.57 shows the system of five agents is also able to be converged with 50% of data loss rate. Fig. 5.58(a) shows at 36 second, the system reaches consensus. Fig. 5.59 shows controller input of gain (K).

5.7 Summary

In order to prove the effectiveness of the controller in a multi-agent system, a five agent example is simulated. Two examples are studied, one is 20% of data loss rate

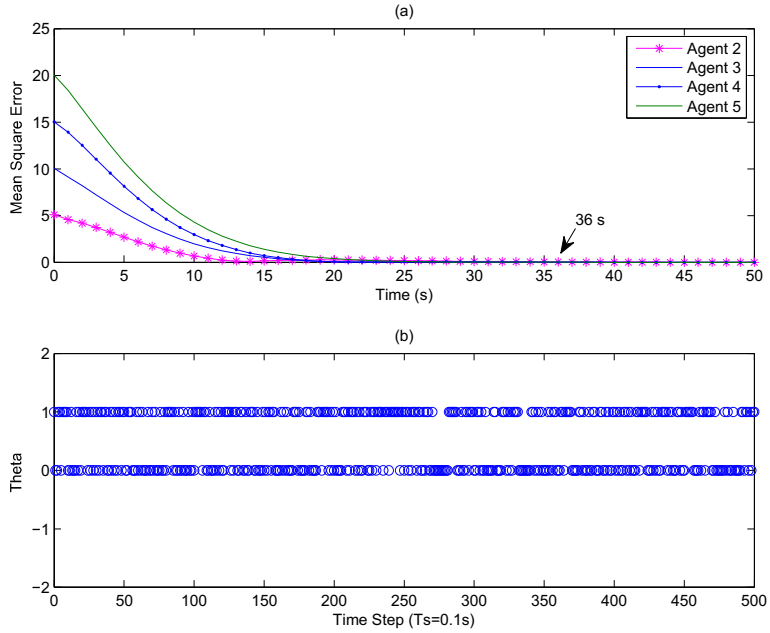


Figure 5.58: Mean Square Error and Loss Rate Distribution with r=50% in Five Agent Case

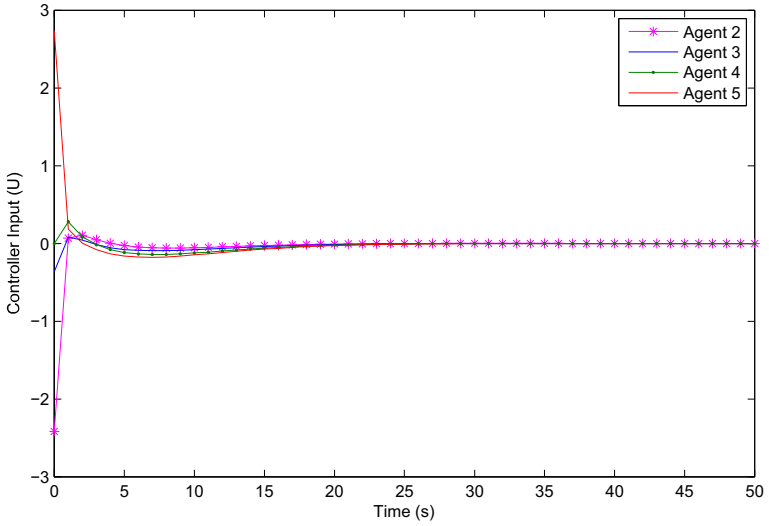


Figure 5.59: Controller Input (u) with r=50% in Five Agent Case

and the other is 50% data loss rate. Both examples prove that the effectiveness in a larger number of system.

Chapter 6

Simulation Studies on Single Integrator Consensus

In this chapter, in order to prove the proposed controller is efficient, a comparison study is conducted based on a Bernoulli-based single integrator system controller. Moreover, an actual hardware model, Pioneer 3 mobile robot, is studied. Since the kinematic of robot is in a single integrator system,the proposed algorithm needs to convert into the case of a single integrator.

6.1 Comparison Studies

There are several similar research papers dealing with random data losses. A very recent one is [56]. It investigates the average consensus with time-varying delays and data losses for multi-agent systems. The problem is formulated under the sampled-data framework by discretizing the first-order agent dynamics. The loss of data across each communication link occurs at certain probability and it is governed by a Bernoulli process. In order to compare two controllers, both methods need to be modified. For simplicity, the proposed algorithm in this thesis is called Method 1, and the algorithm designed in [56] is called Method 2. Since Method 1 is designed based on a double integrator dynamic, and it needs to be converted to a single integrator dynamic in order to match with Method 2. Method 2 is to deal with random data losses and time delays and Method 1 is designed for random data losses only. Therefore, Method 2 needs to eliminate the time delay by setting it to zero.

6.1.1 Method 1 : Modification and Results (Proposed)

In Chapter 3, a double integrator system is defined as

$$\begin{aligned} \dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= u_i \\ u_i &= -K_i \mathbf{x_i},\ i = 1, ..., n, \end{aligned} \tag{6.1}$$

where $\dot{x}_{i1}$ and $\dot{x}_{i2}$ represent the first and second information state of ith agent. In order to convert to a single integrator, Eq.(6.1) becomes

$$\begin{aligned} \dot{x}_i &= u_i \\ u_i &= -K_i x_i,\ i = 1, ..., n. \end{aligned} \tag{6.2}$$

The continuous time system can be converted to a discrete time system by setting the sampling time, $Ts = 0.05$ seconds, (same as in [56] where $Ts = 0.05$ seconds) then Eq.(6.2) becomes

$$\begin{aligned} x_i(k+1) &= Ax_i(k) + Bu_i(k) \\ u_i &= -K_i x_i,\ i = 1, ..., n, \end{aligned} \tag{6.3}$$

where A=1, and B=0.05. The topology of the system is proposed to be a five agent system, shown in Fig. 6.1.

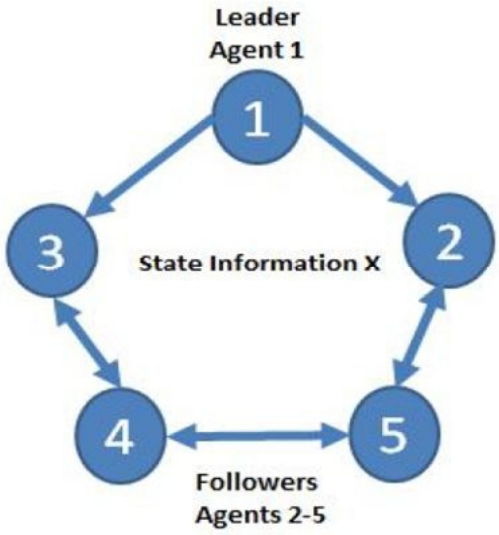


Figure 6.1: A Group of Five Agents in A Directed Graph Topology

Based on Theorem 1, the controller gain (K) is $K = 1.3288$ with the bound of data loss rate as 0.2. Initial values are 1, 5, 10, 15, and 20, respectively for all agents. Fig. 6.2 shows the simulation results by Method 1, when the loss rate is set as 20%. Fig. 6.3 shows that the system is converged in 23 seconds, if the value of error is under 0.05 to be considered as reaching its consensus.

6.1.2 Method 2: Modification and Results ([56])

In [55], the first-order dynamics is defined as

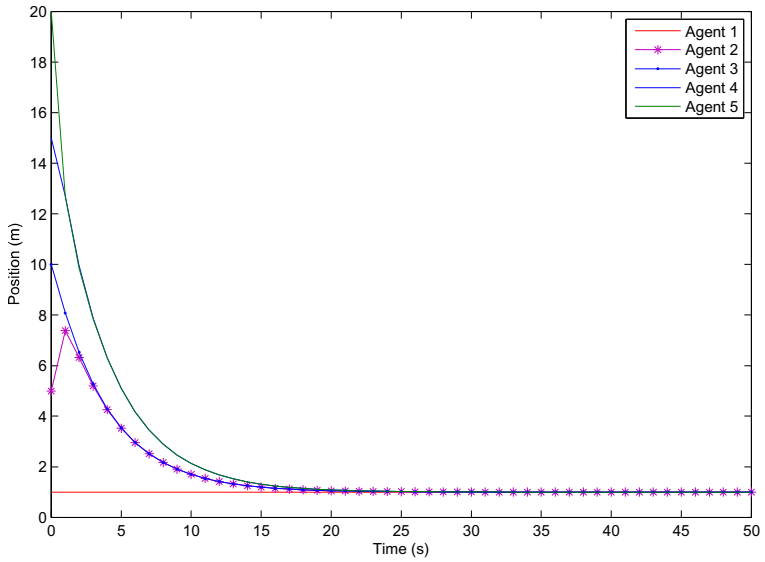


Figure 6.2: Simulation Result of Position and Speed in One Dimensional Plane by Method 1 with r=20%

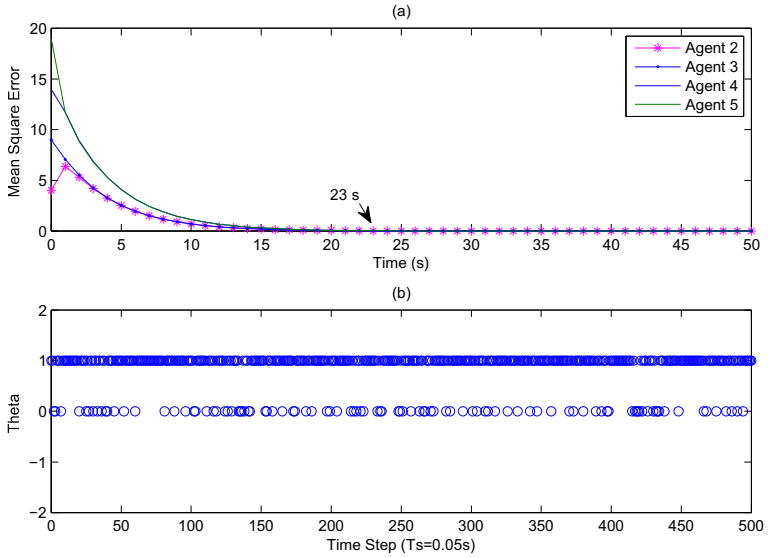


Figure 6.3: Mean Square Error and Data Loss Distribution by Proposed Method with r=20%

$$\dot{x}_i = u_i,\ i = 1, 2, ..., n. \tag{6.4}$$

With sampling period T and a zero-order hold, the agent dynamics is discretized as

$$x_i(k+1) = x_i(k) + Tu_i(k). \tag{6.5}$$

An uniform and time-varying delays and data loss among agents. Therefore a controller is adopted,

$$u_i(k) = -\gamma_c \Sigma_{j \in N_i} \gamma_{ij}(k) a_{ij}[x_i(k - d(k)) - x_j(k - d(k))], \tag{6.6}$$

where $\gamma_{ij}(k)$=1 if there is no packet loss from Agent j to agent i, and $\gamma_{ij}(k)$=0 otherwise. $d(k)$ is the delay from Agent j to Agent i at time instant k. The packet loss is governed by a Bernoulli process with uniform probability $0 < p \leq 1$. In order to compare with Method 1, $d(k)$ is set to be zero. Then Eq.(6.6) becomes

$$u_i(k) = -\gamma_c \Sigma_{j \in N_i} \gamma_{ij}(k) a_{ij}(x_i(k) - x_j(k)), \tag{6.7}$$

Set T=0.05 seconds, $p = 0.8$, which equals $r = 20\%$ in Method 1. The controller gain ω_c is 0.6, according to Section IV in [56].

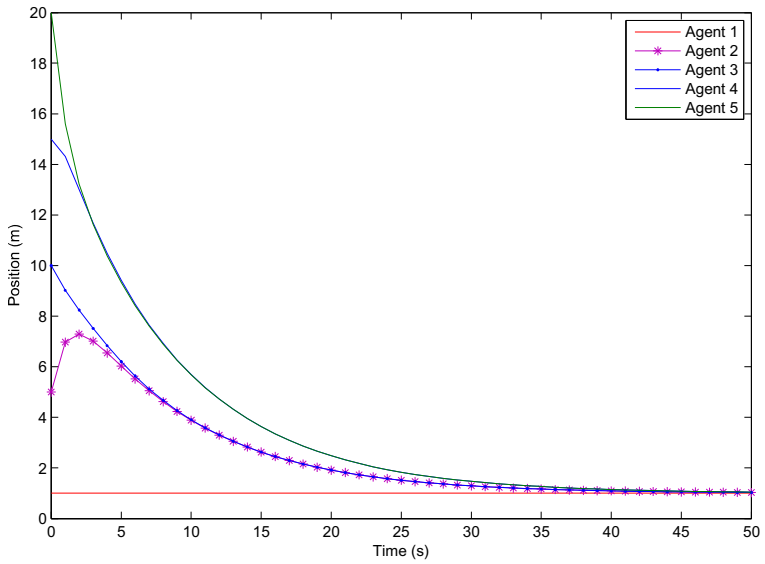


Figure 6.4: Simulation Result of Position and Speed in One Dimensional Plane by [56] with r=20%

Fig. 6.4 shows a simulation result by Method 2 ([56]), when all conditions are the same with Method 1 (Proposed). Fig. 6.3 shows the system is converged in 50 seconds.

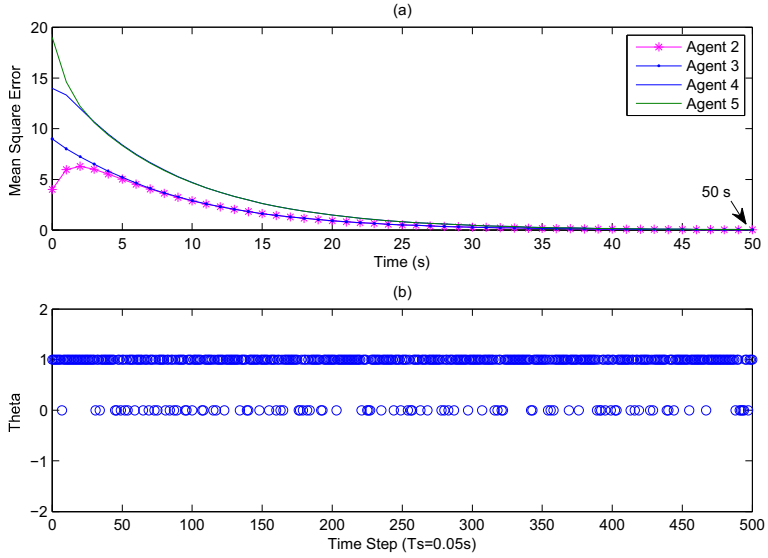


Figure 6.5: Mean Square Error and Data Loss Distribution by Method 2 with r=20%

6.1.3 Discussions

Since both methods are based on different approaches and assumptions, It is hard to compare directly. The comparison results show that at 20% of data loss rate , 0.05 second of sampling time, and 0 of time delay of conditions and assumptions, Method 1 (Proposed) converges faster than Method 2 ([56]). It only gives some basic ideas how efficient of the proposed controller by comparing with others

6.2 Pioneer Robot Modeling and Simulation Studies

As discussed in Chapter 5, all figures are one dimensional, and they only show the system convergency but actual trajectories of each agent. In this section, a actual Pioneer mobile robot is used to be modeled as a gent. Therefore, each agent represents a robot, and then is implemented the proposed controller to study the performance in a real hardware model. Pioneer mobile robots, shown in Fig. 6.2(a) are durable, differential-drive robots for academic researchers. In [54], the base Pioneer 3-DX platform is assembled with motors featuring 500-tick encoders, 19 cm wheels, tough aluminum body, 8 forward-facing ultrasonic (sonar) sensors, 8 optional real-facing sonar, 1, 2 or 3 hot-swappable batteries. The base Pioneer $3-DX$ platform can carry a payload of up to 23 kg with maximum forward and backward speed of 1.2 m/s, and rotation speed of 300 deg/s.

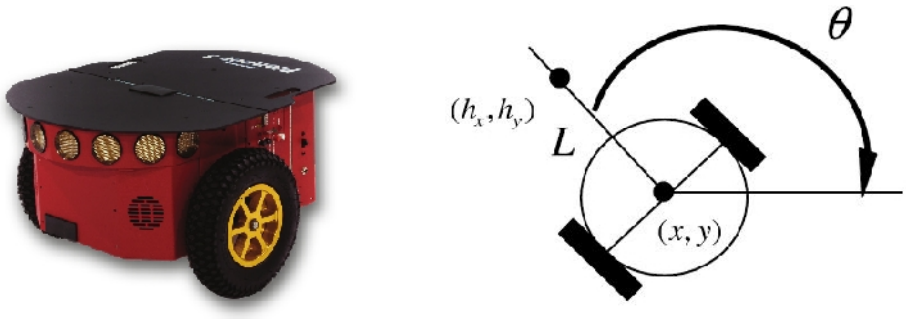


Figure 6.6: (a) Pioneer 3-DX Mobile Robot [54] ; (b) Hand position for P3 mobile robot

A kinematic about center position (x, y) can be described in Eq.(6.8); However the most popular way to describe the kinematic of robots is to define a hand position for each robot. In [48], a hand position is shown in Fig. 6.2(b). The center position of the robot is (x,y), and the hand position is (h_x,h_y) which lies 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 (x, y). Orientation angle (θ) is zero at 0 deg, and counter-clockwise rotation is defined as positive.

$$\begin{aligned} \dot{x} &= v\,cos(\theta) \\ \dot{y} &= v\,sin(\theta) \\ \dot{\theta} &= w, \end{aligned} \tag{6.8}$$

where $[x, y]$ 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.(6.9)

$$\begin{aligned} h_x &= x + Lcos(\theta) \\ h_y &= y + Lsin(\theta), \end{aligned} \tag{6.9}$$

Now, differentiate Eq.(6.9) with respect to time and substitute Eq.(6.8) into it, then

$$\begin{aligned} \dot{h}_x &= cos(\theta)v - Lsin(\theta)w \\ \dot{h}_y &= sin(\theta)v + Lcos(\theta)w, \end{aligned} \tag{6.10}$$

and

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \end{bmatrix} = \begin{bmatrix} cos(\theta) & -Lsin(\theta) \\ sin(\theta) & Lcos(\theta) \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \tag{6.11}$$

Therefore, based on Eq.(6.10) and Eq.(6.11) the hand position of robot can be controlled by manipulating linear and angular speed (v, w). Let

$$\begin{aligned} v &= cos(\theta)u_x - Lsin(\theta)u_y \\ w &= -\tfrac{1}{L}sin(\theta)u_x + \tfrac{1}{L}cos(\theta)u_y \end{aligned} \tag{6.12}$$

and

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} cos(\theta) & Lsin(\theta) \\ -\tfrac{1}{L}sin(\theta) & \tfrac{1}{L}cos(\theta) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{6.13}$$

then

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{6.14}$$

and

$$\dot{\mathbf{h}} = \mathbf{u}. \tag{6.15}$$

Recall from Section 6.1.1, the ith agent is described as

$$\begin{bmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \end{bmatrix} = \begin{bmatrix} x_{i2} \\ u_i \end{bmatrix}. \tag{6.16}$$

We can see that Eq.(6.14) is a single integrator system, and Eq.(6.16) is a double integrator system. Therefore the latter has to be converted to a single integrator system in order to model Pioneer 3 robots. Then

$$\dot{\mathbf{x}} = \mathbf{u} \tag{6.17}$$

$$\begin{bmatrix} \dot{x}_x \\ \dot{x}_y \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \tag{6.18}$$

The standard system equation is

$$\dot{x} = A\mathbf{x} + B\mathbf{u} \tag{6.19}$$

and

$$\begin{bmatrix} \dot{x}_x \\ \dot{x}_y \end{bmatrix} = A \begin{bmatrix} x_x \\ x_y \end{bmatrix} + B \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \tag{6.20}$$

where $A \in R^{2\times2}$ and $B \in R^{2\times2}$. In order to become Eq.(6.18) from Eq.(6.20), then let

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{6.21}$$

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

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{6.22}$$

Since the LMI is not changed, and it still can get K from existing LMI with new A and B in discrete time system. The gain (K) with $r = 20\%$ is

$$K = \begin{bmatrix} 0.6838 & 0 \\ 0 & 0.6838 \end{bmatrix}. \tag{6.23}$$

Example 1: Rectilinear, r=20%

The data loss rate is very varied and depending on a lot of factors. For a User Datagram Protocol (UDP), a reasonable packet loss rate is less than 15%, according to [55]. Moreover, for a Transmission Control Protocol (TCP), there will be no packet loss. Therefore a data loss rate of 20% is used into these simulations. Fig. 6.7 is the topology of the system is to be simulated, Robot 1 is the leader, and Robot 2 and 3 are followers. Fig. 6.8 shows the schematic of data transmission among robots. Firstly, Robot 1 sends the data at the first sampling time step, and at the same time Robot 2 and 3 exchanges the data as well. Secondly, Robot 2 and 3 receive the data from Robot 1 at the second sampling time step, and then start to move. Therefore, the transmission delay is about a sampling period, which is 0.1 second.

Robot 1 is the leader, started at $(0, 500)$ mm. Robot 2 and 3 are followers, start at $(500, 0)$ mm and $(1000, 0)$ mm respectively. Since the sampling time is 0.1 second, Robot 2 and 3 have 0.1 second delay from Robot 1. It is the reason that robot 1 moves farther than robot 2 and 3 in Fig. 6.9, and it is also the reason, Fig. 6.11 shows that the mean square error is 206 mm, instead of 0 mm. After the controller gain is calculated, Fig. 6.9 shows the actual movement trajectory of the system with $r = 20\%$. Furthermore, it is not physically possible for Robot 2 and 3 are converged with Robot 1. Therefore, the initial positions of Robot 2 and 3 are arbitrarily shifted to $(1000, -500)$ mm and $(2000, -1000)$ mm. The virtual trajectory of robotsis converged to one, shown in Fig. 6.9. However, with such shift of starting points, the actual trajectory of robot has a safe distance d from its virtual trajectory.

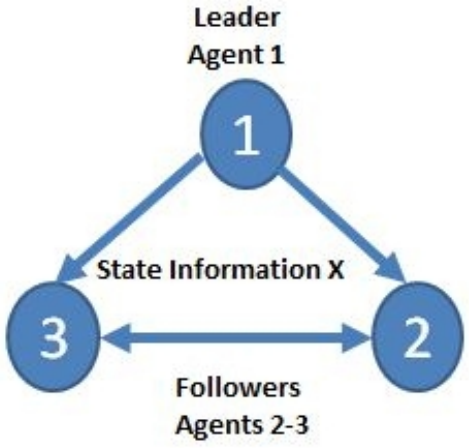


Figure 6.7: A Group of Three Agents in A Directed Graph Topology

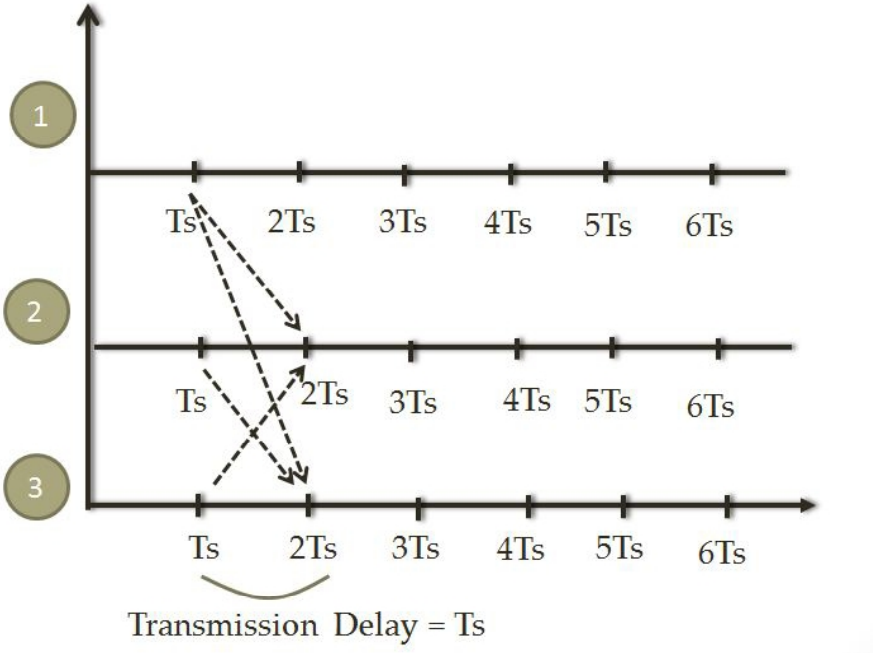


Figure 6.8: Data Flow Diagram with One Sampling Period Delay

$$d = \sqrt{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2} \tag{6.24}$$

where (x_i, y_i),i=2,3,...,n, is the projected initial position of ith robot. $(\bar{x}_i, \bar{y}_i)$ is the actual initial position. For example, in Fig. 6.9, the projected initial position of Robot 2 is $(500, 0)$ mm , and in Fig. 7.4 the actual initial position is $(1000, -500)$ mm. The

d between actual and virtual trajectory is 707.1 mm according to Eq.(6.24).

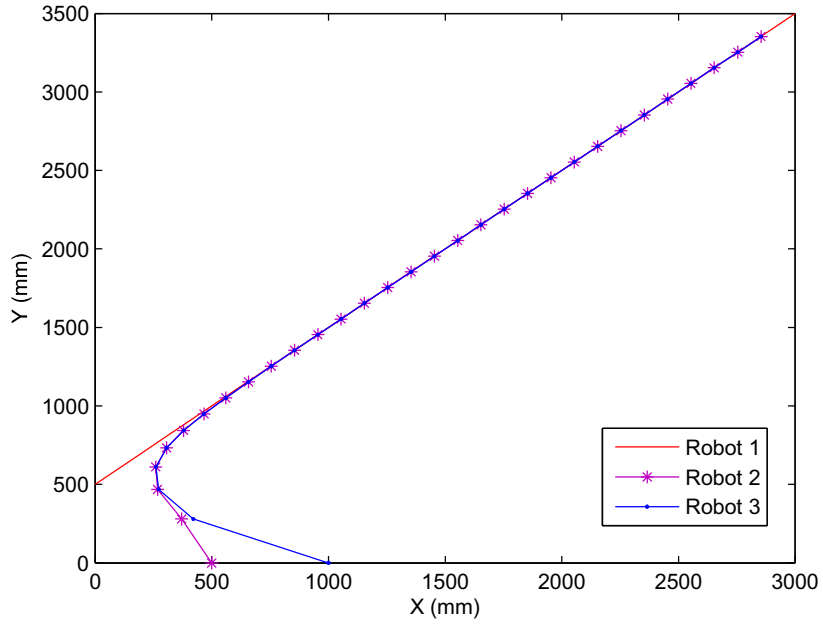


Figure 6.9: Trajectory of Pioneer Robots without Offset in r=20%

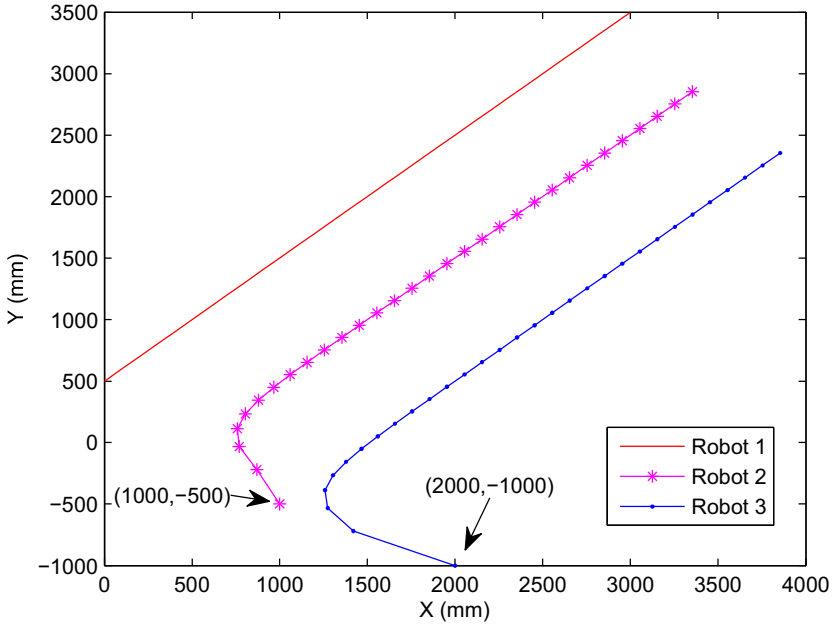


Figure 6.10: Trajectory of Pioneer Robots with Offset in r=20%

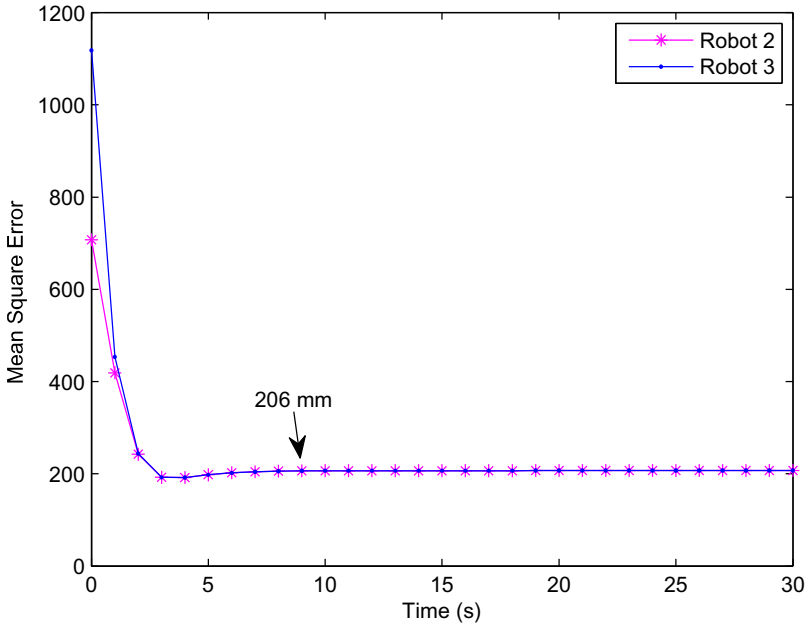


Figure 6.11: Mean Square Error with A Rectilinear Trajectory in r=20%

Example 2: Curvilinear, r=20%

At this time, Robot 2 and 3 need to follow a curvilinear trajectory which has 3000 mm radius. Fig. 6.12 shows the actual movement trajectory of the system of three pioneer $3-DX$ robots in a curvilinear trajectory with starting point offset. Robot 1 starts at $(0, 3000)$ mm, and Robot 2 and 3 start at $(0, 0)$ mm and $(2000, 0)$ mm respectively. Robot 2 and 3 are converged with robot 1 about 11 seconds. In Fig. 6.13, the starting point of Robot 2 and 3 are shifted to $(0, -500)$ mm, $(2000, -1000)$ mm. Therefore the safe distance d is created. Fig. 6.14 shows the error is under 500 mm after 6 seconds. Because the direction of Robot 1 constantly changes, it is very challenging for followers to track it's trajectory. It is hard to maintain the error as stable as Example 1 for a rectilinear trajectory. At 29 seconds, the error is even over 500 mm again due to a sharp turn of Agent 1. However. it verifies that the controller is still very effective in a single integrator dynamics.

6.3 Summary

This chapter explores the effectiveness in the single integrator dynamics. A comparison study is conducted to show the effectiveness of the controller in the single integrator dynamics. The comparison study is based on the condition of 20% of data

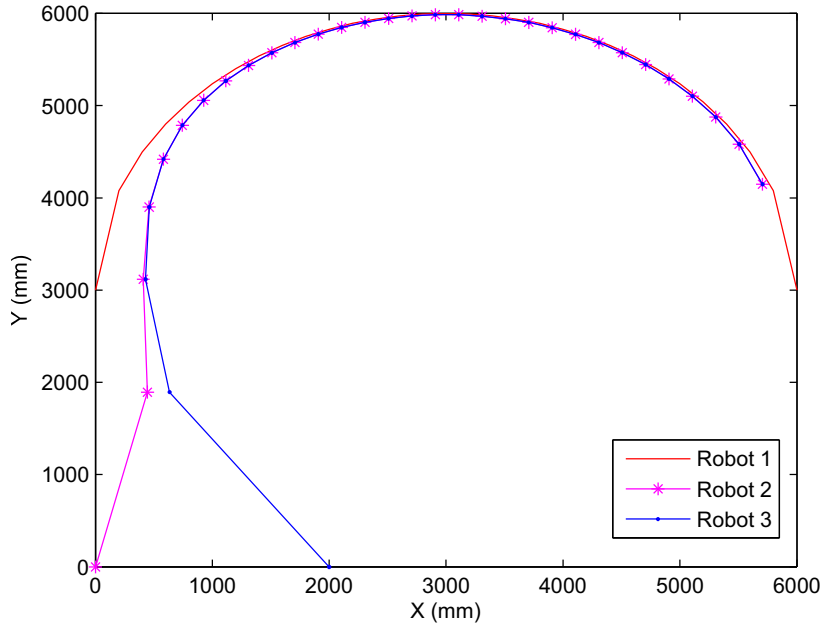


Figure 6.12: Trajectory of Pioneer Robots without Offset in r=20%

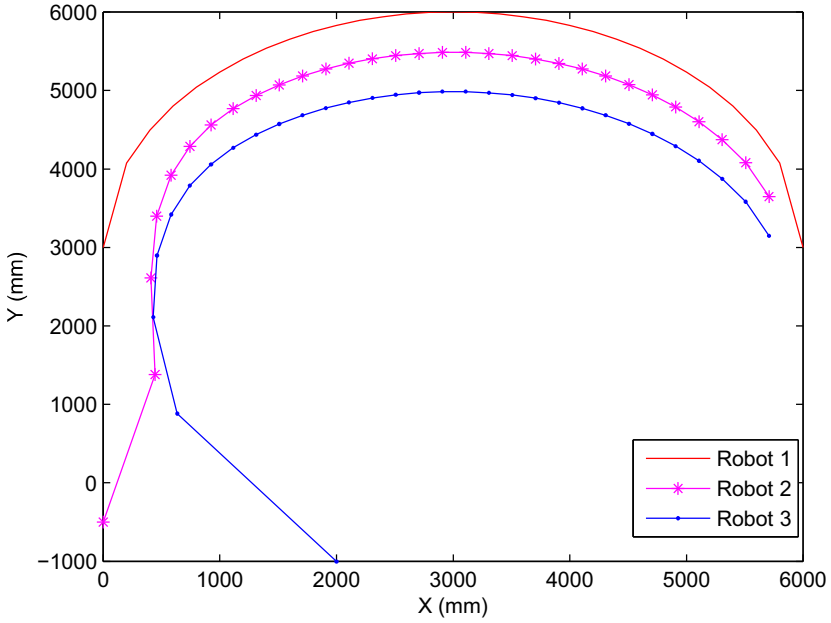


Figure 6.13: Trajectory of Pioneer Robots with Offset in r=20%

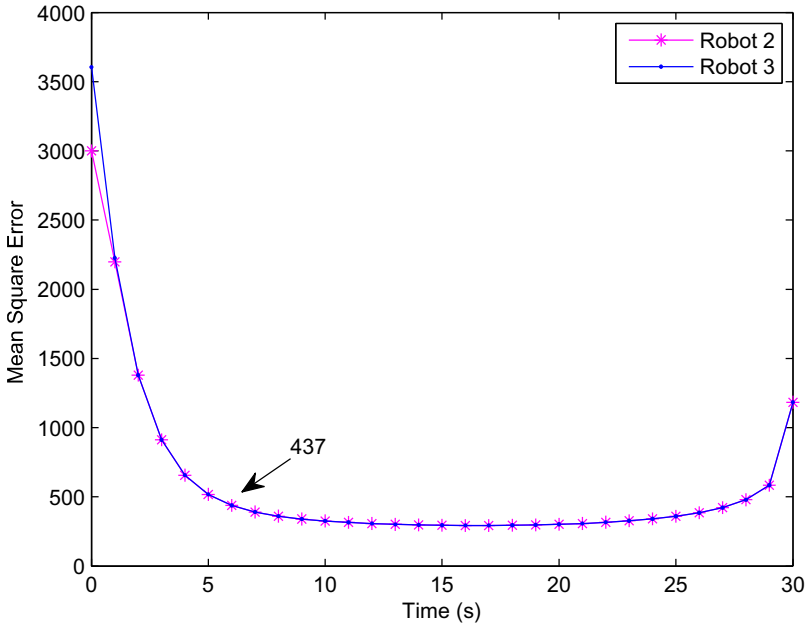


Figure 6.14: Mean Square Error with A Curvilinear Trajectory in r=20%

loss rate , 0.05 second of sampling time, and 0 of time delay. Both methods are able to reach the consensus for a five agent system, but Method 1 (proposed) converges in 23 seconds, while Method 2 ([56]) converges in 50 seconds. It shows that Method 1 is more efficient than Method 2 in this particular case. Moreover, actual trajectories are also shown by simulating a three Pioneer 3-DX robot system. Both recliner and curvilinear examples also verify that the effectiveness in a single integrator dynamics.

Chapter 7

Experimental Studies

7.1 Hardware Setup

An experimental verification is completed based on two Pioneer 3-DX robots, shown in Fig. 7.1 (a), and a Pioneer 3-AT robot, shown in Fig. 7.1 (b). Pioneer 3-DX is introduced in Chapter 6 already. Pioneer 3-AT is a similar class, except that it is an all-purpose outdoor base instead of indoor base for Pioneer 3-DX. The maximum forward or backward speed is 0.7 m/s, and the maximin rotation speed is 140 deg/s. It can be loaded up to 12 kg, and it is suitable for asphalt, flooring, sand, and dirt.

Figure 7.1: (a) Pioneer 3-DX Mobile Robot ; (b) Pioneer 3-AT Mobile Robot [54]

A local wireless network over robots is created by a model DIR-615 D-link Router. Robots are connected through each laptop mounted on the top of corresponding robots. Once the laptop picks up the wireless connection, a topology of a multi-agent system can be established according to pre-requirement. Pioneer robots are programmed by a C++ based interface, called Advanced Robot Interface for Applications (ARIA). Each communication channel is created by ArSocket function according to the proposed topology. In this experiment, the topology of the system is the exactly same with Chapter 6. ArSocket command has two options of choosing internet protocol suite, one is called User Datagram Protocol (UDP), and the other one is called 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 deliver the original data stream in the proper order to the receiver. On the other hand, UDP emphasizes reduced latency over reliability. Therefore it does not guarantee the success of transmission. During an actual network testing on UDP, the proximate data loss rate is less than 2% with 5000 packet transmission between robots. If an UDP used for internet transmission protocol, then the data loss can be negligible among system. Therefore it is not an ideal experimental environment. The reason of choosing TCP is 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 rate is required to 20%, then it is just simply set $rand()\%100 >= 80$. It means 80% data can be successfully transmitted.

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. Eq.(6.12) is derived angular w and linear speed v of robots, then converted to be right wheel speed v_r, and left wheel speed v_l in Eq.(7.1).

$$\begin{aligned} v_r &= v + \tfrac{wl}{2} \\ v_l &= v - \tfrac{wl}{2} \end{aligned} \tag{7.1}$$

where, v_r and v_l are right and left wheel speeds. l is the axial distance between two wheels. Therefore the controller input is easily commanded by $robot.setVel2(vl, vr)$, where vl and vr are v_r and v_l in Eq.(7.1), respectively.

7.2 Experiment Results

In the experiment, two examples are considered like in the simulation on Chapter 6, one is rectilinear trajectory and the the other is curvilinear trajectory. Both examples are subjected with a 20% data loss rate. In [50], the network tested in the experiment was keeping in a stable condition since over 90% packets could be received in a fixed time domain. Therefore, 20% of data loss rate is comparatively large in a reality.

7.2.1 Example 1: Rectilinear

The goal of this experiment is to validate the simulation in result of verifying the proposed algorithm. Therefore the experiment is required to set up as close possible to the simulation in Chapter 6. Robot 1 (Pioneer 3-AT) is the leader due to its larger size. The starting point is $(0, 500)$ mm, which is the same starting point to

simulation in Chapter 6 (Fig. 6.9) with a 45 initial degrees of orientation. Robot 2 and Robot 3 both are Pioneer 3-DX, and supposedly placed at $(0, 0)$ mm and $(1000, 0)$ mm respectively, and both are in 0 degree orientation. These starting points are called projected points, which are what robots take these points into the controlling algorithm. However, all robots would be converged into one point and collided each others if they are placed in the projected starting points. Same as Chapter 6, both starting points of Robot 2 and 3 are shifted vertically downward by 1000 mm to $(500, -1000)$ mm $(1000, -2000)$ mm. These points are called actual starting points in order to avoid the collision. The speed of both wheels in Robot 1 are programmed to be 100 mm/s (0.1 m/s) with a initial angle of 45 degrees. The operation time is 40 seconds. At every 0.5 seconds, a data is stored for further analysis. Same as simulation, the transmission delay is one sampling period, shown in Fig. 7.2. Fig. 7.3 and Fig. 7.4 are the result of trajectory of all three Pioneer robots without and with offset, respectively . Due to hardware limitation, at first a couple of seconds, there is noise-like motions for both Robot 2 and 3 when they was trying to correct their trajectories based on the proposed controlling algorithm. From Fig. 7.5, about 20 seconds, both Robot 2 and 3 are stable compared to Robot 1's position. Fig. 7.6 shows the actual test framework from 0 to 30 seconds.

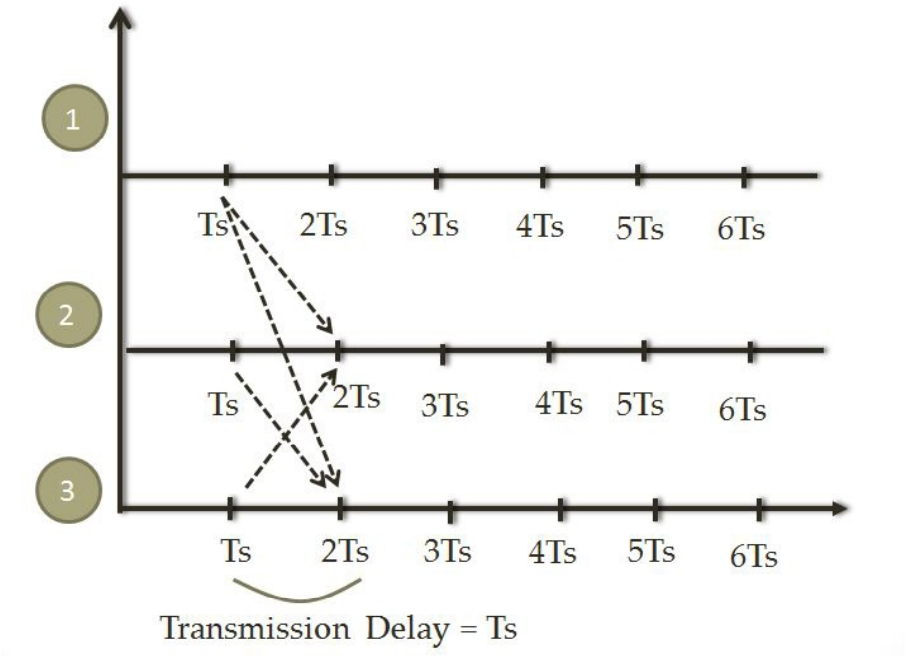


Figure 7.2: Data Flow Diagram with One Sampling Period Delay

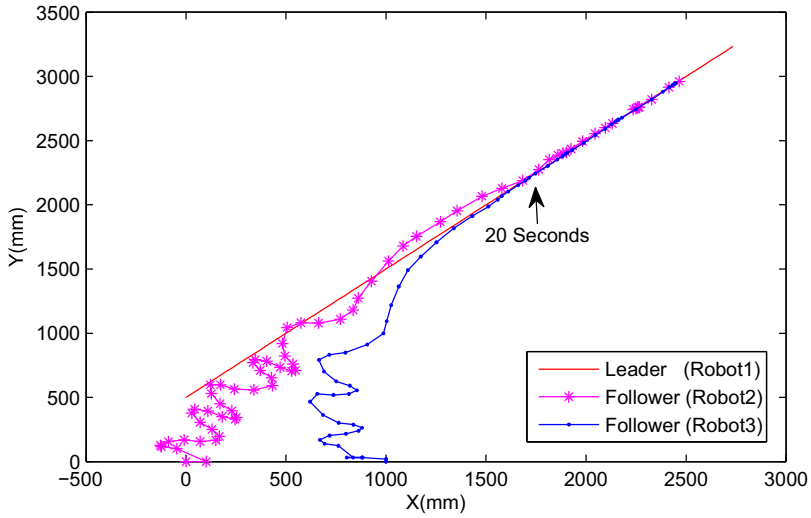


Figure 7.3: Rectilinear Trajectory of Pioneer Robots without Offset in r=20%, Converged in 20 Seconds

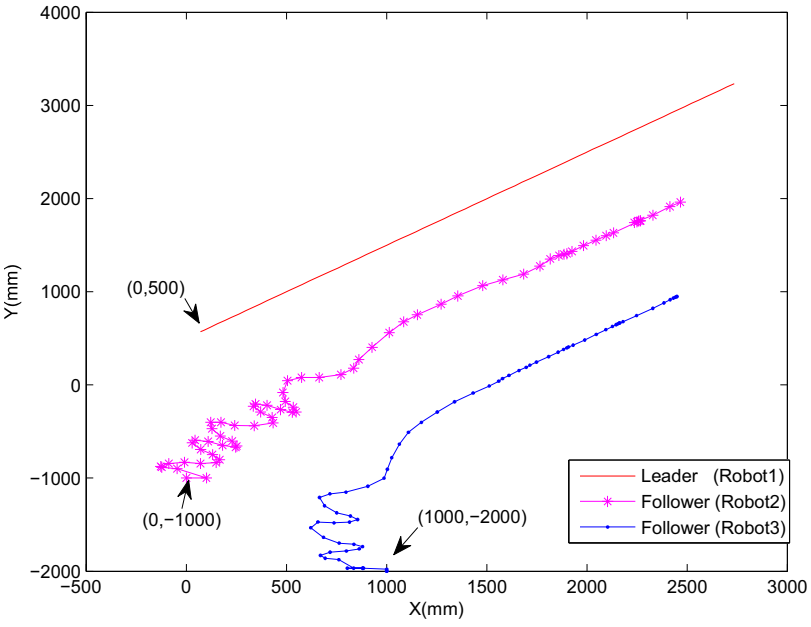


Figure 7.4: Rectilinear Trajectory of Pioneer Robots with Offset in r=20%, Converged in 20 Seconds

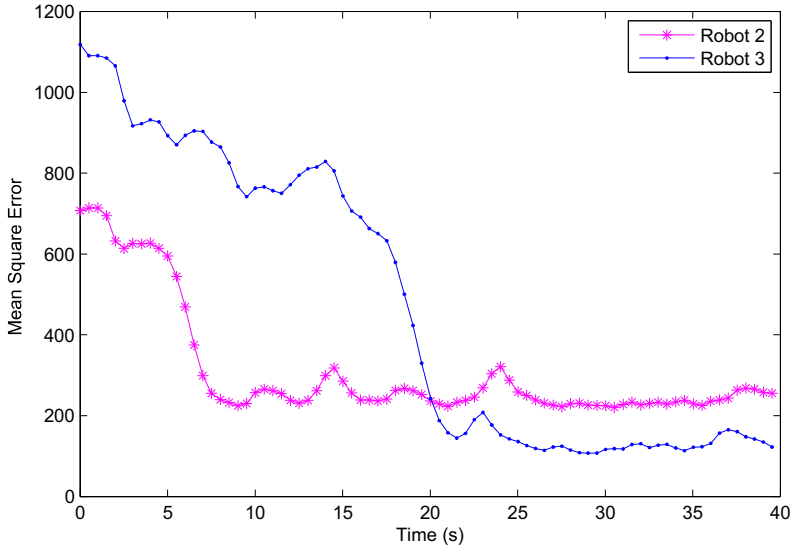


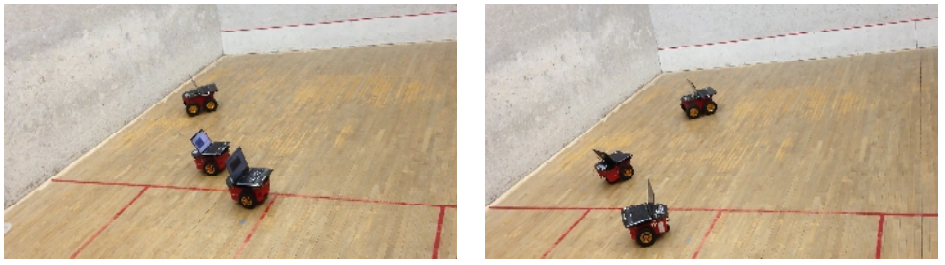
Figure 7.5: Mean Square Error of Rectilinear Trajectory with r=20%, Converged in 20 Seconds

7.2.2 Example 2: Curvilinear

Another example of curvilinear with a 20% data loss rate is also carried out. The projected starting points of Robot 1, 2, and 3 are (0,0),(500,0), and (1000,0) mm, while the actual starting points are (0,0), (500,-1000), and (1000,-2000) mm respectively. Fig. 7.7 and Fig. 7.8 show both of curvilinear trajectories of robots without and with offset, respectively. Since Robot 3 is farther than Robot 2, the trajectory of Robot 3 is much more oscillated than Robot 2. Fig. 7.9 is the error curve. Robot 2 and Robot 3 track Robot 1 at 27 seconds. As mentioned at Chapter 6, the error is theoretically zero. However because of the transmission delay which caused by packet losses.

7.3 Summary

In this chapter, both rectilinear and curvilinear cases are completed based on a test platform of Pioneer 3 robots. In order to consider the data loss, 20% of data loss rate is arbitrarily introduced. Compared to the simulation results in Chapter 6, Fig. 6.9 shows the trajectory of robots in simulation. The speed of the leader is 1 m/s, Robot 2 and 3 are converged with the leader in 7 seconds, which can be shown in Fig. 6.11. However in Fig. 7.3, the hardware experiment result shows Robot 2 and 3 are experienced oscillations in first 10 seconds, and start to reach consensus with the leader in 20 seconds. Both results are subjected with 20% data loss rate, and the



(a) Ts=0 second (b) Ts=10 seconds

(c) Ts=20 seconds (d) Ts=30 seconds

Figure 7.6: Snapshot of Experiment in Rectilinear Trajectory from 0 to 30 seconds

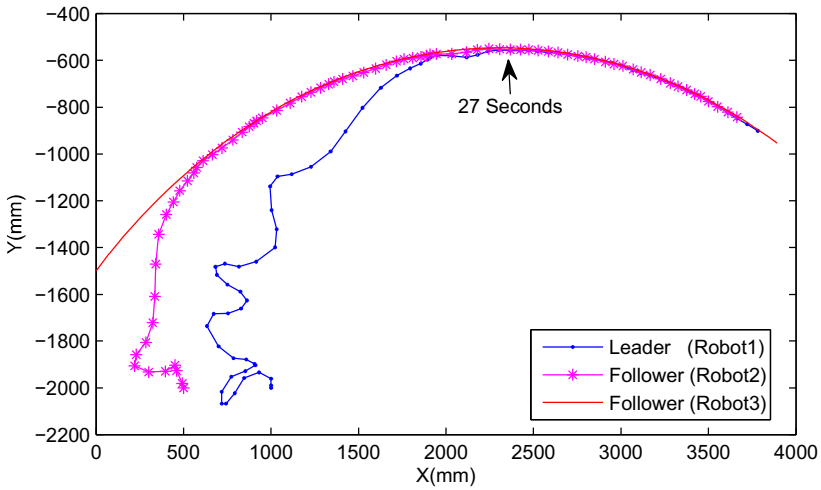


Figure 7.7: Curvilinear Trajectory of Pioneer Robots without Offset in r=20%, Converged in 27 Seconds

leader has the same speed. However the simulation is more efficient in consensus than the experiment, because it does not need to consider any hardware limitation, like the speed of robots. In simulation, Robot 2 and 3 can theoretically reach Robot 1 with infinite speeds. However in hardware experiment, the speed of robots are limited. It is more obvious in the curvilinear example. Fig. 6.12 shows Robot 2 and 3 follow the

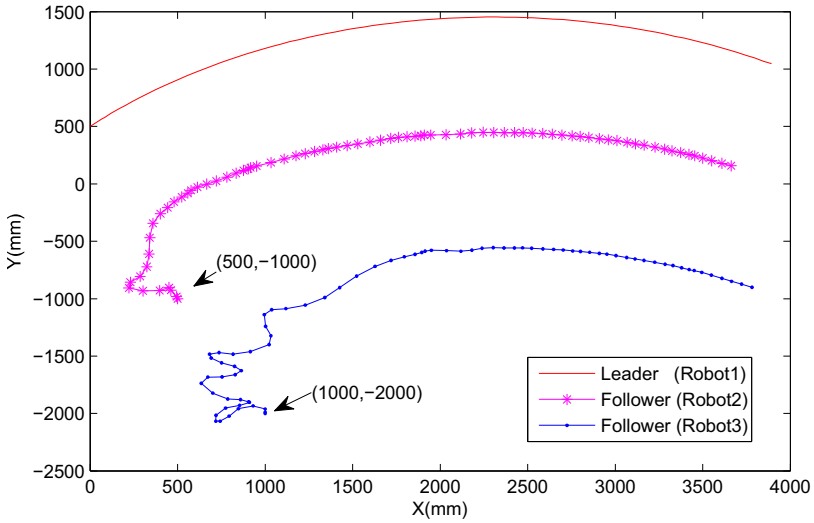


Figure 7.8: Curvilinear Trajectory of Pioneer Robots with Offset in r=20%, Converged in 27 Seconds

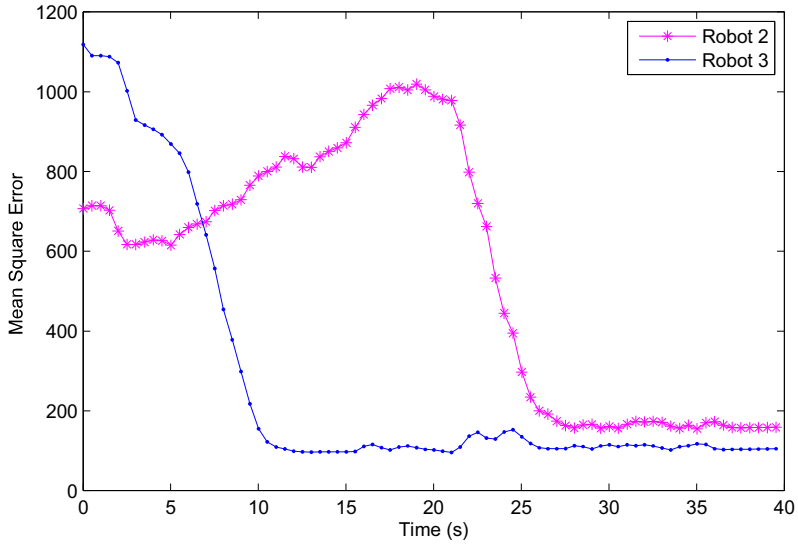


Figure 7.9: Mean Square Error of Curvilinear trajectory with r=20%, Converged in 27 Seconds

leader very tightly. Moreover, the position of robots in hardware are calculated on odometry, any uncertainties and error could increase the inaccuracy of the experiment results. The robot needs time to receive, process, and send data in the experiment rather than the simulation, which is not considered. All of these reasons could cause the uncertainty and inaccuracy in the hardware experiment. However, based on experiment results in both rectilinear and curvilinear examples with consideration of the limitation of the hardware, the hardware experiment certainly verified the validation of simulation results in Chapter 6.

Chapter 8

Conclusions and Future Works

This chapter summarizes any founds and results of this thesis and suggests developments to be pursued in the future.

8.1 Conclusions

In this thesis, a novel consensus algorithm or protocol for multi-agent systems (MAS) in the event of communication link failure over the network was developed and tested. The parametric study work shows that the algorithm provides stable results by investigating four cases; 1) effect of data loss rate, r; 2) effect of communication weight, ω_{ij}; 3) effect of initial values; and 4) effect of sampling time; Based on the results of simulation, within $r = 50\%$, it proves that the controller is more efficient. The maximum of lata loss rate for this controller determined is 95%. The higher weight distribution shortens the consensus time until the weight reaches to 1. However, the consensus time is not proportionately decreased in scale with the corresponding weight. The larger initial values extends the consensus time. However,the initial values only slightly affect the consensus ability. Since each sampling time is required with a new controller gain in order to satisfy the LMI in Theorem 1, the results show that the consensus ability is not only affected by sampling time, but also affected by corresponding new controller gains. The speed and the position are coupled, and the best controller gain is to balance the effect on both of speeds and positions. Therefore the sampling time cannot be neither too big nor too small. 0.1 second is a good comparably good sampling time for this proposed controller. A case with five agents was simulated in two examples, one is 20% of data loss rate and the other is 50% data loss rate. Both example prove that the effectiveness in a five agent examples. Theoretically, the controller can be used in any number of agents of multi-agent systems. Experimental tests were also conducted by using the algorithm to verify performance on a real world system. Two Pioneer 3-DX and a Pioneer 3-AT robots were used for a test platform. A leader following consensus was conducted in two cases, one is rectilinear, and the other is curvilinear with 20% of data loss rate. Both cases of experimental results showed the effectiveness and feasibility of the proposed approach.

8.2 Future Works

There are some interesting work extensions based on the results of this thesis:

- Consider both packet losses and delays in the communication channels of multi-agent systems. For packet losses, an alternative option can be used to govern the random data losses situation, such as Markov chains with time delays introduced in the controlling algorithm.
- Investigate other consensus problems instead of the leader following, such as a certain shape formation, an average consensus, and a cooperative construction.
- Research higher integrator dynamics.

Bibliography

[1] Wooldridge, M., "An Introduction to MultiAgent Systems", John Wiley and Sons, May 2009.

[2] Parnichkun, M.; Ozono, S., "Technical Note GSGM movement model for cooperative robots system Mechatronics", vol. 8, pp. 905-925, December 1998.

[3] Parker, L. E., "Current State of the Art in Distributed Autonomous Mobile Robotics", International Symposium on Distributed Autonomous Robotic Systems, 2001.

[4] Murray, R, M., "Recent Research in Cooperative Control of Multivehicle systems", Journal of Dynamic System, Measurement, and Control, vol. 129, pp. 571-583, September 2007.

[5] Chandler, P. R.; Pachter, M.; and Rasmussen, S., "UAV Cooperative Control", American Control Conference, IEEE, New York, pp. 50-55, 2001

[6] Mastellone, S.; Stipanovi, D.; Graunke, C.; Intlekofer, K.; and Spong, M., "Formation control and collision avoidance for multi-agent non-holonomic systems: theory and experiments", the International Journal of Robotics Research, pp. 107126, 2002

[7] Quick, D. "Global Hawk UAVs fly in close formation as part of aerial refueling program", October 2012 http://www.gizmag.com/global-hawk-uav-formation/24475.

[8] Murray, Richard M., "Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems", Society for Industrial and Applied Mathematics , Philadelphia, PA, 2002.

[9] Ren, W.; and Beard,R. W., "Distributed consensus in multi-vehicle cooperative control: theory and applications", London : Springer, 2008.

[10] Lafferriere, G.; Caughman, J.; and Williams, A., "Graph theoretic methods in the stability of vehicle formations", American Control Conference, vol. 4, pp. 3729-3734, June 2004.

[11] Wu, B.; Wang, D. W.; and Poh, E. K., "Decentralized control for satellite formation using local relative measurements only", Control and Automation , 2010 8th IEEE International Conference on, pp. 661-666, June 2010.

[12] Fax, J. A.; and Murray, R. M., "Information flow and cooperative control of vehicle formations", IEEE Trans. on Automatic Control, vol. 49, pp. 1465-1476, September 2004.

[13] Jin, J. G.; and Zheng, Y. F., "Consensus of multi-agent system under directed network: A matrix analysis approach", Control and Automation, IEEE International Conference on, pp. 280-284, 9-11 December 2009.

[14] Liu, Y.; Passino, K.M.; and Polycarpou, M., "Stability analysis of one-dimensional asynchronous swarms", Automatic Control, IEEE Transactions on, vol. 48, no. 10, pp. 1848- 1854, October 2003.

[15] Jing, J. D; and Zheng, Y. F., "Collective behavior of second-order multi-agent system in directed network", Control and Automation, 2010 8th IEEE International Conference, pp. 376-381, June 2010.

[16] Ren, W.; and Atkins, E., "Second-order Consensus Protocols in Multiple Vehicle Systems with Local Interactions", AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, August 2005.

[17] Wieland, P.; Kim, J. S.; Scheu, H.; and Allgower, F., "On Consensus in Multi-Agent Systems with Linear High-Order Agents", The International Federation of Automatic Control, Seoul, Korea, July 2008.

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

[19] Carpenter,J. R., "Decentralized control of satellite formations", International Journal of Robust and Nonlinear Control, vol. 12, pp. 141-161, 2002.

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

[21] Egerstedt, M.; Hu, X.; Stotsky, A., "Control of mobile platforms using a virtual vehicle approach", Automatic Control, IEEE Transactions on, vol. 46, no. 11, pp. 1777-1782, November 2001.

[22] Tabuada, P.; Pappas, G. J.; Lima, P., "Feasible Formations of Multi-Agent Systems", the American Control Conference Arlington, VA, June 2001.

[23] Ren, W.; and Beard, R.W., "Consensus seeking in multiagent systems under dynamically changing interaction topologies", IEEE Transactions on Automatic Control, vol. 50, no. 5, pp. 655- 661, May 2005.

[24] Cao, M.; Morse, A. S.; and Anderson, B. D. O., "Reaching a consensus in a dynamically changing environment, a graphical approach", Journal of Control and Optimiz, vol. 47, no. 2, pp. 575-600, 2008.

[25] Ferrari-Trecate, G.; Galbusera, L.; Marciandi, M.P.E.; and Scattolini, R., "A model predictive control scheme for consensus in multi-agent systems with single-integrator dynamics and input constraints", Decision and Control, 2007 46th IEEE Conference on, pp. 1492-1497, 12-14 December 2007.

[26] Zheng, Y. F., "Consensus of Dynamical Agents in Time-Varying Networks", 17th World Congress of the International Federation of Automatic Control Seoul, Korea, July 2008

[27] Zheng, Y. F; Shao, H. B.; and Pan, W. Y., "Consensus problem of double-integrator dynamics system under time-varying networks", Asian Control Conference, pp.343-348, August 2009.

[28] Yu, M.; Wang, L.; Chu, T. G.; and Hao, F., "An LMI approach to networked control systems with data packet dropout and transmission delays", Decision and Control, 2004 43rd IEEE Conference on, vol. 4, pp. 3545- 3550, vol. 4, 14-17 December 2004.

[29] Ridwan, W.; and Bambang, R. T., "H∞ controller synthesis for Networked Control Systems with time delay system approach", Electrical Engineering and Informatics, International Conference on, pp. 1-5, July 2011.

[30] Ren, W.; and Beard, R. W., "Consensus seeking in multiagent systems under dynamically changing interaction topologies", IEEE Transactions on Automatic Control, vol. 50, no. 5, pp. 655- 661, May 2005.

[31] Moreau, L., "Stability of multi-agent systems with time-dependent communication links", IEEE Trans. on Automatic Control, vol. 50, pp. 169-182, February 2005.

[32] Jadbabaie, A., "On coordination strategies for mobile agents with changing nearest neighbor sets", IEEE Mediterranian Conference on Control and Automation, Rhodes, Greece, June 2003.

[33] Mei,J; Ren, W.; and Ma, G. F., "Distributed Coordinated Tracking With a Dynamic Leader for Multiple Euler-Lagrange Systems", Automatic Control, IEEE Transactions on, vol. 56, no. 6, pp. 1415-1421, June 2011.

[34] Nu, E.; Ortega, R.; Basaz, L.; and Hill, D., "Synchronization of Networks of Nonidentical Euler-Lagrange Systems With Uncertain Parameters and Communication Delays", IEEE Transactions on Automatic Control, vol. 56, no. 4, pp. 935-941, April 2011.

[35] Zhang, Y.; and Tian, Y. P., "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, April 2010.

[36] Parker, L. E., "Designing control laws for cooperative agent teams", Robotics and Automation, 1993 IEEE International Conference on , vol. 3, pp. 582-587, 2-6 May 1993.

[37] Lavretsky, E., "F/A-18 autonomous formation flight control system design", AIAA Guidance, Navigation, and Control and Co-located Conferences, pp. 2002-4757, 2002.

[38] Welsh, T. U.S. Navy photo by Mass Communication Specialist 3rd Class http://www.navy.mil

[39] Murray, R. M., "Recent Research in Cooperative Control of Multivehicle Systems", Journal of Dynamic Systems, Measurement, and Control vol. 129, no. 5, pp. 571, September 2007.

[40] Monterey Bay Aquarium Research Institute, Autonomous Ocean Sampling Network, 2006. http://www.mbari.org/aosn

[41] Pasqualetti, F.; Franchi, A.; and Bullo, F., "On Optimal Cooperative Patrolling", 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, pp. 7153-7158, 2010.

[42] Technologynewhere, "Intelligent transportation system", May 2010. http://technologynewhere.wordpress.com/2010/05/12/intelligent-transportation-system

[43] California Partners for Advanced Transit and Highways, Department of Transportation, 2006. http://www.path.berkeley.edu

[44] Nguyen-duc, M.; Guessoum, z.; Marin, O.; Perrot, J. F.; and Briot, J. P., "A multi-agent approach to reliable air traffic control", 2nd International Symposium on Agent Based Modeling and Simulation, 2008.

[45] Hill, J. C.; Archibald, J. K.; Stirling, W. C.; and Frost, R. L., "A multi-agent system architecture for distributed air traffic control", AIAA Guidance, Navigation and Control Conference, 2005.

[46] Royle, G.; and Godsil, C., "Algebraic Graph Theory", New York: Springer Graduate Texts in Mathematics, no. 207, 2001.

[47] Mehran, M.; and Egerstedt, M., "Graph Theoretic Methods in Multiagent Networks", Princeton: Princeton UP, 2010.

[48] Ren, W.; and Beard, R. W., "Distributed consensus in multi-vehicle cooperative control: theory and applications", London : Springer, 2008.

[49] Savkin, A. V., "Coordinated collective motion of groups of autonomous mobile robots: Analysis of Vicsek's model", IEEE Trans. Autom. Control, vol. 49, no. 6, pp. 981-982, June 2004.

[50] Sheng, L., "Lyapunov-based Control Approaches for Networked Single and Multi-agent Systems with Communication Constraints", Thesis. Dalhousie University, 2010.

[51] http://forum.processing.org/topic/markov-chain, 2012.

[52] Roger, H.; and Charles J., "Topics in Matrix Analysis", Cambridge University Press. 1999.

[53] Cooper, D., "Practical Process Control - Proven Methods and Best Practices for Automatic PID Control", 2008. http://www.controlguru.com .

[54] Cyberbotics, "Pioneer 3-DX, an all-purpose base, used for research and applications", December 2012. http://www.cyberbotics.com/guide/section8.3.php.

[55] Long, S., "Lyapunov-based Control Approaches for Networked Single and Multi-agent Systems with Communication Constraints", Doctral Thesis, 2010.

[56] Jian, W.; and Shi, Y., "Average consensus in multi-agent systems with time-varying delays and packet losses", American Control Conference, Montreal, Canada, June 2012.

Appendix A

Operations Manual

Simulation

1. Set the sampling time, Ts at c2d.m file, then matrix A and B will be given after running of this .m file.

2. According to matrix A and B with a desired data loss rate,r, a controller gain K will be calculated by lmi.m file

3. Run robot3.m file with r and k obtained from step 2. It is for parametric studies.

4. Run robot3xy.m file with r and k obtained from step 2. It is for Pioneer robot simulations.

Experiment

1. Turn on DIR-615 D-link Router to create a local wireless network.

2. Connect each laptop with corresponding Pioneer robot with RS-232 Serial Cable by ensuring connect in the COM 1, labeled at each laptop.

3. Open service&client folder. For DELL Inspiron, open service project file. For TOSHIBA Satellite Pro, open client1 project file. For TOSHIBA Satellite, open client1.

4. Set a data loss rate in line $(rand()\%100 >= 1 - r)$, data loss rate is $0.0r\%$.

5. Set operation time in line (int interval = time) in ms.

6. Place each robot at appropriate starting spots. Make sure each robot has at least 1 m of safe distance.

7. Run service first, and wait from command window, and then run client1, and client2.

8. Save and load datalog.txt, datalog 1.txt, and datalog 2.txt in plot robot trajectory.m for plotting robots' trajectories.

9. For a rectilinear case, set (double vl = 100; double vr = 100;).

10. For a curvilinear case, set (double vl = 100; double vr = 130;).

Appendix B

Author's Publication List

1. Sheng, L,; Pan, Y. J.; and Gong, X., "Consensus Formation Control for a Class of Networked Multiple Mobile Robot Systems", Journal of Control Science and Engineering, Vol. 2012, March 2012. (Accepted)

2. Sheng, L.; Pan, Y. J.; Gong, X.; and Wu, J., "Distributed Consensus Formation Control of Multiple Mobile Robots", 2011 International Conference on Control and Automation, Shanghai, China. (Accepted)

3. Scott, D.; Pan, Y. J.; and Gong, X., "Multi-Robot Distributed Control for Construction Tasks based on Intelligent Beacons", Control and Intelligent System, Paper No. 201-2441, 2013. (Accepted)

4. Gong, X.; Pan, Y. J.; and Li, J. N.; and Su, H., "Leader Following Consensus for Multi-agent Systems with Stochastic Packet Dropout", the 10th IEEE International Conference on Control and Automation , Hangzhou, China, 2013. (Accepted)