LYAPUNOV-BASED CONTROL APPROACHES FOR NETWORKED SINGLE AND MULTI-AGENT SYSTEMS WITH COMMUNICATION CONSTRAINTS

by

Long Sheng

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
November 2010

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 "LYAPUNOV-BASED CONTROL APPROACHES FOR NETWORKED SINGLE AND MULTI-AGENT SYSTEMS WITH COMMUNICATION CONSTRAINTS" by Long Sheng in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dated: November 25, 2010

External Examiner: ________________________________

Research Supervisor: ________________________________

Examining Committee: ________________________________

____

Departmental Representative: ____________________________

DALHOUSIE UNIVERSITY

DATE: November 25, 2010

AUTHOR: Long Sheng

TITLE: Lyapunov-based Control Approaches for Networked Single and Multi-agent Systems with Communication Constraints

DEPARTMENT OR SCHOOL: Department of Mechanical Engineering

DEGREE: PhD CONVOCATION: May YEAR: 2011

Signature of Author

To my parents and my wife.

Table of Contents

List of Figures

Abstract

Networked control systems (NCSs) are feedback control systems with the feedback control loops closed via network. The origin of the term NCSs is from industrial systems where the plant and controller are often connected through networks. The applications of NCSs cover a wide range of industries, for example, manufactory automation, domestic robots, aircraft, automobiles and tele-operations.

The research activities in NCSs are focused on the following three areas: control of networks, control over networks and multi-agent systems. Control of networks is mainly concerned with the problem of how to efficiently utilize the network resource by controlling and routing the network data flows. Control over networks is mainly concerned with the design of feedback control strategies of control systems in which signals are transmitted through unreliable communication links. Multi-agent systems deal with two problems: how the topology of the network connections between each component influences global control goals and how to design local control law describing the behavior of each individual to achieve the global control goal of the whole systems. The objective in this thesis is to deal with control over networks and multi-agent systems.

The most challenging problem in the control over networks field is that the unreliable communication channels can degrade system performance greatly. The main unreliable properties of networks are delays and packet loss. In order to deal with this problem, a Lyapunov-based method has been used to design the sampled-data stabilization control strategy for a networked single system by choosing proper delay and packet loss dependent Lyapunov functional candidates. Linear matrix inequality techniques have been used to find the sufficient and necessary conditions for the controller design. Furthermore, the consensus formation control problem of multiple robotic vehicle systems has been investigated. The consensus-based design scheme has been applied to the formation control of multiple wheeled mobile-robot group with a virtual leader. A novel delay-dependent Lyapunov functional candidate has been constructed to investigate the convergence of the system states. The proposed control strategy is experimentally implemented for multiple wheeled mobile robots under neighbor-to-neighbor information exchange with group communication delays involved. In conclusion, through the simulation results and experimental validations, the proposed new Lyapunov-based control methods can effectively deal with the networked control systems discussed in this thesis.

List of Abbreviations and Symbols

$\forall$: for all.

$\exists$: exists.

$\in$: belongs to.

$\subset$: a strict subset of.

$\subseteq$: a subset of.

$\rightarrow$: tends to.

$\sum$: summation.

max: maximum.

min: minimum.

$\Re^{m\times n}$: set of $m \times n$ real matrices.

$\rho(A)$: spectral radius of matrix A.

$\lambda(A)$: eigenvalue of matrix A.

$\lambda(A)_{max}$: the maximum eigenvalue of matrix A.

$\lambda(A)_{min}$: the minimum eigenvalue of matrix A.

$\Gamma(A)$: directed graph of matrix A.

$diag(A_1, \ldots, A_n)$: a block diagonal matrix with diagonal blocks A_1 to A_n.

$\mathbf{x}$: system state.

$\mathbf{y}$: system output.

$\mathbf{u}$: control input.

$\|\mathbf{x}\|$: norm of vector $\mathbf{x}$.

$\c{C}_p$: graph.

ν_p: node set of a graph.

ε_p: edge set of a graph.

L_p: $p \times p$ Laplacian matrix.

$NSCs$: networked control systems.

AUV: autonomous underwater vehicle.

GPS: global positioning system.

UAV: unmanned air vehicle.

$P3$: pioneer 3 mobile robot.

Acknowledgments

I am heartily thankful to my supervisor, Dr.Ya-Jun Pan, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. I would like to thank Dr. Robert Bauer and Dr. Guy Kember who are my committee members for their time and efforts put in improving my thesis. I would like to show my gratitude to Dr. Simon X. Yang who would like to become my external examiner and spend his time on reading and improving my thesis. I also would like to thank NSERC and CFI Canada for the funding support and all kind help from ACM group.

Finally I would like to give my deepest gratitude to my parents and my wife Xiaocan Zhang. I cannot make it without the concern, help and support provided by them.

Chapter 1

Introduction

1.1 Research Motivation

Networked control systems(NCSs) are feedback control systems with control loops closed through real time network communication channels. In recent decades NCSs have gained increasing attention and have become one of the main research focus in the field of controls as well as in industrial applications. Gaining attention is due to their advantages they have in reducing the complexity of wiring connections, decreasing the costs of cables and power, simplifying the installation and maintenance of the whole system and increasing the reliability. As a result of extensive research and development, several network protocols for industrial control have been realized, such as Control Area Network developed by Robert Bosch Company in 1983. NCSs also have been used in many other industrial control applications, such as automobiles, manufacturing plants and aircraft. However, insertion of network communication channels make the analysis and design of NCSs complicated and hence raises new interesting and challenging problems such as quantization, time delays and packet losses. Traditional control theory framework is about the study of interconnected dynamical systems linked via ideal channels, whereas for NCSs we need to study the transmission of information over imperfect channels. As a result, a combination of these two frameworks is needed to model NCSs which means the traditional control theories must be re-evaluated before applying them to NCSs. These new interesting and challenging problems regarding NCSs are the motivation of the works presented in this thesis.

1.2 Networked Control Systems Overview

For Networked control systems there are two main configurations called “Direct Structure” and “Hierarchical Structure” respectively. The direct structure of NCSs is composed of a controller and a remote system with a physical plant, sensor and actuators. As shown in Fig.1.1, the controller and the plant are located at different places and are directly connected by the network to perform remote closed-loop feedback control. Before being sent to the remote plant via the network, the control signals will first be encapsulated into packets by the controller. Similarly the system states of the plant measured by the sensors will be put into packets before being sent back to the controller. Hierarchical structure is composed of a main controller and a remote closed-loop system as shown in Fig.1.2. The task of the main controller is to send the packeted reference signals to the remote closed-loop system through the network. By processing the reference signals the remote system can perform local closed-loop feedback control and send the system outputs back to the main controller. Teleoperation control is a famous application of this structure.

Main activities in the area of NCSs generally contain three major fields [1]: 1) control of networks, which is mainly concerned with developing new technology to

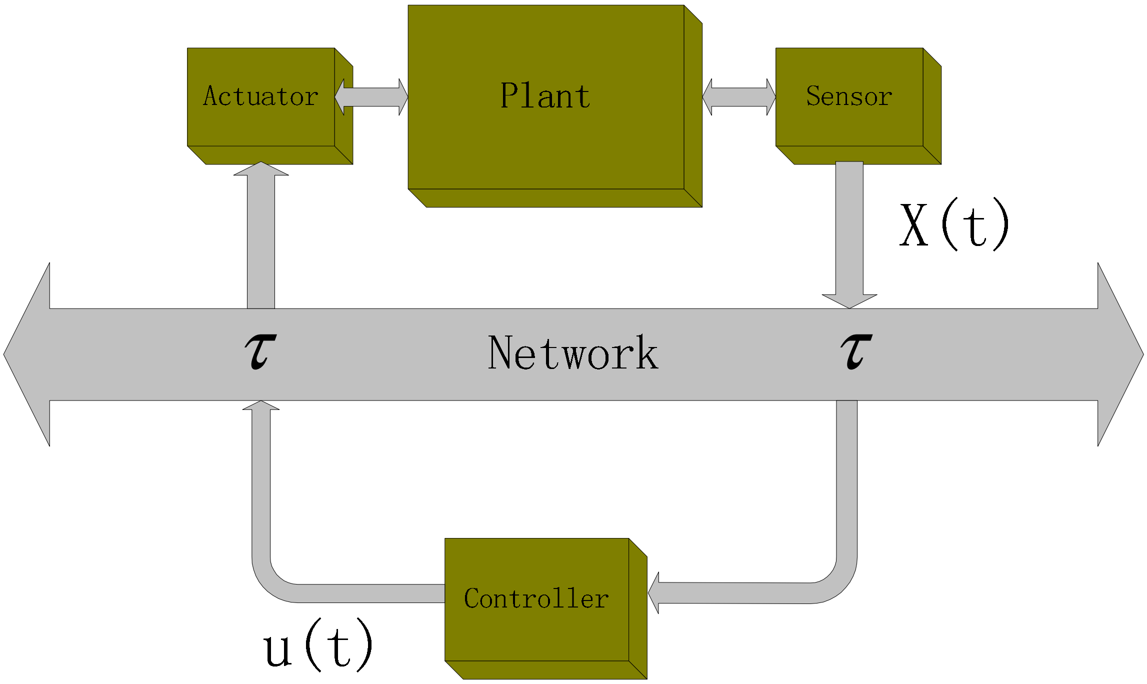


Figure 1.1. Framework of networked control system: direct structure

improve the performance to a network data flow so that the network resources can be efficiently and fairly utilized; 2) control over networks, which deals with designing feedback control strategies to stabilize the remote systems with network induced communication constraints; 3) multi-agent system, which is concern with the problem that how the network architecture and interactions between network components influence global control goals and how local laws describing the behavior of the individual agents influence the global behavior of the whole system.

1.2.1 Control of Networks

A basic network system can be treated as a feedback mechanism in which the data transfer through the internet with predefined internet protocol is controlled. Control of networks is a very large research area. The basic topic in control of networks include controlling congestion across network links, routing the data flow in the network

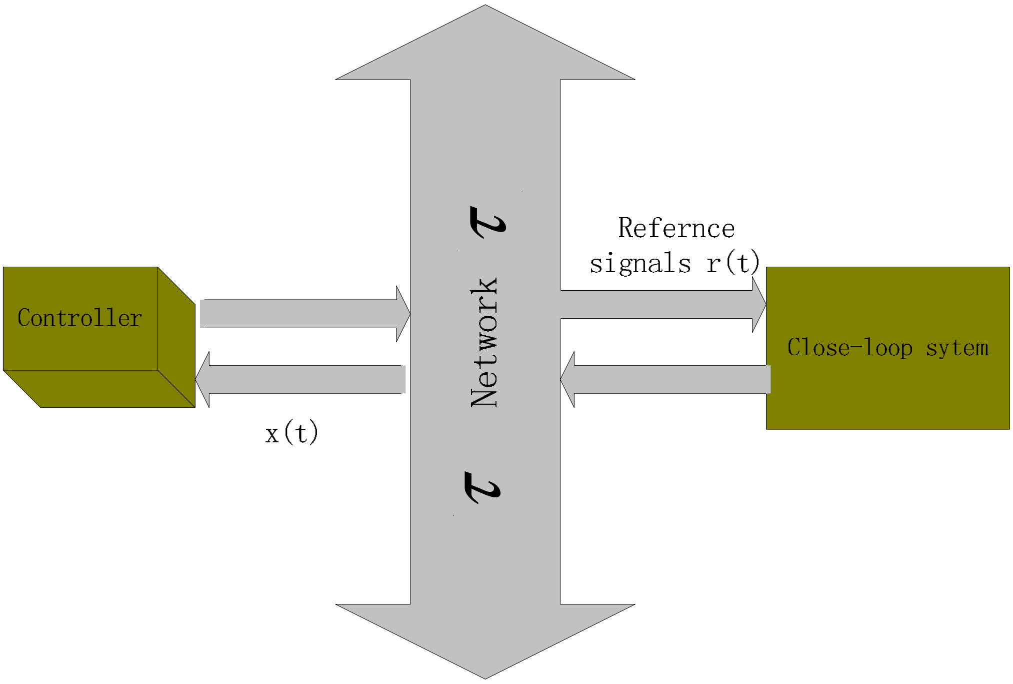


Figure 1.2. Framework of networked control system: hierarchical structure

channel, power control, and resource allocation problems.

Some specific challenges in this field are related to issues, such as how to deal with the extremely large scale of the system; how to make local decisions quickly only by using local information; how to tackle the uncertainty and variation in the network caused by the unpredictable changes of network topology, transmission channel characteristics, traffic demand and available resources and how to deal with the various traffic characteristics of the network.

Those challenges make it extremely difficult to model and analyze the control of network problem. Computing, storage and transmission must be managed in this complicated environment with the given available resources. Researchers in this area pay more attention on the investigation of mathematical theory that can offer possible improvements in this field.

The research works in [2] [3] show significant progress in the theoretical studies of network congestion control by explicitly modelling the congestion measurement and posing the network flow control as an optimization problem. The main objective of this work is to find the optimal conditions to maximize the total resource utility and solve the rate control problem in a decentralized manner. Recent works related to this area focus on developing mathematical models for flow control under diverse internet protocols [4]; developing scalable and distributed optimization algorithms for the control systems [5]- [8] and investigating the effects of time delays and nonlinearities in the network flow models [9].

1.2.2 Control over Networks

For control over networks, the key issue is how to reduce the effects caused by the characteristic features of the network communication channel in the feedback control strategy. Those drawbacks include channel band limit, time delay and packet dropout.

A network communication channel has band limit since any network channel can only carry a finite amount of information per unit of time. In many applications, this limitation results in significant constraints for the operation of NCSs, such as in unmanned air vehicles, sensor networks, underwater vehicles and large arrays of micro actuators and sensors.

Because the data has been transferred via a network, there is a network induced delay between the controller and the remote system in addition to the controller processing delay. The network induced delays in the control loop is shown in Fig.1.1, where u and x are the control input signal and system state respectively, and τ is the

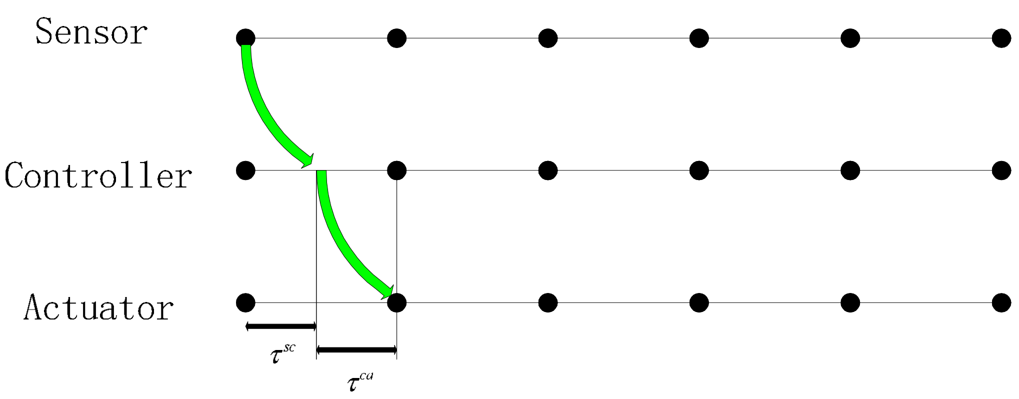


Figure 1.3. Timing diagram of network time delay propagations

time delay value. Fig.1.3 shows the corresponding timing diagram of network time delay propagations. From the data transferring direction, network time delays of NCSs can be categorized as the sensor-to-controller delay τ^{sc} and the controller-to-actuator delay τ^{ca}. Existence of network time delays in NCSs is a significant challenging problem which is needed to be tackled. First it can degrade system performance of a NCS. Second, which is even worse, it can destabilize the system by reducing the system stability margin. So reducing the effects caused by network induced time delay is one of the main tasks regarding NCSs.

In order to deal with time delays in NCSs, in some recent works predictor-based control approach has been applied. The time delays in both constant and variable form have been modelled by dynamical ordinary differential equations to predict the delayed states in NCSs. A successful designed predictor can efficiently reduce the effects of time delays in this way. Another approach is concerned with applying robust control theory into NCSs to deal with time delays. In robust control approach, time delays are translated into uncertainties in the system, then the robust stability conditions for the NCSs are investigated and the bound of the time delay is estimated.

Another significant nature of NCSs is the possibility that data may be lost during transmission through the network. Typically, packet loss can cause transmission errors in physical network links, which happens more in wireless than wired network, and buffer overflows due to congestion in the network communication channel. Long time delays may cause packet dropout when the receiver ignores outdated arrivals. Some reliable network transmission protocols, such as TCP/IP protocol, guarantee that the packets dropped during transmission will be retransferred again. However, these protocols are not appropriate for NCSs since the retransmission of outdated data is not very useful. Similar to network induced transmission delay, reducing the effects of packet loss is another essential challenging task regarding NCSs.

In recent works, the Bernoulli packet loss effect between the sensor and the estimator has been considered in the modelling of packet loss. In some situations, packets received after the bound of time delays are considered as packet loss, so packet losses can be modeled as additional time delays in some recent works [21].

1.2.3 Multi-agent Systems

The current key issues associated with multi-agent systems deal with consensus based control. For example, for multiple mobile autonomous robots, teams of autonomous agents coupled via network, are required to implement some global control aim. Those team agents are expected to become economically feasible and complete a variety of spatially distributed sensing tasks, such as search, rescue, surveillance, environmental monitoring and exploration. Performing team tasks requires all mobile autonomous agents to be able to coordinate and cooperate with each other. Such cooperative tasks for teams of autonomous agents include, for example, searching a region of interest, moving with special team formation and converging to a common point.

The challenging task for multi-agent system is concerned with the design of distributed control strategies that can be applied into the multi-agent cooperative control applications.

1.3 Applications

In this section some applications regarding the NCSs are given. All the examples are taken from recent existent works. The whole section is divided into three parts: control of networks, control over networks and multi-agent systems.

1.3.1 Application of Control of Networks

Recently, significant progress in the theoretical investigation of network congestion control has been made. In this progress, the congestion measure signal feedback to the source has been modelled. The network flow control has been posed as an optimization problem where the objective is to maximize the total source utility. See the following example proposed by [3].

In this work, an optimization approach to flow control has been proposed. The objective is to maximize the aggregate source utility over their transmission rates. The logic topology of a network is shown in Fig.1.4. Consider a network that consists of a set $L = \{1, \ldots, L\}$ of unidirectional links of capacity c_l, $l \in L$. The network is shared by a set $S = \{1, \ldots, S\}$ of sources. Source S is characterized by four parameters: the first one $L(s) \subseteq L$ is a set of links that source S uses; the second one U_s is a utility function; the other two $m_s \geq 0$ and $M_s < \infty$ are minimum and maximum transmission rates, respectively, required by source S. Source S attains a utility $U_s(x_s)$ when it transmits at rate x_s that satisfies $m_s \leq x_s \leq M_s$. For each link l let $S(l) = \{s \in S | l \in L(s)\}$ be the set of sources that use link l.

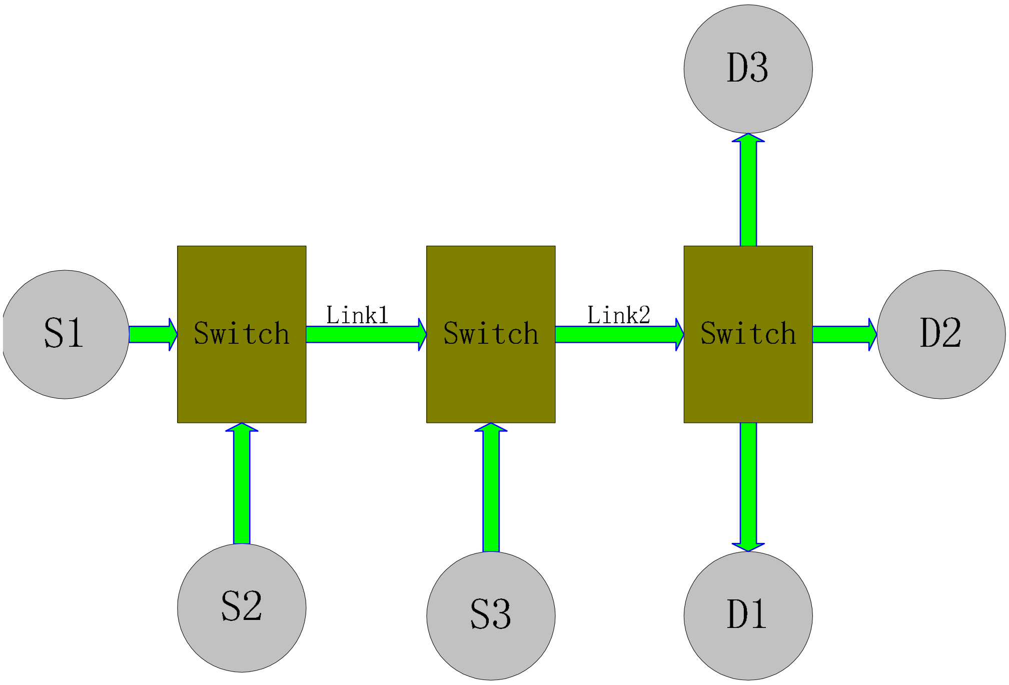


Figure 1.4. Logical topology. Source S_1 transmits to destination D_i, $i = 1, 2, 3$

The objective of this work is to choose source rates $x = (x_s, s \in S)$ so that:

$$\max_{x_s \in I_s} \sum_s U_s(x_s) \tag{1.1}$$

subject to

$$\sum_{s \in S(l)} x_s \leq c_l, l = 1, \ldots, L. \tag{1.2}$$

Solving this problem directly requires coordination among possibly all sources and is impractical in real networks. The key to a distributed and decentralized solution in this work is to solve the dual problem by using gradient projection algorithm.

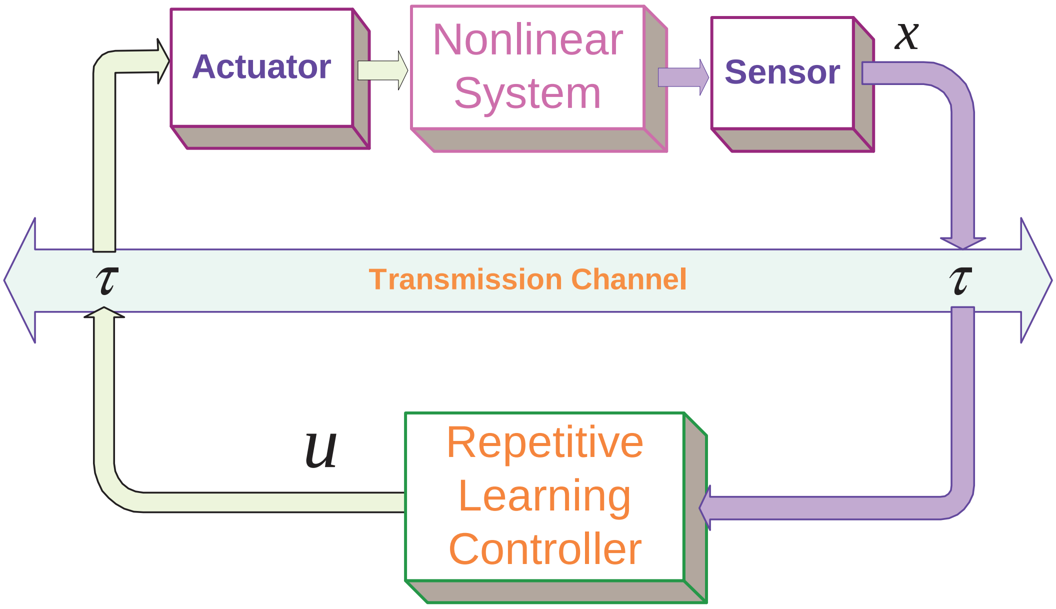


Figure 1.5. Diagram of networked RLC system

1.3.2 Application of Control over Networks

A predictor-based networked repetitive learning control (RLC) approach has been introduced in [10].

As shown in Fig.1.5, a repetitive learning controller is designed to repetitively stabilize the plant located in a distant place via network communication channel. Since there is network induced time delay, the system performance is affected by the time delay. In order to reduce the effect caused by the time delay, a predictor is designed on the controller side to predict the system states. The diagram of the new system is shown in Fig.1.6. In this work, the sufficient conditions for predictor design are derived. If the predictor is designed properly and meets the sufficient conditions, it can predict the system states accurately. With accurate predicted information, the repetitive learning controller can calculate and send correct control signals to the plant. The effects caused by network induced time delay is, therefore, reduced.

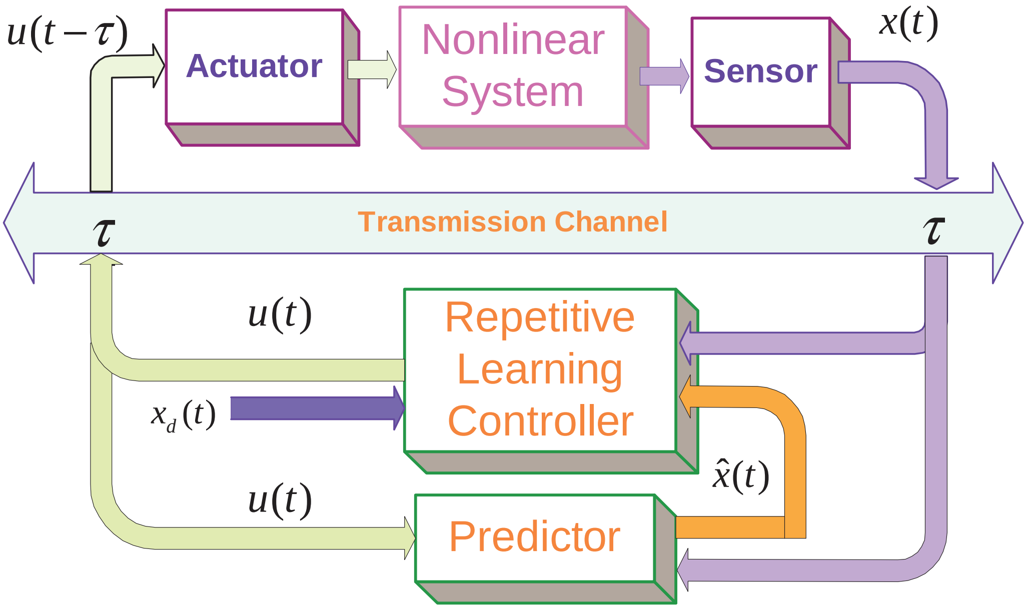


Figure 1.6. Diagram of predictor-based RLC system

1.3.3 Application of Multi-agent Systems

A novel distributed control scheme has been applied to formation control of multiple robotic vehicles in [11]. In this work, several mobile robots are considered to move on a plane. The objective is to make the robotic vehicle group form some formations.

The basic idea is from the consideration of a simple case where two mobile robots are free to move on one-dimensional space as shown in Fig.1.7. These two robots, labelled as R_1 and R_2 respectively, try to make some relative distance between them. Mathematical formula is designed for R_1 to guide R_1 to make its relative distance to R_2 d_1. For R_2, another mathematical formula is designed to guide R_2 to make the relative distance to R_1 d_2, $d_1 > d_2$. When the current relative distance d_0 is longer than d_1, both of them try to get close to each other. When d_0 is shorter than d_2, then both of them keep away from each other. Finally d_0 is kept to be shorter than d_1 and longer than d_2 and both of them keep moving at some velocity determined by

d_1 and d_2. d_1 and d_2 are used as the control input in the distributed control scheme designed for both of robots to deal with physical features of this robotic system, such as stability of the robots and the relative distance between them quantitatively.

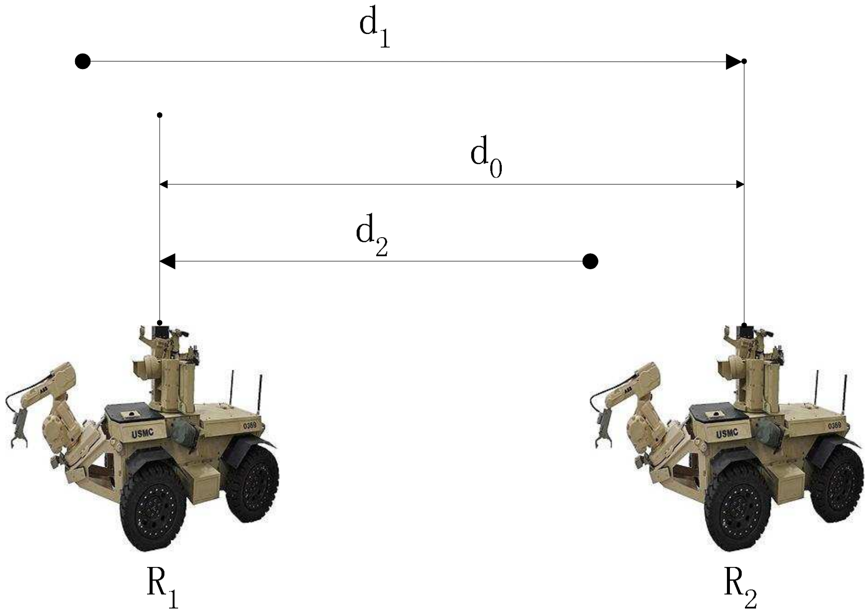


Figure 1.7. Relative distance between mobile robots

The simple case described above is expanded into a more general case where several (n) mobile robots are free to move on plane. They try to make group formations on this two dimensional space. Each robot is labelled as R_i and a two dimensional vector (d_{xi}, d_{yi}) called the "formation vector" are given. This "formation vector" physically specifies the relative position of R_i to other robots sensed by R_i on the plane. A distributed control scheme for making formations of mobile robot groups is proposed. Each robot in the group can sense and control its relative position to other robots. The motion of each robot is determined by rules given as mathematical formulas including the formation vector.

The robot groups are considered in two types of configuration which are called as

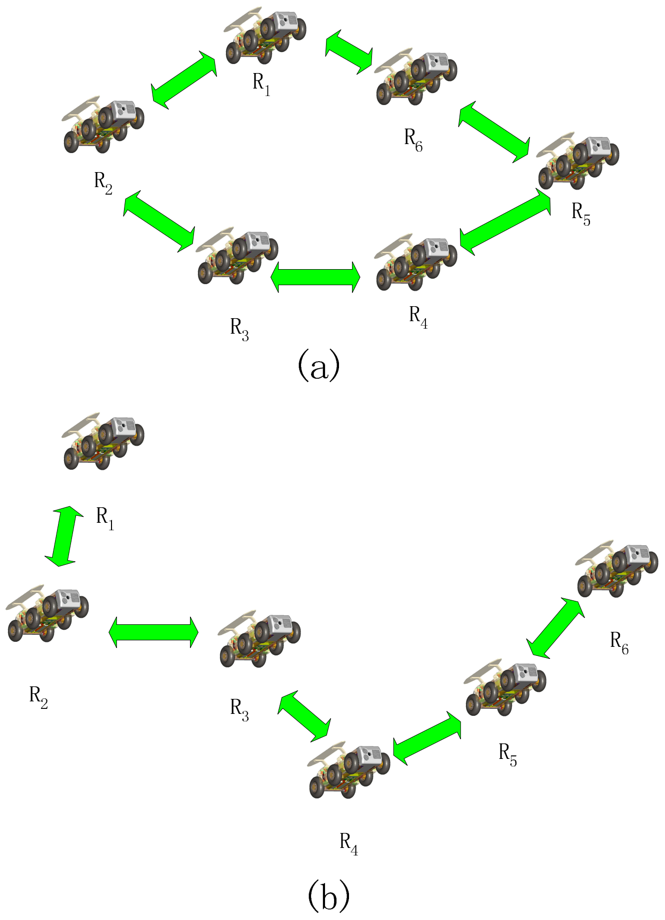


Figure 1.8. Mobile group configuration: (*a*) closed loop; (*b*) open loop

closed and open chain configuration respectively. These two configurations are shown in Fig.1.8. The proposed distributed control scheme is applicable in cases where the groups have other configurations. However, considering only these two configurations is practical because groups in other configurations easily get tangled.

1.4 Thesis Objectives and Contributions

The objectives of this thesis can be summarized as follows: 1) develop feedback control strategies to stabilize a class of linear systems controlled by the remote controller through network with natural communication constraint; 2) develop consensus based design methodologies for the distributed multi-agent formation control with group communication delays; 3) apply the consensus based design methodologies to the distributed formation control problem for a team of multiple wheeled differentially driven mobile robots.

The first contribution of this thesis is the design of a stabilization control approach for a class of linear NCSs with natural networked induced communication constraints. The networked induced communication constraints have been monitored, modeled and considered in designing the feedback stabilization control strategies. Networked induced communication constraint-dependant Lyapunov-based methodologies have been proposed. Linear matrix inequality (LMI) techniques has been utilized to find the sufficient conditions for the stabilization controller design. The second contribution is the design of consensus based formation control strategy for distributed cooperation control problems of multi-agent system with group communication delay. A novel consensus tracking control algorithm has been posed. The convergence of the tracking errors has been analyzed by utilizing delay-dependent Lyapunov approach. Comparison between multiple agent group performance with and without group communication is posed by simulation works. This comparison addresses the importance of information exchange in multi-agent system. The third contribution is applying the consensus based design methodology to the distributed formation control of the multiple mobile robot vehicle group with group communication delay. The kinematic model of the nonholonomic differentially driven mobile robots has been studied and analyzed. A new delay-dependent multiple Lyaponov-based methodology has been developed to analyze the convergence of the tracking errors. LMI technology has been used to provide sufficient conditions of the controller design. An experimental platform for distributed multi-vehicle cooperative control has been built. Experiments have been done to test the feasibility of the proposed approaches.

1.5 Thesis Outline

The thesis outline is structured as follows: In Chapter 2, a literature review regarding control over network and multi-agent system is accomplished and advances in NCSs research area are introduced. In Chapter 3 the important issues of network communication channels such as delays and packet losses are introduced. In Chapter 4, the design of a stabilization control approach for sampled-data NCSs is introduced. In Chapter 5, the design of the consensus based formation control strategy for distributed cooperation control problems of multi-agent system with group communication delay is demonstrated. In Chapter 6, the consensus based design methodology for the distributed formation control of the multiple mobile robotic vehicle group with group communication delay is presented. In Chapter 7, the simulation works are proceeded to test the feasibility of the proposed approaches in the previous three chapters. In Chapter 8, the experimental results regarding the main works represented in Chapter 4 and Chapter 6 are shown. Chapter 9 presents the conclusions and future work.

Chapter 2

Advances in Networked Control Systems

2.1 Introduction

In this chapter, the advance achievements regarding control over network and multi-agent system have been reviewed. In control over network part, improved techniques, determination of NCS stability and control synthesis are introduced. For multi-agent system area, consensus-based cooperative control synthesis are analyzed and reviewed.

2.2 Challenges in NCSs

Network control systems (NCSs) are spatially distributed systems in which the communication between sensors, actuators and controllers occurs through a shared band-limited digital communication network [12]. Fig.2.1 shows the general architecture of NCSs.

NCSs lie at the intersection of control and communication theories. Traditional control theory focuses on the the traditional dynamic systems connected via “ideal channels”. Communication theory focuses on data transmission via practical or imperfect channels. The studies of NCSs is a combination of these two frameworks.

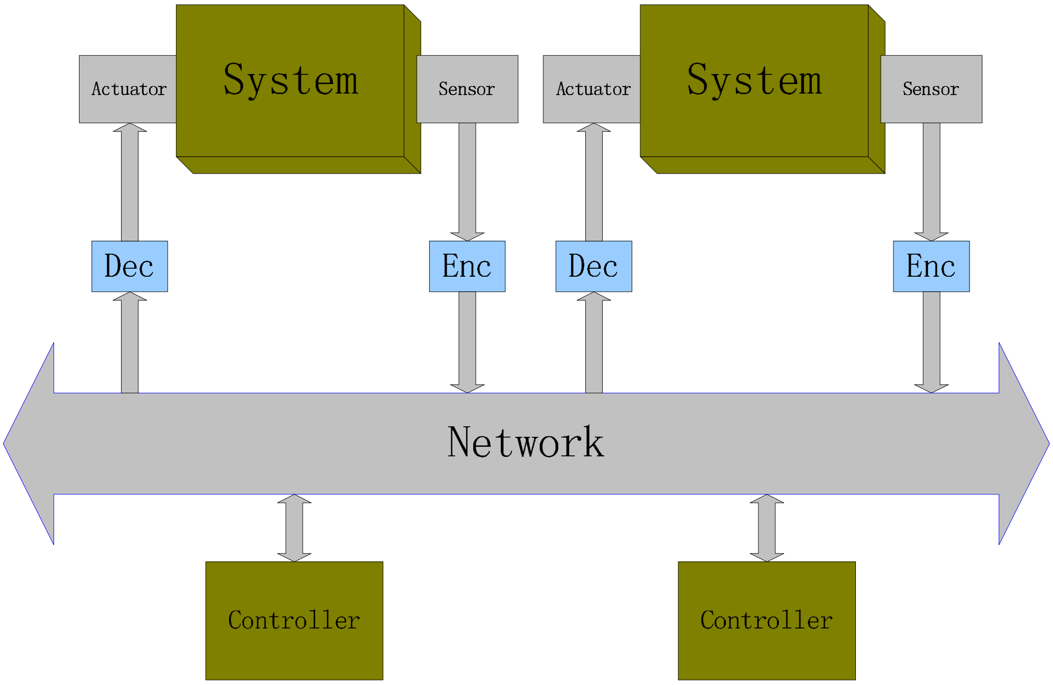


Figure 2.1. General NCS architecture

This combination brings NCSs unique challenges which are different with traditional feedback control systems such as:

1) Band-Limited Channels; band-limit is due to the amount limitation of information per unit that the communication network can carry. In some applications of NCSs, significant constraints on the operation of NCSs are posed by this limitation. These applications include unmanned air vehicles (UAVs), power starved vehicles, long-endurance energy limited systems, underwater vehicles and large arrays of micro actuators and sensors. Significant research efforts have been devoted to the problem of determining the maximum bit rate that a communication channel can carry and the minimum bit rate that is needed to stabilize a linear system through feedback over a finite capacity channel. Some recent work are also related to solving the finite capacity stabilization problem for nonlinear systems and for linear systems with unknown

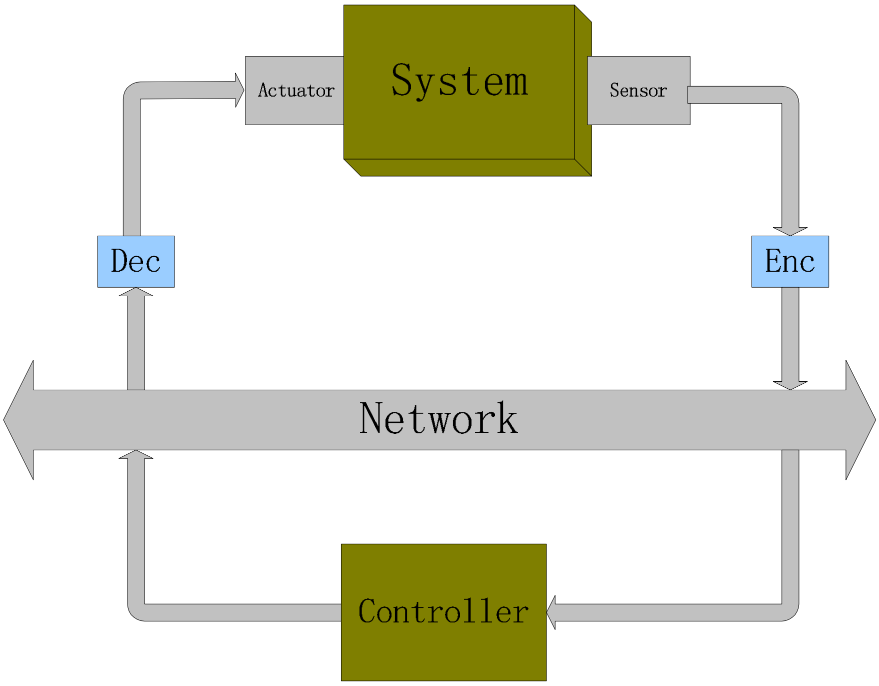


Figure 2.2. Signal loop NCS architecture

parameters. 2) Sampling and Time Delay; before being transmitted over a network, the signal must be sampled and encoded in a digital format. After the transmission the data must be decoded at the receiver side. This process is significantly different from the usual periodic sampling in digital control. The overall network induced time delay can be highly variable because it depends on highly variable network conditions such as congestion and channel quality. Significant research efforts are devoted to the problem of stabilization control strategy design and characterizing the maximum upper bound on the sampling interval that can guarantee the stability of NCSs. 3) Packet Dropout; data dropout is another significant difference between NCSs and traditional feedback control. Many reasons can cause a data dropout happen, such as transmission errors in physical network links or buffer overflows due to congestion. Packet reordering due to lone transmission delays sometimes are treated as a packet dropout if the receiver discards outdated arrivals. Recently the stabilization problems of NCSs with data dropouts are more and more considered in the control strategy design. 4) System Architecture; the general architecture of a NCS is shown by Fig.2.1. The main tasks of encoders are mapping measurements of continuous time signals into streams of "symbols" that can be transmitted across the network. Two problems are tackled by encoders: when to sample a continuous time signal for transmission and what to send through the network. Conversely the main tasks of decoders is mapping the streams of symbols received from the network into continuous actuation signals. Recently simpler structures are considered in most of the research on NCSs. For example, some controllers may be collocated with the corresponding actuators. The signal feedback loop as shown in Fig.2.2 is also often considered. Although it is simpler than the one shown in Fig.2.1, the main challenges of NCSs, such as bandwidth, time delays and packet dropouts, are still needed to be considered in this architecture.

2.3 Control Over Networks

The research and developments of feedback control system with data transmission supported by a shared communication network have a long history. Principle data networks such as ALOHAnet and ARPAnet, have been developed around $30 - 40$ years ago. Network provides several benefits on linking data points like computers and enable remote data transfers among different nodes conveniently.

Control over networks is one of the main activities in NCSs area. It is mainly concerned with the design of feedback control strategies adapted to control systems in which data packets are transferred through unreliable communication links. The study of this area raises the following new interesting and challenging problems: *i*) Data transmission through a communication network unavoidably introduces time delays in the control loops; *ii*) Data traffic congestion, data collision or interference can cause packet loss.

One of the most important issues in NCSs is the problem of losing data which occurs because of limited bandwidth and too many data packet transmitted over one line. For the real industrial applications, a good design of the feedback controller using the most fresh information to stabilize the NCS is very essential. Another common issue is the network-induced delay effect on the control loop. For real applications, delays and packet loss could become the potential sources of instability and poor performance of NCSs due to the critical real-time requirements in control systems.

Recently in the study of NCSs, the impact of packet loss and time delay have been paid increased attention such as shown in the literature: [13] - [27].

In [28], an overview regarding recent advances of time delay system is given, classical and predictor-based control laws are introduced. Several works about teleoperation control with constant time delay can are given by [29]- [31]. For more complicated case of time varying delays, [32] [33] proposed the predictor based approach for teleoperation by assuming that a dynamical ordinary differential equation model of the delay is available.

In [13], the issue of data packet loss is modelled as a Markovian process, but it dealt with the delay which is less than one sampling time interval. In [34], the maximum packet-loss rate under which the overall system remains stable was investigated. In [18], the NCS has been formulated as a Markovian jump system with known packet loss rate, the techniques developed for Markovian jump systems are applied in the work. [17] presented a solution to the problem of stabilization of NCSs with the effect of one sampling delay and arbitrary switching packets dropout in which the theory of switched systems has been applied to stabilize the NCSs. However, only one communication channel from the sensor to controller has been considered. The controller was supposed to be connected to the actuator directly and no packets were lost or delayed through the communication between them. In [21], the packet loss process has been defined as the sequence of the time intervals between consecutively successfully transmitted data and categorized into two types called the arbitrary packet loss process and the Markovian packet loss process, respectively. However, only one-step time delay has been considered in the design, analysis and synthesis methods are provided based on pure discrete-time model. The effect caused by network induced time varying delay has not been considered.

In next three subsections some existent works about stabilization control methodologies for NCSs with networked communication constraints are reviewed.

2.3.1 Lyapunov-Based Control Approaches

For control over network area, it is well-know that the choice of an appropriate Lyapunov-Krasovskii functional (LFK) is crucial for deriving stability and bounded real criteria. Choosing an LFK can obtain a solution to various control problems of NCSs. The general form of LKF has been used by many authors [35]- [38].

In [20], the sufficient conditions for stability of the linear system with uncertain delay via a new complete nominal LKF has been derived. This new nominal LKF along the trajectories of the nominal system which depends on the state. The state derivative allows a less conservative treatment of the delay perturbation. In their design, the uncertain delay includes two terms: the constant term and the varying term. It has been modeled by

$$\tau(t) = h + \eta(t), \tag{2.1}$$

where $h > 0$ is a nominal constant value and $\eta(t)$ is a time-varying perturbation with an artificial constraint that

$$|\eta(t)| \leq \mu \leq h, \tag{2.2}$$

where μ is the known upper bound of the varying term. By choosing a new complete nominal LKF, the sufficient conditions for stability of the nominal system are given in their work. All algorithms are given in terms of linear algebraic operation, definite integral and linear matrix inequalities (LMIs).

In [39], it presents a new delay system approach, which is based on a new time-delay model to NCSs. New results on stability are proposed for systems with two successive delay components by exploiting a new LKF and by using some new techniques for the time delay system. They consider the following system with two successive delays

$$\dot{\mathbf{x}} = A\mathbf{x} + A_d\mathbf{x}(t - d_1(t) - d_2(t)), \tag{2.3}$$

where $\mathbf{x}(t)$ is the state vector, A and A_d are known matrices with appropriate dimensions, $d_1(t)$ and $d_2(t)$ represent the two delay components in the state with the assumption that

$$\begin{aligned} 0 &\leq d_1 \leq \bar{d}_1 < \infty \\ 0 &\leq d_2 \leq \bar{d}_2 < \infty \\ \dot{d}_1 &\leq \tau_1 < \infty \\ \dot{d}_2 &\leq \tau_2 < \infty, \end{aligned} \tag{2.4}$$

and $\bar{d} = \bar{d}_1 + \bar{d}_2$, $\tau = \tau_1 + \tau_2$.

A Lyapunov-Krosovski functional has been defined as

$$V(t) \;=\; V_1(t) + V_2(t) + V_3(t) + V_4(t), \tag{2.5}$$

where

$$\begin{aligned} V_1(t) &= \mathbf{x}^T(t)P\mathbf{x}(t) \\ V_2(t) &= \int_{t-d_1(t)}^{t} \mathbf{x}^T(s)Q_1\mathbf{x}(s)ds + \int_{t-d(t)}^{t-d_1(t)} \mathbf{x}^T(s)Q_2\mathbf{x}(s)ds \\ V_3(t) &= \int_{t-\bar{d}}^{t} \mathbf{x}^T(s)R\mathbf{x}(s)ds \\ V_4(t) &= \int_{-\bar{d}_1}^{0}\int_{\beta}^{0} \dot{\mathbf{x}}^T(t+\alpha)Z_1\dot{\mathbf{x}}(t+\alpha)d\alpha d\beta \\ &\quad + \int_{-\bar{d}}^{-\bar{d}_1}\int_{\beta}^{0} \dot{\mathbf{x}}^T(t+\alpha)Z_2\dot{\mathbf{x}}(t+\alpha)d\alpha \\ &\quad + \int_{-\bar{d}}^{0}\int_{\beta}^{0} \dot{\mathbf{x}}^T(t+\alpha)M\dot{\mathbf{x}}(t+\alpha)d\alpha, \end{aligned} \tag{2.6}$$

The stability condition under which the system (2.3) is asymptotically stable for all delays $d_1(t)$ and $d_2(t)$ satisfying (2.4) are derived by finding the conditions for which the derivative of the LKF (2.5) is negative.

Similar works have been accomplished in [40], the authors divide the delay interval into two subintervals. A new LKF,

$$V(t) = \sum_{i=1}^{4} V_i(t), \tag{2.7}$$

where

$$\begin{aligned} V_1(t) &= \mathbf{x}^T(t)P\mathbf{x}(t) \\ V_2(t) &= \eta\int_{-\eta}^{0}\int_{t+\beta}^{t} \dot{\mathbf{x}}^T(\alpha)Z_1\dot{\mathbf{x}}(\alpha)d\alpha d\beta \\ &\quad +(h-\eta)\int_{-h}^{-\eta}\int_{t+\beta}^{t} \dot{\mathbf{x}}^T(\alpha)Z_1\dot{\mathbf{x}}(\alpha)d\alpha d\beta \\ V_3(t) &= \int_{t-h}^{t-\eta} \mathbf{x}^T(s)Q_1\mathbf{x}(s)ds + \int_{t-\eta}^{t} \mathbf{x}^T(s)Q_2\mathbf{x}(s)ds \\ V_4(t) &= 2\sum_{i=1}^{m}\lambda_i\int_{0}^{\sigma} f_i(\alpha)d\alpha, \end{aligned} \tag{2.8}$$

with $P > 0$, $Q_1 > 0$, $Q_2 > 0$, $Z_1 > 0$, $Z_2 > 0$. V_2 and V_3 in (2.8) splits the whole delay interval $[-h, 0]$ into two sub-internals which is $[-h, -\eta]$ and $[-\eta, 0]$ such that each sub-internals has a different Lyapunov matrix. Some new delay dependent stability criteria for the system are given by choosing the new LFK in (2.7).

The theoretic definition with constructing a LKF for linear time delay system has been investigated in [41]. In this work the following linear system with delayed states has been considered:

$$\dot{x}(t) = A_0x(t) + \sum_{i=1}^{k} A_ix(t-\tau_i), \tag{2.9}$$

the initial condition $x(\theta) = \phi(\theta)$, $\theta \in [-\tau, 0]$, where $\tau = max\{\tau_1, \ldots, \tau_k\}$. A_0 and A_i, $i = 1, \ldots, k$ are known real constant matrices with appropriate dimensions. The following LKF theorem is given in this work to prove asymptotic stability of the above system.

Theorem 2.1. *The system described by Eq.*(2.9) *is asymptotically stable if there exists a bounded quadratic Lyapunov functional* $V(x_t)$ *such that for some* $\epsilon > 0$*, it satisfies:*

$$V(x_t) \geq \epsilon \|x_t(0)\|^2 \tag{2.10}$$

and its derivative along the system trajectory satisfies

$$\dot{V}(x_t) \leq -\epsilon \|x_t(0)\|^2. \tag{2.11}$$

An example of a LKF that would yield a delay-independent condition is given first:

$$V(x_t) = x_t(0)^T P x_t(0) + \int_{-\tau}^{0} x_t(s)^T S x_t(s) ds. \tag{2.12}$$

To compare with the structure in (2.12), the complete LKF which is necessary and sufficient for delay-dependent stability of the linear system in (2.9) is given as follows:

$$\begin{aligned} V(x_t) \;=\; & x_t^T(0) P x_t(0) + x_t^T(0) \int_{-\tau}^{0} P_1 x_t(\theta) d\theta \\ & + \int_{-\tau}^{0} x_t^T(\theta) P_1^T(\theta)(\theta) d\theta x_t(0) \\ & + \int_{-\tau}^{0} \int_{-\tau}^{0} x_t^T(\theta) P_2(\theta, s) x_t(s) ds d\theta \\ & + \int_{-\tau}^{0} x_t^T(\theta) Q x_t(\theta) d\theta \end{aligned} \tag{2.13}$$

with appropriate continuity conditions on P, P_1 and P_2. The sufficient and necessary conditions for the asymptotic stability of the nonlinear system have been derived by considering the conditions in Theorem.2.1.

Another construction of LKF for time varying delay systems has been proposed in [42]. In this work the following linear time varying delay system has been considered:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + A_d\mathbf{x}(t-h(t)), \forall t \geq 0, \\ \mathbf{x}(t) &= \phi(t), \forall t \in [-h_m, 0], \end{aligned} \tag{2.14}$$

where $\mathbf{x}(t) \in \Re^n$ is the state vector, $A, A_d \in \Re^{n\times n}$ are known constant matrices and ϕ is the initial condition. The delay $h(t)$ is assumed to be a time-varying continuous function that satisfies

$$0 \leq h(t) \leq h_m, \tag{2.15}$$

where $h_m > 0$ may be infinite if delay independent conditions are looked for. In order to get the sufficient and necessary conditions for asymptotic stability of the system (2.14), the following LKF has been defined in this work:

$$V = \sum_{i=1}^{5} V_i(\mathbf{x}) \tag{2.16}$$

where

$$\begin{aligned} V_1 &= \mathbf{x}^T(t)P\mathbf{x}(t), \\ V_2 &= \int_{t-h(t)/2}^{t} \begin{bmatrix} \mathbf{x}(s) \\ \mathbf{x}(s-h(s)/2) \end{bmatrix}^T Q_1 \begin{bmatrix} \mathbf{x}(s) \\ \mathbf{x}(s-h(s)/2) \end{bmatrix} ds, \\ V_3 &= \int_{t-h(t)}^{t} \mathbf{x}^T(s)Q_0\mathbf{x}(s)ds, \\ V_4 &= \int_{t-h_m/2}^{t} \int_{s}^{t} \dot{\mathbf{x}}^t(\theta)R_1\dot{\mathbf{x}}(\theta)d\theta ds, \\ V_5 &= \int_{t-h_m}^{t} \int_{s}^{t} \dot{\mathbf{x}}^t(\theta)R_0\dot{\mathbf{x}}(\theta)d\theta ds, \end{aligned} \tag{2.17}$$

In this subsection, some examples of choosing an appropriate LKF have been reviewed. It is well known that the choice of an appropriate LKF is crucial for deriving stability and bounded real criteria and, as a result, for obtaining a solution to various NCS problems. In the next subsection, robust control approaches are reviewed and discussed.

2.3.2 Robust Control Approaches

When the concerned networked control system contains uncertainty parameters or disturbance or both, robust control method is needed to stabilize the system. The basic idea for robust control in NCSs is investigated. Consider the following uncertain system with time-varying delay:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= (A+\Delta A)\mathbf{x}(t) + (A_d + \Delta A_d)\mathbf{x}(t-\tau(t)) + B\mathbf{u}(t) + B_{\varpi}\varpi(t) \\ \mathbf{z}(t) &= C\mathbf{x}(t) + D_{\varpi}\varpi(t) + C_d\mathbf{x}(t-\tau(t)) + D\mathbf{u}(t) \\ \mathbf{x}(t) &= \phi(t), \end{aligned} \qquad (2.18)$$

where $\mathbf{x}(t)$ is the state vector; $\mathbf{u}(t)$ is the control input; $\varpi(t)$ is the disturbance input that belongs to $L_2[0,\infty)$; $\mathbf{z}(t)$ is the controlled output. $\phi(t)$ is the initial condition of the system. A and B are constant matrices with appropriate dimensions. ΔA and ΔB denote the parameter uncertainties satisfying that

$$[\Delta A, \Delta B] = MF(t)[E_a, E_b], \qquad (2.19)$$

where M, E_a and E_b are constant matrices with appropriate dimensions; $F(t)$ is an unknown time-varying matrix satisfying that $F^T(t)F(t) \leq I$. The varying time delay

$\tau(t)$ satisfies that

$$0 \leq \tau(t) \leq \tau_M, |\dot{\tau}(t)| \leq d \leq 1, \forall t \geq 0, \tag{2.20}$$

where τ_M and d are constants. The definition of robust stability is addressed by the following definition:

Definition 2.1. *The system* (2.18) *is robustly asymptotically stable with an H_∞ norm bound γ if the following conditions hold:*

1) *For the system with $\varpi(t) \equiv 0$, the trivial solution (equilibrium point) is globally asymptotically stable if* $\lim_{t\to\infty} \boldsymbol{x}(t) = 0$*;*

2) *Under the assumption of zero initial condition, the controller output $\boldsymbol{z}(t)$ satisfies*

$$\|\boldsymbol{z}(t)\|_2 \leq \gamma \|\varpi(t)\|_2 \tag{2.21}$$

for any nonzero $\varpi(t) \in L_2[0, \infty)$.

The key problem for robust control in NCSs is to find under what condition the system (2.18) is robustly asymptotically stable with an H_∞ norm bound γ. The potential approach regarding this problem can be classified into two categories: delay-dependent criteria and delay-independent criteria. Since delay-dependent criteria make use of information on the length of delays, it has less conservative than the delay-independent one. The main objective of the delay-dependent H_∞ control is to design a control approach that allows a maximum delay size for a fixed H_∞ performance bound or achieves a minimum H_∞ performance bound for a fixed delay size. The analysis is usually processed based on the common assumption (2.20) that the time delay has a bound. The main challenge of the delay-dependent H_∞ control is how to reduce the conservation of delay-dependent conditions for system stabilization as much as possible. LKF still plays an important role in this area.

In the past few years, some approaches have been proposed to reduce the conservation of delay-dependent conditions by using new bounding for cross terms or choosing new LKF. Park's inequality for bounding cross terms is proposed in [43] to investigate the delay-dependent stability criterion and reduce the conservation of delay-dependent conditions. However, some matrix variables must be limited to a certain structure form to obtain control synthesis conditions in terms of LMIs. And some conservatism has been brought in by this limitation. In [44], the authors propose a new inequality, which is more general than the one in [43], for bounding cross terms. By using this new bound in derivative of the proposed LKF, the authors present a delay-dependent robust H_∞ control which has less conservative than that in [43] for uncertain linear system with state delays. The drawback of this work is that the term $x(t-\tau)$ is replaced with $x(t)-\int_{t-\tau}^{t}\dot{x}(s)ds$ in the expression $2x^T(t)PA_1\dot{x}(t)$, but not with $\tau\dot{x}^T(t)Z\dot{x}(t)$. Since both $x(t-\tau)$ and $x(t)-\int_{t-\tau}^{t}\dot{x}(s)ds$ affect the result, there must be some relationship between the two terms and there must exist optimal weighting matrices for those terms. The authors select some fixed weighting matrices without giving a method for determining them.

Besides bounding for cross terms, the conservatism can be further reduced by taking a new LKF. In [45], the authors combined a descriptor model transformation with Park's inequality to produce less conservative criteria. However, since the basic idea is based on the substitution of $x(t)-\int_{t-\tau}^{t}\dot{x}(s)ds$ for $x(t-\tau)$, the conservatism of the approach in [43] is still there.

In [46], the authors proposed a new LKF to avoid some kind of model transformation and bounding of cross terms. A less conservative delay-dependent robust H_∞ control is proposed for uncertain linear systems with a state-delay and parameter uncertainties base on the new LKF. A new delay-dependent bounded real lemma for the system is derived in terms of LMIs. The condition for the proposed robust H_∞ control is given in terms of nonlinear matrix inequalities to obtain less conservatism. Similar work has been accomplished in [44].

In [47], the authors develop an approach in which neither model transformation nor free weighting matrix variables are employed in the proposed LKF. A tighter bounding technology for cross terms is employed in the proposed LKF to reduce the conservative. The proposed approach in their work can lead to a improvement in system analysis and synthesis for a large class of delay systems.

In [48], the authors present some delay-dependent stability criteria for linear systems with time-varying delays. In their work, first the asymptotical stability of a linear system with time-varying delays which has fixed system matrices (system does not have any disturbance) has been investigated. The upper bound of time-varying delays has been found by deriving a LKF. In the derivative of the LKF, the term $\dot{x}(t)$ is remained. However, the relationship among the terms in the systems equation is expressed by some free weighting matrices. Moreover, the relationship between $x(t)$, $x(t-d(t))$ and $\int_{t-d(t)}^{t} \dot{x}(s)ds$ is expressed in terms of free weighting matrices. The difficulties in the handling of the LKF is avoided by the introducing those free weighting matrices. LMI techniques are used to obtain all of the parameters numerically. Second, this idea is applied to a time-varying system with polytopic-type uncertainties.

A less conservative criterion is obtained.

In [49], the authors introduce the new free-weighting matrices to estimate the upper bound of the derivative of LKF without ignoring some useful terms. This new types of LKF brings less conservative delay-dependent stability criteria for systems with time-varying delays. The resulting criteria are extended to the stability analysis for systems with time-varying structured uncertainties.

In [50], the authors present a new criteria with some new advantages, such as there is not any system transformation so that the conservatism of such transformation can be avoided and the Park's inequality is not be used to estimate the upper bound of the cross term. It can reduce the conservatism in the derivation of the stability conditions. Some free weighting matrices, which are determined by solving LMIs, are employed to the system.

In [51], the authors propose a new criteria for the robust control approach of uncertain NCSs. They introduce slack matrix parameters to the derived criteria for reducing the conservative results. The lower bound of the network-induced delay is employed to derive the criteria to obtain less conservative results especially for the case where the lower bound of the delay is nonzero.

2.4 Networked Multi-agent Systems

Recently the study on the area of coordinated and cooperative control of multi-agent systems has been paid more attention due to its wide potential application background, such as platooning of vehicles in urban transportation [52] [53], the operation of the multiple robots [54], autonomous underwater vehicles [55] [56] and formation of aircrafts in military affairs [57] [58].

Figure 2.3. Agents in schools

Investigation for multi-agent systems begins with studying the behavior of a large number of interacting agents with a common group objective. These agents include fish, ants and birds (Fig.2.3). In a flock of fish, the desired track for each agent in the group has been already decided according to some external elements. Each agent should track the existent desired trajectory and acquire some information, such as the velocities and relative positions from its neighbors in order to: *i*) stay a proper distance to nearby flock-mates; *ii*) avoid collisions; *iii*) match velocity with each other. Then all agents asymptotically move with the same velocity and form a cohesive flock without collisions. This phenomena is called consensus. Notice that in the whole group movement there is no one performs as the“leader”. Animal behavior scientists believe that“schools need no leaders” [59].

Consensus problems have a long history in automata theory. In multi-agent systems, consensus means that the states of all agents reach an agreement asymptotically regarding a certain quantity of interest that depends on the state of all agents. Generally speaking, when multiple agent agree to the value of a variable of interest, they are said to reach consensus. For example, in many applications involving multi-agent or multi-vehicle systems, groups of agents need to agree upon certain quantities of interest. Such quantities might or might not be related to the motion of the individual agents. So in order to achieve consensus, there must be a shared variable of interest, called the information state, as well as appropriate algorithmic methods, called consensus algorithms, for negotiating to reach consensus on the value of that variable. The most common continuous time consensus algorithm is given by:

$$\dot{x}(t)_i = -\sum_{j=1}^{n} a_{ij}(t)[x_i(t) - x_j(t)], i = 1, \ldots, n, \tag{2.22}$$

where $a_{ij}(t)$ is the (i, j) entry of adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$, x_i is the information state of the ith agent. $a_{ij} = 0$ means agent i cannot receive information from agent j. The key problem to investigate when the information states of all of the agents converge to a common value (consensus)?

A good literature review for the consensus problem can be found in [60]. The most often used method to tackle the consensus problem with multi-agent systems is called graph theory (graph Laplacians) [61] [62]. It plays a crucial role in the convergence analysis of consensus. In [11] [63], the authors used graph theory to tackle the formation stabilization for groups of linear agents. For this method, the consensus algorithm in (2.22) is rewritten in matrix form as

$$\dot{\mathbf{x}}(t) = -\mathcal{L}_n \mathbf{x}(t), \tag{2.23}$$

where $\mathbf{x} = [x_1, \ldots, x_n]$ is the information state and $\mathcal{L}_n$ is the Laplacian matrix. Consensus is achieved by a team of agents if $|x_i(t) - x_j(t) \to 0|$ as $t \to \infty$. The key idea for graph theory is to investigate the conditions of the Laplacian matrix $\mathcal{L}_n$ that can make the whole team achieve consensus.

Another approach is to use Lyapunov method to derive sufficient conditions for stabilization controllers. This Lyapunov function also can be renamed as the disagreement function which is a measure of group disagreement in a network. In [64] a common Lyapunov function that guarantees asymptotic convergence to a group decision value in networks is proposed. Similar analysis idea was used in [65]. First the authors introduced two control algorithms which included the relative position term, the relative velocity term and the navigational feedback term. Then the disagreement functions were used to analyze the stability of the whole system. Compared with the graph method, Lyapunov stability analysis is more control theoretic, it can visually give the sufficient conditions for stabilizations and be closer to real applications. In this thesis work LKF method is used to tackle the problem in this thesis.

For coordinated and cooperative control of multi-agent systems, information exchange becomes a central issue. For some applications, e.g. multi-vehicle system is required to observe on the same target. All vehicles should agree as what changes took place in the environment. To agree to the same interest, each vehicle need to share information with others so that the environment change can be captured and known by the whole team.

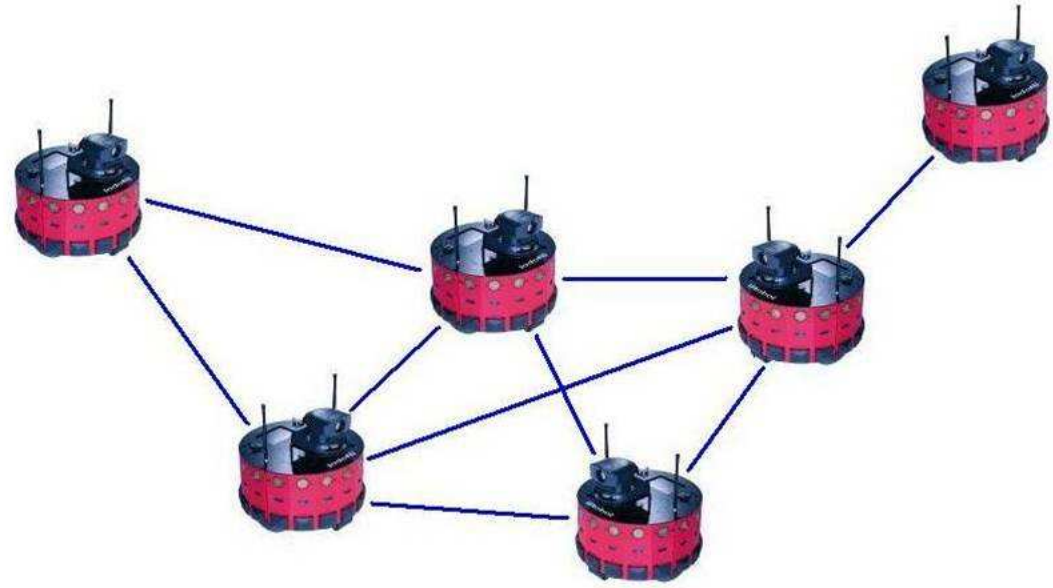


Figure 2.4. Robot group formation

Formation control is an important application of consensus-based design methodologies for cooperative control (Fig.2.4). There are several advantages for formation control such as increased feasibility, robustness, efficiency and probability of success. The approaches regarding formation control can be categorized into leader-follower and virtual leader (structure) approaches. In the leader-follower approach, one of the agents is designated as the leader, and others as followers. The basic idea is that the followers track the position and orientation of the leader with some prescribed offset. The key problem is to find an appropriate consensus-based algorithm that can make the information states of all agents converge to the common prescribed value of interest.

The first study on leader-follower strategies is in [66]. The formation control strategies are discussed in this work. In [67], the formation control strategies for keeping desired formation and relative attitude alignment based on nearest neighbor tracking error are proposed. In [68], a leader-following technique is use to control a group of mobile robots to move an objective cooperatively. In [69], nonholonomic robots are used for leader-follower. In addition, the formation configuration is described as a directed graph. The shape of the formation is changed when the graph is changed. Another work regarding multiple nonholonomic robot formation control is represented in [70].

In the virtual leader and virtual structure approach, the entire formation is treated as a single structure. The basic idea is, first define the desired dynamics of the virtual leader; second, translate the virtual leader and virtual structure into the desired motion for each vehicle; last design a consensus-based tracking control strategy for each agent. In [71], the authors use the virtual structure approach to acquire high precision formation control for mobile robots. In [72], the virtual structure is applied to formation control of spacecraft in free space. In [73], the virtual leader approach is applied to formation control of mobile robots. In [74], a Lyapunov formation function is used to define a formation error and formation feedback is incorporated in the virtual leaders through parameterized trajectories.

2.5 Summary

In this chapter, the works related to control over network have been reviewed in section 2.3 which include two areas: 1) Lyapunov-based control approach; 2) robust control approach. In section 2.4, existence works regarding consensus problem and consensus based formation control for multi-agent system have been reviewed and summarized.

Chapter 3

Network Communication Channel: Issues and Modeling

This chapter begins with introducing the process of network communication. Now take a simple network program as an example. As shown in Fig.3.1, the simple program is used by some clients to login to a remote system just like login in locally. The commands are sent by a local computer and all other activity is happening on the remote computer. There is a similar issue between human beings and computers that they all need to "speak" the same language in order to communicate. For computers, the communication is carried out in a pre-defined manner, called a "protocol". The computer protocols determine how each side behaves and how it should react to behavior by its counterpart. Generally speaking even the interaction between the computer and the hardware, such as the hard disk, can be considered as a "protocol".

The most common protocols used today is Transmission Control Protocol/ Internet Protocol (TCP/IP) and the User Datagram Protocol (UDP). TCP/IP is the most important internetworking protocol suite in the world. Because it actually consists of several different protocols, it is more accurate to call TCP/IP a "protocol suite".

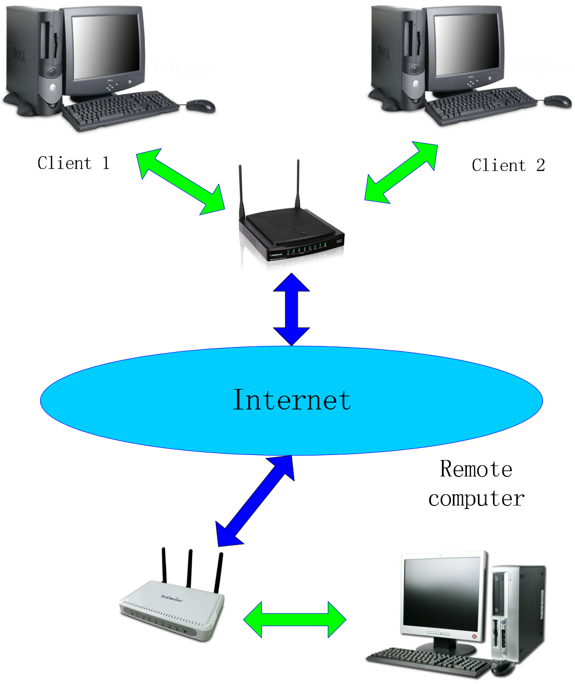


Figure 3.1. Network configuration

TCP/IP is connection-oriented 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. However, re-sending the out-of-date packets is usually not a big help for NCSs. UDP is another important internet protocol suite designed by David P. Reed in 1980. UDP uses a simple transmission model without dialogues for guaranteeing reliability, ordering or data integrity. Thus, the datagrams may arrive out of order, appear duplicated or get lost with UDP protocol. It is better to be used in some real-time applications such as voice and video transmissions since a lost packet will not be retransmitted with UDP protocol. For these reasons, UDP is chosen here as the investigative objective.

The main issues regarding the network communication channel with UDP protocol is time delays and packet losses. So these two issues are considered to the modelling of the UDP link. The time delays are not explicitly modelled, but the upper bounds of delays are considered in some models of network links. The packet losses are modelled by using Markov Chains which will be discussed more in Chapter 8. These models are used in the control gain designs, but not for the simulations or experiments. In the simulations and experiments discussed in Chapter 7 and 8, the real networked communication environment is applied. In the next subsection, some general issues regarding network communication channels are discussed.

3.1 Bounded Time Delays

In some real applications of NCSs, out-of-date packets are usually discarded. So there is a common assumption regarding NCSs that the delay is bounded. This assumption can be imposed in network communication links by dropping these packets which are received at the times that exceed the delay bounds. The bound of delay can be determined by measuring networked induced delays.

Fig.3.2 shows the time delays of up to 6000 continuous packets transferred in a real network communication link. From this picture, most of the packets can be received in 50 ms. So the upper bound of delays in the simulation and experiment is set to $\tau_{bound} = 50ms$. In another model, the packet, whose delay τ_i is larger than the summation of sampling time T_s and the next packet delay τ_{i+1}, is dropped. This condition can be expressed by the following equation:

$$\tau_i > \tau_{i+1} + T_s. \tag{3.1}$$

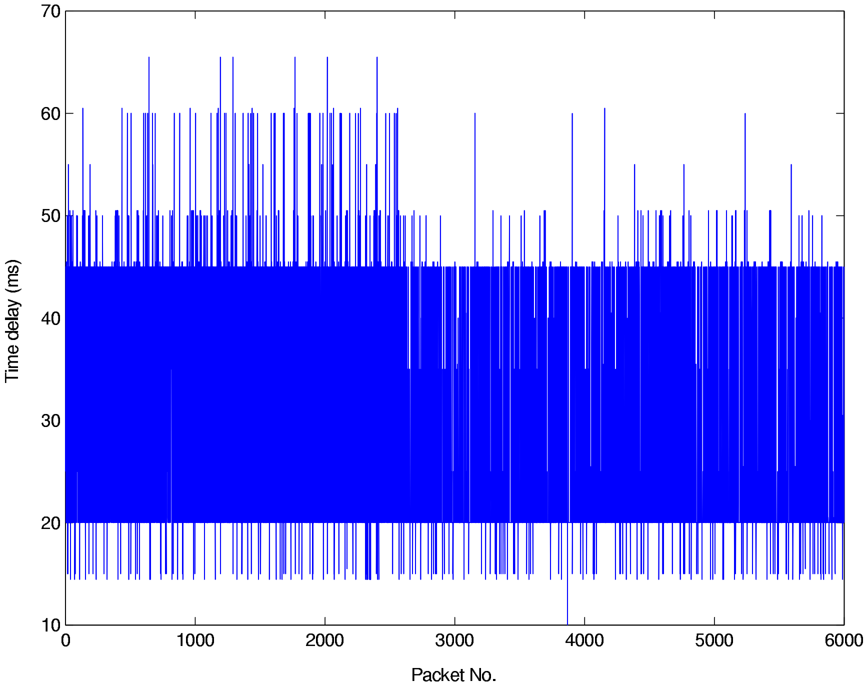


Figure 3.2. Time delay vs numbers of data packet

Dropping out-of-order packets results in changing the delay distribution and increasing packet loss rate. More details are discussed in Chapter 8.

3.2 Deterministic and Stochastic Packet Dropout

Packet dropouts can be modeled either as deterministic or stochastic phenomena. It is also often assumed that when the packet containing x_k is dropped, the NCS use the previous value of $\hat{x}_k$ as represented in the following equation:

$$\hat{x}_k = \theta_k x_k + (1 - \theta_k)\hat{x}_{k-1}, \tag{3.2}$$

where $\theta_k = 1$ where there is no packet dropout, $\theta_k = 0$ when there is packet dropout.

In [34], the authors present a deterministic dropout model with packet dropouts occurring at an asymptotic rate define by the following time average:

$$r = \lim_{T \to \infty} \frac{1}{T} \sum_{k=k_0}^{k_0+T-1} (1 - \theta_k), \forall k_0 \in N \tag{3.3}$$

In [75] and [18], the authors consider stochastic data dropout. In their formulation, θ_k is a Bernoulli process that the probability of dropout ($\theta_k = 0$) is equal to $\rho \in [0, 1)$. Under this stochastic data dropout model, the system (3.3) is a special case of a discrete-time Markovian jump linear system. The index θ_k in (3.3) is the state of a discrete-time Markov chain with a finite number of states and a given transition probability matrix. In Bernoulli process, the Markov chain only has two states and the transition probability between the two states is shown in Fig.3.3.

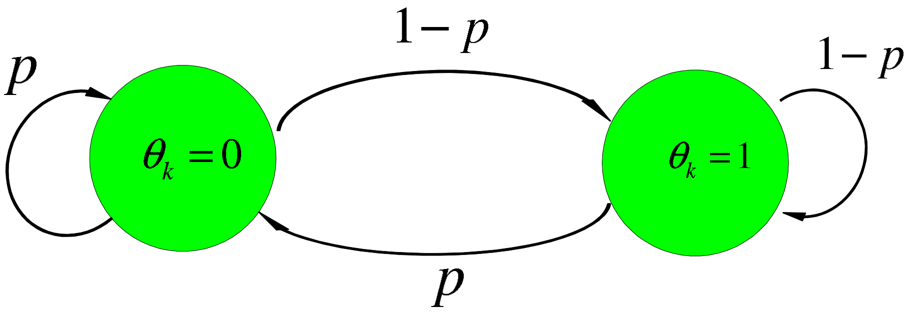


Figure 3.3. Transition probability for stochastic data dropout

3.3 Summary

In this chapter, the network communication protocols were introduced and discussed. The main issues for network communication links were generally introduced. The assumption regarding the bounds of the delay was discussed. Deterministic and stochastic model of the packet dropout were studied and discussed.

Chapter 4

Stochastic Stabilization of Sampled-data Networked Control Systems

This chapter mainly investigated a sampled-data control approach to deal with the stabilization problem of Networked Control Systems (NCSs) with packet losses and bounded time varying delays. A new Lyapunov-Krasovskii functional candidate is constructed to analyze the stability of the overall system with bounded random packet losses and time varying delays. As a result, the stabilizing sampled-data controller is designed based on the stability conditions. A real-time network measurement system has been developed based on MATLAB applications. Instrument Control toolbox was used to implement communications between two computers with MATLAB applications via the internet. Experiments were done to demonstrate the real network properties. A real-time networked control system has been constructed to test the stabilizing ability of the controller design in a real network environment. Experimental results illustrate the effectiveness of the proposed approach, a good combination of the theory and the real applications.

4.1 Introduction

In this work, first a real-time network induced delay and packet loss measurement system was built to study the real network property. With the experimental measurements, the characters of time varying delays and packet losses in the real network were figured out, which was applied to the stochastic stabilization analysis. Since in many practical systems, such as computer-based control systems, the continuous-time system is controlled by a sampled-data controller with sample and hold devices, the control objective is set to design a sampled-data stabilizing controller via communication channels with Markovian packet losses and bounded time varying delays. To solve the problem on the stochastic stabilization of the NCSs with the effect of Markovian packet loss only, stability conditions are derived via Lyapunov approach and the corresponding stabilizing sampled-data controller design techniques are also given based on the conditions. Then the effects of both Markovian packet loss and time varying delays occurring in both channels are considered. The delays and packet losses that were measured during the network experiments are firstly replayed for the simulation. A real-time networked control system has been built based on MATLAB programming environment to test the stabilizing ability of the controller. The experimental results illustrate that the controller works well under real situation.

Notations: $\lambda(D)$ denotes the eigenvalue of the matrix D, where $D \in R^{n\times n}$. $E(\cdot)$ is the expectation. $\rho(A) = \sqrt{\lambda_{max}(A^T A)}$.

4.2 Review of the Sampled-data System

In many practical control system, such as computer-based control systems, the continuous-time system is controlled by a digital controller. The advantages offered by digital controllers include improved sensitivity, better reliability, less effect due to noise and disturbance and less cost and weight. Fig.4.1 shows the block diagram of the digital control system. The plant is part of the whole system. It accepts continuous-time signals as inputs and gives out continuous-time signals as outputs. The sample-and-hold device (S/H) can convert continuous signals into a train of amplitude-modulated pulses and maintain the value of the pulse for a prescribed time duration. And then the converted signals can be read by digital controllers.

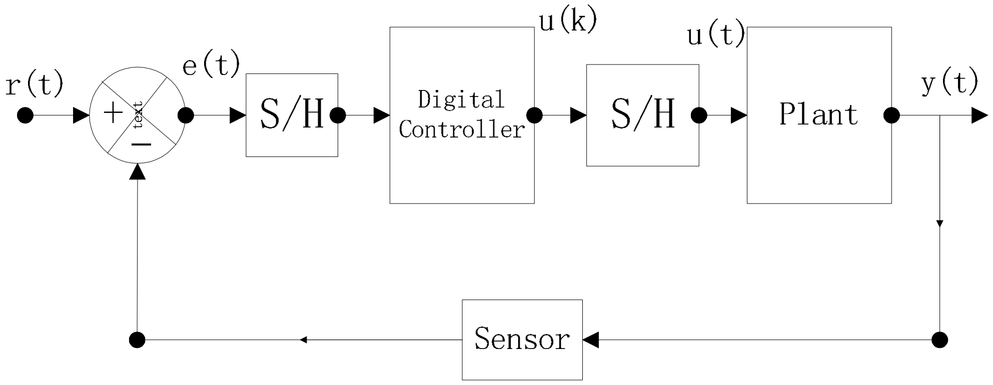


Figure 4.1. Block diagram of digital control system

Sampled-data control formulation is a direct design method for digital controllers. Modelling of continuous-time systems with digital control in the form of continuous-time systems with delayed control input is introduced in [76]. The delayed digital control law can be represented as follows:

$$u(t) = u_d(t_k) = u_d(t - \tau(t)), \tag{4.1}$$

where $t_k \leq t < t_{k+1}$ and $\tau(t) = t - t_k$ is the varying time delay, $\tau \leq t_{k+1} - t_k$, u_d is a discrete-time control signal. Based on such a model, for small enough sampling intervals $t_{k+1} - t_k$, asymptotic approximations of the trajectory can be constructed.

In [77], a new approach is suggested to solve the problem for a continuous-time system with uncertain but bounded time-varying delay in the control input. The system can be expressed as follows:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{4.2}$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the control input. A sampled-data control law of the form:

$$\mathbf{u}(t) = \mathbf{u}_d(t_k), t_k \leq t \leq t_{k+1}, \tag{4.3}$$

where $\mathbf{u}_d$ is a sampled-data control signal and t_k is the sampling instant. In this work, a state-feedback controller

$$\mathbf{u}(t) = K\mathbf{x}(t_k), \tag{4.4}$$

has been designed to stabilize the system (4.2).

4.3 Mathematic Model of Communication Channels with Random Packet Loss

NCSs with pure Markovian packet-loss are considered first in this section. Specially, a linear continuous-time system is studied,

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{4.5}$$

where $\mathbf{x}(t) \in \Re^n$ and $\mathbf{u}(t) \in \Re^m$ represent the system state and control input, respectively. $\mathbf{x}_0 = \mathbf{x}(0)$ is the initial state. A and B are two known constant matrices of appropriate dimensions.

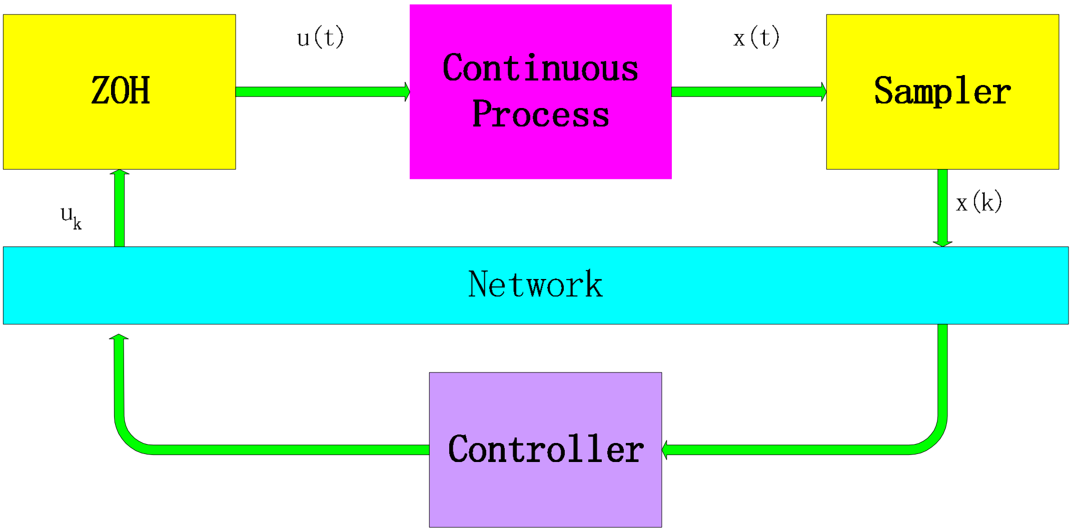


Figure 4.2. Block diagram of the sampled-data NCSs

As shown in Fig.4.2, the plant communicates with the sampled data controller via the networks on both channels, which are called controller-actuator and sampler-controller channels. The controller in the the system is assumed to be time driven.

Let $\ell = \{n_1, n_2, \ldots\}$ be a subsequence of $1, 2, 3, \ldots$, which denote the sequence of time points of successful data transmissions from the sampler to the controller. $S = \max_{n_J \in \ell}(n_{J+1} - n_J)$ is the maximum packet-loss upper bound. In order to capture the nature of packet loss in the network, the following concept and mathematical models are first introduced.

Definition 4.1. *Packet-loss process in the communication channel is modelled as*

$$\eta(n_J) = n_{J+1} - n_J : n_J \in \ell, \tag{4.6}$$

which takes values in the finite state space $\zeta = \{1, 2, \ldots S\}$ *where* S *is a positive* *integer.*

Definition 4.2 (11). *Packet-loss process* (4.6) *is said to be Markovian if it is a discrete-time homogeneous Markov chain on a complete probability space, and takes values in* ζ *with known transition probability matrix* $\Pi = (\pi_{ij}) \in \Re^{S\times S}$, *where*

$$\pi_{ij} = Pr(\eta(n_{J+1}) = j|\eta(n_J) = i) \geq 0 \tag{4.7}$$

for all $i, j \in \zeta$, *and* $\sum_{j=1}^{S} \pi_{ij} = 1$ *for each* $i \in \zeta$.

An illustrative example of data flow with packet loss is shown in Fig.4.3.

In order to catch the random packet loss properties, experiments have been done in real network environment. All the experimental data has been recorded and analyzed to derive the transition probability matrix Π in the real case.

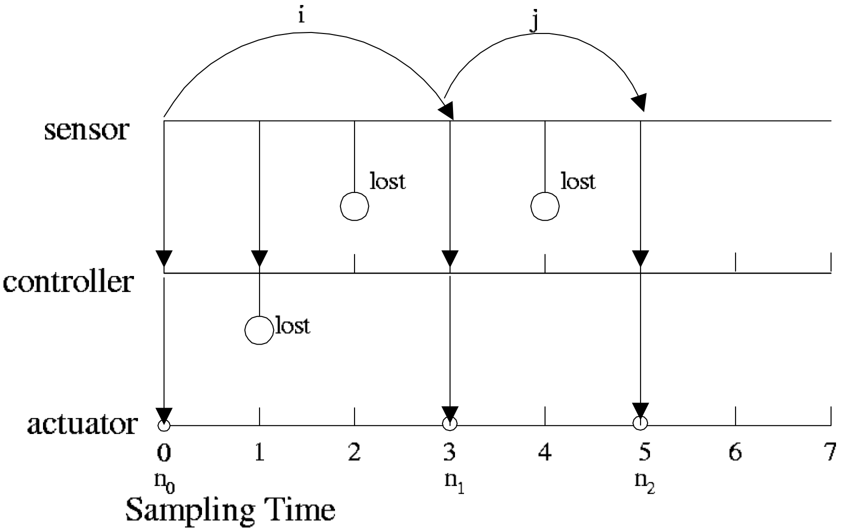


Figure 4.3. Data flow diagram of the Markovian Packet loss

Definition 4.3. *For system* (4.5) *with Markovian packet-loss process* (4.6), *the equilibrium point 0 of* $\boldsymbol{x}$ *is stochastically stable if, for every initial state* x_0, *the following* *holds:*

$$E\{\sum_{J=0}^{\infty} \boldsymbol{x}^T(n_J)\boldsymbol{x}(n_J)|\eta(n_0)\} < \infty. \tag{4.8}$$

Throughout this chapter, the sampled-data controller is designed as a state-feedback controller

$$\mathbf{u}(t) = \mathbf{u}(n_J T_s) = K\mathbf{x}(n_J T_s), \tag{4.9}$$

where $K \in \Re^{m\times n}$ is designed as constant matrix with suitable dimension. The initial control input is set to zero: $\mathbf{u}(0) = 0$. Then the closed-loop system becomes

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + BK\mathbf{x}(n_J T_s), \quad n_J \in \ell. \tag{4.10}$$

Definition 4.4. *System* (4.5) *with Markovian packet-loss process* (4.6) *is stochastically stable if, for every initial condition $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$, there exists a sampled-data linear feedback control law $\boldsymbol{u}(t) = \boldsymbol{u}(n_J T_s) = K\boldsymbol{x}(n_J T_s)$ such that the closed-loop system*

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + BK\boldsymbol{x}(n_J T_s), \quad n_J \in \ell, \tag{4.11}$$

is stochastically stable.

Lemma 4.1. - *Jensen Inequality [26] For any constant matrix $E \in \Re^{n\times n}$, $E = E^T > 0$, vector function $\boldsymbol{\omega} : [0,\tau] \to \Re^n$ such that the integrations concerned are well defined, then,*

$$\tau \int_0^{\tau} \boldsymbol{\omega}^T(s)E\boldsymbol{\omega}(s)ds \geq \left[\int_0^{\tau} \boldsymbol{\omega}(s)ds\right]^T E \left[\int_0^{\tau} \boldsymbol{\omega}(s)ds\right]. \tag{4.12}$$

The control objective is to design the controller (4.9) so that the system (4.10) with Markovian packet-loss process (4.6) is stochastically stable.

4.4 Controller Design and Stability Analysis

4.4.1 NCSs with Markovian Packet Losses

Now the stability property of NCSs will be analyzed. For NCSs with Markovian packet-loss process, the stability condition is established by analyzing the theory from Markovian jump linear systems. The conditions are given in the following theorem.

Theorem 4.1. *Consider the NCS* (4.10) *with Markovian packet-loss process* (4.6), $\boldsymbol{u}(0) = 0$ *and* $\boldsymbol{x}(n_0T_s) = \boldsymbol{x}(0)$*. If there exist symmetric positive definition matrices* $P_i = P = P^T$*,* $i \in \zeta$*, matrices* $Q > 0$*,* $S > 0$*,* X *and* Y*, scalar* $\beta > 0$ *such that*

$$\begin{bmatrix} X^TA^T + AX & BY \\ Y^TB^T & -R \end{bmatrix} < 0, \tag{4.13}$$

and

$$\begin{bmatrix} -X & * & * & * \\ \Phi & X - G & * & * \\ \Phi & 0 & -X & * \\ X & 0 & 0 & -\beta^{-1}I \end{bmatrix} < 0, \tag{4.14}$$

with $\Phi = \sum_{i=1}^{S} \pi_{ij} \cdot j \cdot BY + X,\ \ X = P^{-1},\ \ R = X^TSX,\ \ G = X^TA^{-T}QA^{-1}X$*, hold, then the system is stochastically stable with the controller gain designed as* $K = YX^{-1}$*.*

Proof. Given that $\mathbf{x}(n_0T_s) = \mathbf{x}(0)$ and $n_1 - n_0 = i$, from the system (4.10)

$$\begin{aligned} \mathbf{x}(n_1T_s) - \mathbf{x}(n_0T_s) &= \int_{n_0T_s}^{n_1T_s} \dot{\mathbf{x}}(s)ds \\ &= \int_{n_0T_s}^{n_1T_s} A\mathbf{x}(s)ds + \int_{n_0T_s}^{n_1T_s} BK \cdot \mathbf{x}(n_0T_s)ds \end{aligned}$$

$$
\begin{aligned}
&= A\int_{n_0T_s}^{n_1T_s}\mathbf{x}(s)ds + BK\cdot\mathbf{x}_0\int_{n_0T_s}^{n_1T_s}ds \\
&= A\int_{n_0T_s}^{n_1T_s}\mathbf{x}(s)ds + BK\cdot\mathbf{x}_0\cdot iT_s, \qquad (4.15)
\end{aligned}
$$

then the difference between the system state and its expectation can be expressed as,

$$
\begin{aligned}
&E[\mathbf{x}(n_2T_s)|n_1-n_0=i]-\mathbf{x}(n_1T_s) \\
&= A\int_{n_1T_s}^{n_2T_s}\mathbf{x}(s)ds + \Sigma_{j=1}^{S}\pi_{ij}\cdot jT_sBK\mathbf{x}(n_1T_s). \qquad (4.16)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
&E[\mathbf{x}(n_JT_s)|n_{J-1}-n_{J-2}=i]-\mathbf{x}(n_{J-1}T_s) \\
&= A\int_{n_{J-1}T_s}^{n_JT_s}\mathbf{x}(s)ds + \Sigma_{j=1}^{S}\pi_{ij}\cdot jT_sBK\mathbf{x}(n_{J-1}T_s). \qquad (4.17)
\end{aligned}
$$

Now take the packet-loss dependent Lyapunov functional candidate as

$$
V_1(t) = \mathbf{x}^T(t)P_i\mathbf{x}(t) + (n_{J+1}T_s - t)\mathbf{x}^T(n_JT_s)S\mathbf{x}(n_JT_s), \qquad (4.18)
$$

where $P_i = P \in \Re^{n\times n}$ are singular positive definition matrices, and $S \in \Re^{n\times n} > 0$. Then the derivative of $V_1(t)$ becomes

$$
\begin{aligned}
\dot{V}_1(t) &= \dot{\mathbf{x}}^T(t)P_i\mathbf{x}(t) + \mathbf{x}^T(t)P_i\dot{\mathbf{x}}(t) - \mathbf{x}^T(n_JT_s)S\mathbf{x}(n_JT_s) \\
&= [A\mathbf{x}(t)+BK\mathbf{x}(n_JT_s)]^TP_i\mathbf{x}(t) + \mathbf{x}^T(t)P_i[A\mathbf{x}(t)+BK\mathbf{x}(n_JT_s)] \\
&\quad -\mathbf{x}^T(n_JT_s)S\mathbf{x}(n_JT_s) \\
&= \mathbf{x}^T(t)A^TP_i\mathbf{x}(t) + \mathbf{x}^T(n_JT_s)K^TB^TP_i\mathbf{x}(t) + \mathbf{x}^T(t)P_iA\mathbf{x}(t) \\
&\quad +\mathbf{x}^T(t)P_iBK\mathbf{x}(n_JT_s) - \mathbf{x}^T(n_JT_s)S\mathbf{x}(n_JT_s) \\
&= \mathbf{x}^T(t)[A^TP_i + P_iA]\mathbf{x}(t) + \mathbf{x}^T(n_JT_s)K^TB^TP_i\mathbf{x}(t) \\
&\quad +\mathbf{x}^T(t)P_iBK\mathbf{x}(n_JT_s) - \mathbf{x}^T(n_JT_s)S\mathbf{x}(n_JT_s)
\end{aligned}
$$

$$= \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(n_J T_s) \end{bmatrix}^T \begin{bmatrix} A^T P_i + P_i A & P_i BK \\ K^T B^T P_i & -S \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(n_J T_s) \end{bmatrix} < 0. \qquad (4.19)$$

The inequality above holds if the following inequality is satisfied:

$$\begin{bmatrix} A^T P_i + P_i A & P_i BK \\ K^T B^T P_i & -S \end{bmatrix} < 0, \qquad (4.20)$$

for symmetric positive definite matrices $P_i = P, P = P^T$. Define $X = P^{-1}$, $M = diag(X, X)$ and $Y = KX$. Then by pre-multiplying the inequality in (4.20) by M^T and post-multiplying by M, we can obtain the following inequality,

$$\begin{bmatrix} XA^T + AX & BY \\ Y^T B^T & -R \end{bmatrix} < 0, \qquad (4.21)$$

where $R = X^T S X$. If (4.21) holds, then the closed loop system (4.10) is asymptotically stable.

In order to achieve the condition for stochastic stability of the closed loop system (4.10), here take another packet-loss dependent Lyapunov function as

$$V_2(t) \;=\; \mathbf{x}^T(t) P_i \mathbf{x}(t) + (n_{J+1}T_s - t) \int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}^T(s) Q \mathbf{x}(s) ds, \qquad (4.22)$$

where P_i are the same matrices as in $V_1(t)$, and $Q \in \Re^{n \times n}$ is symmetric positive definiton matrix. Then from (4.17) and Lemma 4.1

$$\begin{aligned} & E[V_2(n_{J+1}T_s) | n_J - n_{J-1} = i] - V_2(n_J) \\ = \; & E[\mathbf{x}(n_{J+1}T_s) | n_J - n_{J-1} = i]^T P_j E[\mathbf{x}(n_{J+1}T_s) | n_J - n_{J-1} = i] \\ & -\mathbf{x}^T(n_J T_s) P_i \mathbf{x}(n_J T_s) - (n_{J+1}T_s - n_J T_s) \int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}^T(s) Q \mathbf{x}(s) ds \end{aligned}$$

$$
\begin{aligned}
= \quad & [A\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds + (\sum_{j=1}^{S}\pi_{ij}\cdot jT_sBK+I)\mathbf{x}(n_JT_s)]^TP_j[A\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds \\
& +(\sum_{j=1}^{S}\pi_{ij}\cdot jT_sBK+I)\mathbf{x}(n_JT_s)] - \mathbf{x}^T(n_JT_s)P_i\mathbf{x}(n_JT_s) \\
& -(n_{J+1}T_s - n_JT_s)\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}^T(s)Q\mathbf{x}(s)ds \\
= \quad & [\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds]^TA^TP_jA[\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds] \\
& +[\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds]^TA^TP_j[(\sum_{j=1}^{S}\pi_{ij}\cdot jBK+I)]\mathbf{x}(n_JT_s) \\
& +\mathbf{x}^T(n_JT_s)[\sum_{j=1}^{S}\pi_{ij}\cdot jBK+I]^TP_jA[\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds] \\
& +\mathbf{x}^T(n_JT_s)[\sum_{j=1}^{S}\pi_{ij}\cdot jBK+I]^TP_j[\sum_{j=1}^{S}\pi_{ij}\cdot jBK+I]\mathbf{x}(n_JT_s) \\
& -\mathbf{x}^T(n_JT_s)P_i\mathbf{x}(n_JT_s) - (n_{J+1}T_s - n_JT_s)\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}^T(s)Q\mathbf{x}(s)ds \qquad (4.23)
\end{aligned}
$$

then

$$
\begin{aligned}
& E[V_2(n_{J+1}T_s)|n_J - n_{J-1} = i] - V_2(n_J) \\
= \quad & [\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds]^TA^TP_jA[\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds] + [\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds]^TA^TP_j \\
& [(\sum_{j=1}^{S}\pi_{ij}\cdot jT_sBK+I)]\mathbf{x}(n_JT_s) + \mathbf{x}^T(n_JT_s)[\sum_{j=1}^{S}\pi_{ij}\cdot jT_sBK+I]^TP_jA \\
& [\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}(s)ds] + \mathbf{x}^T(n_JT_s)\{[\sum_{j=1}^{S}\pi_{ij}\cdot jT_sBK+I]^TP_j \\
& [\sum_{j=1}^{S}\pi_{ij}\cdot jT_sBK+I] - P_i\}\mathbf{x}(n_JT_s) + \beta\mathbf{x}^T(n_JT_s)\mathbf{x}(n_JT_s) \\
& -(n_{J+1}T_s - n_JT_s)\int_{n_JT_s}^{n_{J+1}T_s}\mathbf{x}^T(s)Q\mathbf{x}(s)ds - \beta\mathbf{x}^T(n_JT_s)\mathbf{x}(n_JT_s)
\end{aligned}
$$

$$
\begin{aligned}
\leq\ & [\int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}(s)ds]^T A^T P_j A[\int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}(s)ds] + [\int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}(s)ds]^T A^T P_j \\
& [(\Sigma_{j=1}^{S}\pi_{ij} \cdot jT_s BK + I)]\mathbf{x}(n_J T_s) + \mathbf{x}^T(n_J T_s)[\Sigma_{j=1}^{S}\pi_{ij} \cdot jT_s BK + I]^T P_j A \\
& [\int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}(s)ds] + \mathbf{x}^T(n_J T_s)\{[\Sigma_{j=1}^{S}\pi_{ij} \cdot jT_s BK + I]^T P_j \\
& [\Sigma_{j=1}^{S}\pi_{ij} \cdot jT_s BK + I] - P_i\}\mathbf{x}(n_J T_s) + \beta \mathbf{x}^T(n_J T_s)\mathbf{x}(n_J T_s) \\
& -[\int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}(s)ds]^T Q[\int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}(s)ds] - \beta \mathbf{x}^T(n_J T_s)\mathbf{x}(n_J T_s). \qquad (4.24)
\end{aligned}
$$

If the following inequality holds,

$$
\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}^T \begin{bmatrix} \Delta + \beta I & \Xi^T P_j A \\ A^T P_j \Xi & A^T P_j A - Q \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} < 0, \qquad (4.25)
$$

where

$$
\begin{aligned}
\Xi &= \sum_{j=1}^{S} \pi_{ij} \cdot j \cdot BK + I \\
\Delta &= [\sum_{j=1}^{S} \pi_{ij} \cdot j \cdot BK + I]^T P_j [\sum_{j=1}^{S} \pi_{ij} \cdot j \cdot BK + I] - P_i \\
Z_1 &= \mathbf{x}(n_J T_s), \quad Z_2 = [\int_{n_J T_s}^{n_{J+1}T_s} \mathbf{x}(s)ds], \qquad (4.26)
\end{aligned}
$$

then it means

$$
E[V_2(n_{J+1}T_s)|n_J - n_{J-1} = i] - V_2(n_J) < -\beta \mathbf{x}^T(n_J T_s)\mathbf{x}(n_J T_s). \qquad (4.27)
$$

The following inequality is satisfied by Schur complement of (4.25) as,

$$
\begin{bmatrix} \beta I - P_i & \Xi^T P_j A & \Xi^T P_j \\ A^T P_j \Xi & A^T P_j A - Q & 0 \\ P_j \Xi & 0 & -P_j \end{bmatrix} < 0, \qquad (4.28)
$$

if there exists symmetric positive definition matrices $P_i = P, P = P^T$. Define $X = P^{-1}$, $W = diag(X, A^{-1}X, X)$ and $Y = KX$. Then by pre-multiplying the inequality in (4.28) by W^T and post-multiplying by W, the following inequality can be obtained,

$$\begin{bmatrix} X^T\beta IX - X & * & * \\ \sum_{j=1}^{S} \pi_{ij} \cdot j \cdot BY + X & X - G & * \\ \sum_{j=1}^{S} \pi_{ij} \cdot j \cdot BY + X & 0 & -X \end{bmatrix} < 0. \tag{4.29}$$

The inequality (4.29) could be represented as the form in (4.14) by Schur complement. Notice that the inequality in (4.14) is a sufficient condition for the solvability of (4.25) based on the derivation.

If (4.27) holds, then

$$\frac{E[V_2(n_{J+1}T_s)|n_J - n_{J-1} = i] - V_2(n_JT_s)}{V_2(n_JT_s)} \leq \frac{-\beta \mathbf{x}^T(n_JT_s)\mathbf{x}(n_JT_s)}{V_2(n_JT_s)}. \tag{4.30}$$

Define $0 < \alpha = 1 - \beta \min\{\frac{1}{\lambda \max(P_i)}\} < 1$, it is obvious that

$$\frac{-\beta \mathbf{x}^T(n_JT_s)\mathbf{x}(n_JT_s)}{\mathbf{x}^T(n_JT_s)P_i\mathbf{x}(n_JT_s)} \leq -\beta \min\{\frac{1}{\lambda \max(P_i)}\} = \alpha - 1. \tag{4.31}$$

From (4.30), there exist a positive γ with $\alpha \leq \gamma < 1$, such that

$$\frac{E[V_2(n_{J+1}T_s)|n_J - n_{J-1} = i] - V_2(n_JT_s)}{V(n_JT_s)} \leq \gamma - 1. \tag{4.32}$$

Then

$$\begin{cases} E[V(n_{J+1}T_s)|n_J - n_{J-1} = i] \leq \gamma V(n_JT_s), \\ \qquad \vdots \\ E[V(n_2T_s)|\eta(n_1) = i] \leq \gamma V(n_1T_s), \\ E[V(n_1T_s)|\eta(n_0) = i] \leq \gamma V(n_0T_s). \end{cases} \tag{4.33}$$

Taking the expectation $E[\cdot|\eta(n_0) = i]$ on both sides of (4.33)

$$\begin{aligned} E[V_2(n_2T_s)|\eta(n_0) = i] &\leq \gamma E[V_2(n_1T_s)|\eta(n_0) = i] \\ &\leq \gamma^2 V_2(n_0T_s). \end{aligned} \tag{4.34}$$

By iterative derivation,

$$E[V_2(n_{J+1}T_s)|\eta(n_0) = i] \leq \gamma E[V_2(n_JT_s)|\eta(n_0) = i] \leq \ldots \leq \gamma^{J+1} V_2(n_0T_s). \tag{4.35}$$

From (4.35), the summation of $E[V_2(\cdot)|\eta(n_0) = i]$ when $J = 0 \sim N$ becomes

$$E[\sum_{J=0}^{N} V_2(n_{J+1}T_s)|\eta(n_0) = i] \leq \frac{1-\gamma^N}{1-\gamma} V_2(n_0T_s).$$

As a result,

$$\begin{aligned} \lim_{N\to\infty} E[\sum_{J=0}^{N} V_2(n_{J+1}T_s)|\eta(n_0) = i] &= \lim_{N\to\infty} E[\sum_{J=0}^{N} \mathbf{x}^T(n_{J+1}T_s)P_j\mathbf{x}(n_{J+1}T_s)|\eta(n_0) = i] \\ &\leq \frac{1}{1-\gamma} V_2(n_0T_s). \end{aligned} \tag{4.36}$$

From (4.36),

$$\lim_{N\to\infty} E[\sum_{J=0}^{N} \mathbf{x}^T(n_{J+1}T_s)\mathbf{x}(n_{J+1}T_s)|\eta(n_0) = i] \leq \frac{1}{\max \rho(P_j)(1-\gamma)} V_2(n_0T_s) \tag{4.37}$$

The limit of the expectation in (4.37) is bounded, this completed the proof for the stochastic stability of the closed loop system.

□

Remark 4.1. *In Theorem.*(4.1)*, Eq.*(4.13) *and Eq.*(4.14) *are used to get the feasible solution of the control gain K. Then the system is stochastically stable with the existence of random packet loss in both of the transmission channels.*

In this section a sampled-data controller has been designed to stabilize the NCSs with pure Markovian packet loss process, packet-loss dependent Lyapunov functional candidates are used to establish the stabilization conditions. In the next section, network induced time varying delays are taken into account too.

4.4.2 NCSs with Bounded Delays and Stochastic Packet Losses

Network-induced time delay is also an important issue. Normally network-induced delay is varying during the data transmission process. In this work, the time delay is considered as varying with a known upper bound. The system considered here is assumed to be a simple linear continuous-time system with time varying delay of the form,

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t - d_2(t)), \tag{4.38}$$

where $\mathbf{x}(t) \in \Re^n$ and $\mathbf{u}(t) \in \Re^m$ are the system state and control input, respectively. $d_2(t)$ is the time varying delay with a certain bound. $\mathbf{x}_0 = \mathbf{x}(0)$ is the initial state. A and B are two constant matrices of appropriate dimensions. $d_2(t)$ is the controller to actuator delay. $S = \max_{n_J \in \ell}(n_{J+1} - n_J)$ is the maximum packet-loss upper bound. One example of data flow of the NCSs with random time varying delays and packet losses is shown as follows. In Fig.4.4, the data flow of NCSs with time varying delay and packet loss is shown with $d_1(t - d_2(t))$ being the sampler to controller delay.

The actuator and controller considered in this work are both time driven and $d_1(t - d_2(t)) + d_2(t) = m_J T_s$ where $m_J \in \{1, 2, ...M_b\}$ with a known positive integer M_b. The sampled-data controller is designed as a static state-feedback controller and

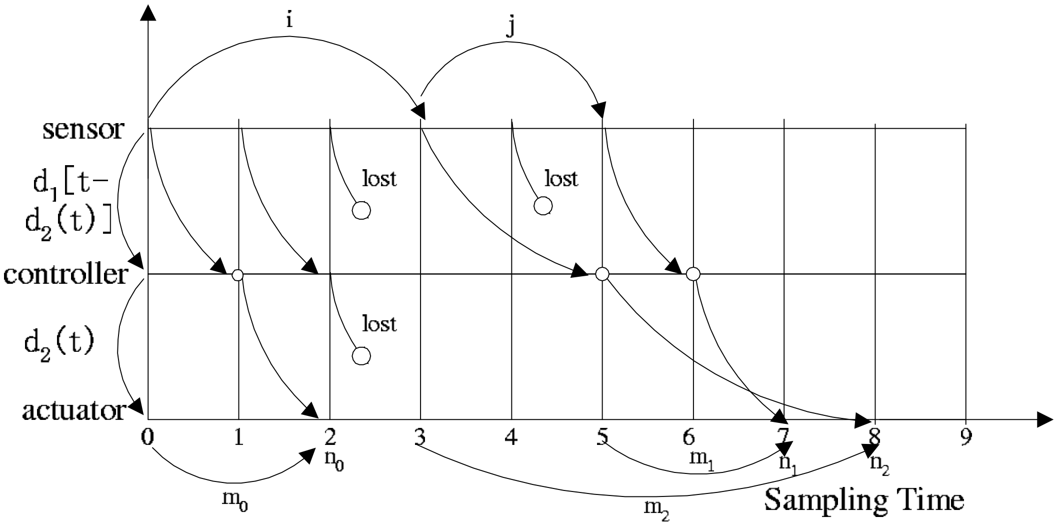


Figure 4.4. Data flow diagram of NCSs with time varying delay and packet loss

it can be represented as

$$\begin{aligned}\mathbf{u}(t-d_2(t)) &= K\mathbf{x}[t-d_2(t)-d_1(t-d_2(t))] \\ &= K\mathbf{x}(n_J T_s - m_J T_s), \qquad (4.39)\end{aligned}$$

where $K \in \Re^{m\times n}$ is to be designed. The initial control input is set to be zero: $\mathbf{u}(0) = 0$. T_s is the sampling time of the sampled-data controller. Then the closed-loop system becomes

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + BK\mathbf{x}(n_J T_s - m_J T_s), \quad n_J \in \ell. \qquad (4.40)$$

The objective is to design the controller (4.39) so that the closed loop system (4.40) with Markovian packet-loss process (4.38) is stochastically stable.

Theorem 4.2. *Consider the system* (4.40) *with Markovian packet-loss process* (4.6), $\boldsymbol{u}(0) = 0$ *and* $\boldsymbol{x}(n_0) = \boldsymbol{x}(0)$. *If there exists symmetric positive definition matrice* $S > 0, P = \begin{bmatrix} P_1 & P_2 \\ P_2^T & P_3 \end{bmatrix} > 0,\ Q = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix} > 0,\ R = \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix} > 0,$ *matrices* $P_i = \hat{P} = \hat{P^T} \in \Re^+, i \in \zeta, \hat{Q} = \hat{Q}^T > 0, \hat{S} > 0,$ *X and Y with appropriate* *dimensions and scalars* $\beta > 0$, $M_b > 0$ *such that the following two inequalities holds,*

$$\begin{bmatrix} H_{11} & * & * & * & * & * & * & * \\ H_{21} & H_{22} & * & * & * & * & * & * \\ H_{31} & H_{32} & H_{33} & * & * & * & * & * \\ H_{41} & H_{42} & H_{43} & H_{44} & * & * & * & * \\ H_{51} & H_{52} & H_{53} & H_{54} & H_{55} & * & * & * \\ H_{61} & H_{62} & H_{63} & 0 & 0 & H_{66} & * & * \\ H_{71} & H_{72} & H_{73} & H_{74} & H_{75} & H_{76} & H_{77} & * \\ H_{81} & H_{82} & H_{83} & H_{84} & H_{85} & H_{86} & H_{87} & H_{88} \end{bmatrix} < 0, \tag{4.41}$$

where

$$\begin{aligned}
H_{11} &= Q_{11} - M_1^T A - A^T M_1 + M_b T_s R_{11}, \quad H_{21} = P_2^T + N_1 - M_2^T A \\
H_{22} &= N_2 + N_2^T + S, \quad H_{31} = P_1 + M_1 - M_3^T A, \quad H_{32} = N_3^T + M_2 \\
H_{33} &= Q_{22} + M_b T_s R_{22} + M_3^T + M_3, \quad H_{41} = -M_4^T A - P_2^T \\
H_{42} &= N_4^T, \quad H_{43} = M_4, \quad H_{44} = -Q_{11}, \quad H_{51} = -M_5^T A \\
H_{52} &= P_3 + N_5^T, \quad H_{53} = P_2^T + M_5^T, \quad H_{54} = -P_3, \quad H_{55} = \frac{-1}{M_b T_s} R_{11} \\
H_{61} &= -M_6^T A, \quad H_{62} = N_6^T, \quad H_{63} = M_6^T, \quad H_{66} = \frac{-1}{M_b T_s} R_{22} \\
H_{71} &= -N_1 - M_7^T A, \quad H_{72} = -N_2 + N_7^T, \quad H_{73} = -N_3 + M_7^T \\
H_{74} &= -N_4, \quad H_{75} = -N_5, \quad H_{76} = -N_6, \quad H_{77} = -N_7^T - N_7 \\
H_{81} &= -N_1 - M_8^T A - K^T B^T M_1, \quad H_{82} = N_8^T - N_2 - K^T B^T M_2 \\
H_{83} &= -N_3 + M_8^T - K^T B^T M_3, \quad H_{84} = -N_4 - K^T B^T M_4 \\
H_{85} &= -N_5 - K^T B^T M_5, \quad H_{86} = -N_6 - K^T B^T M_6, \\
H_{87} &= -N_8^T - N_7 - K^T B^T M_7, \\
H_{88} &= -S - N_8^T - N_8 - M_8 B K - K^T B^T M_8
\end{aligned}$$

and

$$\begin{bmatrix} -X & * & * & * \\ \sum_{i=1}^{S} \pi_{ij} \cdot j \cdot BY + X & X - \hat{E} & * & * \\ \sum_{i=1}^{S} \pi_{ij} \cdot j \cdot BY + X & 0 & -X & * \\ X & 0 & 0 & -\beta^{-1} I \end{bmatrix} < 0, \tag{4.42}$$

with $\hat{E} = X^T A^{-T} \hat{Q} A^{-1} X$, then the system is stochastically stable with the controller gain $K = YX^{-1}$.

Proof. As shown in Fig.4.4, at the beginning of data transmissions, the equation

$$(n_1 - m_1) - (n_0 - m_0) = i, \tag{4.43}$$

holds. From the system (4.40), we have

$$\begin{aligned} & \mathbf{x}[(n_1 - m_1)T_s] - \mathbf{x}[(n_0 - m_0)T_s] = \int_{(n_0-m_0)T_s}^{(n_1-m_1)T_s} \dot{\mathbf{x}}(s) ds \\ = & \int_{(n_0-m_0)T_s}^{(n_1-m_1)T_s} A\mathbf{x}(s) ds + \int_{(n_0-m_0)T_s}^{(n_1-m_1)T_s} BK \cdot \mathbf{x}[(n_0 - m_0)T_s] ds \\ = & A \int_{(n_0-m_0)T_s}^{(n_1-m_1)T_s} \mathbf{x}(s) ds + BK \cdot \mathbf{x}[(n_0 - m_0)T_s] \int_{(n_0-m_0)T_s}^{(n_1-m_1)T_s} ds \\ = & A \int_{(n_0-m_0)T_s}^{(n_1-m_1)T_s} \mathbf{x}(s) ds + BK \cdot \mathbf{x}[(n_0 - m_0)T_s] \cdot iT_s, \end{aligned} \tag{4.44}$$

then we can get

$$\begin{aligned} & E[\mathbf{x}[(n_2 - m_2)T_s] | n_1 - n_0 = i] - \mathbf{x}[(n_1 - m_1)T_s] \\ = & A \int_{(n_1-m_1)T_s}^{(n_2-m_2)T_s} \mathbf{x}(s) ds + \sum_{j=1}^{S} \pi_{ij} \cdot jT_s BK\mathbf{x}[(n_1 - m_1)T_s]. \end{aligned} \tag{4.45}$$

By the iterative derivation we get the following formulation:

$$\begin{aligned}&E[\mathbf{x}((n_{J+1}-m_{J+1})T_s)|n_J-n_{J-1}=i]-\mathbf{x}[(n_J-m_J)T_s]\\=A&\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s}\mathbf{x}(s)ds+\sum_{j=1}^{S}\pi_{ij}\cdot jT_sBK\mathbf{x}((n_J-m_J)T_s).\end{aligned}\tag{4.46}$$

Now take the Lyapunov Krasovskii functional candidate

$$\begin{aligned}V_1 &= \mathbf{x}^T(t)P_1\mathbf{x}(t)+\sum_{l=0}^{M_b-1}2\mathbf{x}^T(t)P_2[\int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s}\mathbf{x}(s)ds]\\&+\sum_{l=0}^{M_b-1}[\int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s}\mathbf{x}(s)ds]^TP_3[\int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s}\mathbf{x}(s)ds]\\&+\sum_{l=0}^{M_b-1}\int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s}[\mathbf{x}^T(s),\dot{\mathbf{x}}^T(s)]Q\begin{bmatrix}\mathbf{x}(s)\\\dot{\mathbf{x}}(s)\end{bmatrix}ds\\&+\int_{n_JT_s}^{t}[\mathbf{x}^T(s),\dot{\mathbf{x}}^T(s)]Q\begin{bmatrix}\mathbf{x}(s)\\\dot{\mathbf{x}}(s)\end{bmatrix}ds\\&+\sum_{l=0}^{L}\int_{(n_J-m_J+l)T_s}^{(n_J-m_J+l+1)T_s}\mathbf{x}(s)S\mathbf{x}(s)ds\\&+\sum_{l=0}^{M_b-1}\int_{-(M_b-l)T_s}^{-(M_b-l-1)T_s}\int_{n_JT_s+\theta}^{t}[\mathbf{x}^T(s),\dot{\mathbf{x}}^T(s)]R\begin{bmatrix}\mathbf{x}(s)\\\dot{\mathbf{x}}(s)\end{bmatrix}dsd\theta.\end{aligned}\tag{4.47}$$

With appropriate dimensions, the following two zero equations hold:

$$\begin{aligned}\phi_1 &= 2\mathbf{z}^TN\left\{\mathbf{x}(n_JT_s)-\sum_{l=0}^{L}\int_{(n_J-m_J+l)T_s}^{(n_J-m_J+l+1)T_s}\dot{\mathbf{x}}(s)ds-\mathbf{x}[(n_J-m_J)T_s]\right\}\\&= 0\\\phi_2 &= 2\mathbf{z}^TM\left\{\dot{\mathbf{x}}(t)-A\mathbf{x}(t)-BK\mathbf{x}[(n_J-m_J)T_s]\right\}=0\\L &= m_J-1,\end{aligned}\tag{4.48}$$

where

$$
\begin{aligned}
&\mathbf{z} = [\mathbf{x}^T(t), \mathbf{x}^T(n_J T_s), \dot{\mathbf{x}}^T(t), \mathbf{x}^T[(n_J - M_b)T_s], \sum_{l=0}^{M_b-1} \int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s} \mathbf{x}^T(s)ds,\\
&\sum_{l=0}^{M_b-1} \int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s} \dot{\mathbf{x}}^T(s)ds, \sum_{l=0}^{L} \int_{(n_J-m_J+l)T_s}^{(n_J-m_J+l+1)T_s} \dot{\mathbf{x}}^T(s)ds, \mathbf{x}^T[(n_J - m_J)T_s]]^T\\
&N = [N_1^T, N_2^T, N_3^T, N_4^T, N_5^T, N_6^T, N_7^T, N_8^T]^T\\
&M = [M_1^T, M_2^T, M_3^T, M_4^T, M_5^T, M_6^T, M_7^T, M_8^T]^T. \qquad (4.49)
\end{aligned}
$$

Then the derivative of the Lyapunov function candidate is as follows:

$$
\begin{aligned}
\dot{V}_1 &= \dot{V}_1 + \phi_1 + \phi_2\\
&= \dot{\mathbf{x}}^T(t)P_1\mathbf{x}(t) + \mathbf{x}^T(t)P_1\dot{\mathbf{x}}(t)\\
&\quad +2\dot{\mathbf{x}}^T(t)P_2[\sum_{l=0}^{M_b-1} \int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s} \mathbf{x}(s)ds]\\
&\quad +2\mathbf{x}^T(t)P_2[\mathbf{x}(n_J T_s) - \mathbf{x}[(n_J - M_b)T_s]] + [\mathbf{x}(n_J T_s)\\
&\quad -\mathbf{x}[(n_J - M_b)T_s]]^T P_3[\sum_{l=0}^{M_b-1} \int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s} \mathbf{x}(s)ds]\\
&\quad +[\sum_{l=0}^{M_b-1} \int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s} \mathbf{x}(s)ds]^T P_3[\mathbf{x}(n_J T_s) - \mathbf{x}[(n_J - M_b)T_s]]\\
&\quad +[\mathbf{x}^T(t), \dot{\mathbf{x}}^T(t)]Q \begin{bmatrix} \mathbf{x}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} + \mathbf{x}^T(n_J T_s)S\mathbf{x}(n_J T_s)\\
&\quad -[\mathbf{x}^T[(n_J - M_b)T_s], \dot{\mathbf{x}}^T[(n_J - M_b)T_s]]Q \begin{bmatrix} \mathbf{x}[(n_J - M_b)T_s] \\ \dot{\mathbf{x}}[(n_J - M_b)T_s] \end{bmatrix}\\
&\quad -\mathbf{x}^T[(n_J - m_J)T_s]S\mathbf{x}[(n_J - m_J)T_s]\\
&\quad +M_b T_s[\mathbf{x}^T(t), \dot{\mathbf{x}}^T(t)]R \begin{bmatrix} \mathbf{x}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix}\\
&\quad -\sum_{l=0}^{M_b-1} \int_{(n_J-M_b+l)T_s}^{(n_J-M_b+l+1)T_s} [\mathbf{x}^T(s), \dot{\mathbf{x}}^T(s)]R \begin{bmatrix} \mathbf{x}(s) \\ \dot{\mathbf{x}}(s) \end{bmatrix} ds
\end{aligned}
$$

$$
\begin{aligned}
&+2\mathbf{z}^T N[\mathbf{x}(n_J T_s) - \sum_{l=0}^{L} \int_{(n_J - m_J + l)T_s}^{(n_J - m_J + l + 1)T_s} \dot{\mathbf{x}}(s)ds - \mathbf{x}[(n_J - m_J)T_s]] \\
&+2\mathbf{z}^T M[\dot{\mathbf{x}(t)} - A\mathbf{x}(t) - BK\mathbf{x}[(n_J - m_J)T_s]]
\end{aligned}
\qquad (4.50)
$$

Using Lemma 4.1 and (4.50), we have

$$
\begin{aligned}
\dot{V}_1 \leq\ & \dot{\mathbf{x}^T}(t)P_1\mathbf{x}(t) + \mathbf{x}^T(t)P_1\dot{\mathbf{x}}(t) \\
&+2\dot{\mathbf{x}}^T(t)P_2[\sum_{l=0}^{M_b-1} \int_{(n_J - M_b + l)T_s}^{(n_J - M_b + l + 1)T_s} \mathbf{x}(s)ds] \\
&+2\mathbf{x}^T(t)P_2[\mathbf{x}(n_J T_s) - \mathbf{x}[(n_J - M_b)T_s]] \\
&+[\mathbf{x}(n_J T_s) - \mathbf{x}[(n_J - M_b)T_s]]^T P_3[\sum_{l=0}^{M_b-1} \int_{(n_J - M_b + l)T_s}^{(n_J - M_b + l + 1)T_s} \mathbf{x}(s)ds] \\
&+[\sum_{l=0}^{M_b-1} \int_{(n_J - M_b + l)T_s}^{(n_J - M_b + l + 1)T_s} \mathbf{x}(s)ds]^T P_3[\mathbf{x}(n_J T_s) - \mathbf{x}[(n_J - M_b)T_s]] \\
&+[\mathbf{x}^T(t), \dot{\mathbf{x}}^T(t)]Q \begin{bmatrix} \mathbf{x}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} \\
&-[\mathbf{x}^T[(n_J - M_b)T_s], \dot{\mathbf{x}}^T[(n_J - M_b)T_s]]Q \begin{bmatrix} \mathbf{x}[(n_J - M_b)T_s] \\ \dot{\mathbf{x}}[(n_J - M_b)T_s] \end{bmatrix} \\
&+\mathbf{x}^T(n_J T_s)S\mathbf{x}(n_J T_s) - \mathbf{x}^T[(n_J - m_J)T_s]S\mathbf{x}[(n_J - m_J)T_s] \\
&+M_b T_s[\mathbf{x}^T(t), \dot{\mathbf{x}}^T(t)]R \begin{bmatrix} \mathbf{x}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} \\
&-\frac{1}{M_b T_s} \sum_{l=0}^{M_b-1} [\int_{(n_J - M_b + l)T_s}^{(n_J - M_b + l + 1)T_s} \mathbf{x}(s)ds]^T R_{11}[\int_{(n_J - M_b + l)T_s}^{(n_J - M_b + l + 1)T_s} \mathbf{x}(s)ds] \\
&-\frac{1}{M_b T_s} \sum_{l=0}^{M_b-1} [\int_{(n_J - M_b + l)T_s}^{(n_J - M_b + l + 1)T_s} \dot{\mathbf{x}}(s)ds]^T R_{22}[\int_{(n_J - M_b + l)T_s}^{(n_J - M_b + l + 1)T_s} \dot{\mathbf{x}}(s)ds] \\
&+2\mathbf{z}^T N[\mathbf{x}(n_J T_s) - \sum_{l=0}^{L} \int_{(n_J - m_J + l)T_s}^{(n_J - m_J + l + 1)T_s} \dot{\mathbf{x}}(s)ds - \mathbf{x}[(n_J - m_J)T_s]] \\
&+2\mathbf{z}^T M[\dot{\mathbf{x}}(t) - A\mathbf{x}(t) - BK\mathbf{x}[(n_J - m_J)T_s]] \\
=\ & z^T H z,
\end{aligned}
\qquad (4.51)
$$

where H was shown in (4.41).

If (4.41) holds, then from (4.51), we have $\dot{V}_1 \leq 0$ which means that the closed loop system (4.40) is asymptotically stable. In order to achieve the condition for stochastic stability of the closed system, here consider the Lyapunov functional candidate

$$V_2 = \mathbf{x}^T(t)P_i\mathbf{x}(t) + [(n_{J+1} - m_{J+1})T_s - t]\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}^T(s)\hat{Q}\mathbf{x}(s)ds. \quad (4.52)$$

Then take the expectation of the Lyapunov functional candidate,

$$\begin{aligned}
& E[V_2[(n_{J+1} - m_{J+1})T_s]|n_J - n_{J-1} = i] - V_2[(n_J - m_J)T_s] \\
= \; & E[\mathbf{x}[(n_{J+1} - m_{J+1})T_s]|n_J - n_{J-1} = i]^T P_j E \\
& [\mathbf{x}[(n_{J+1} - m_{J+1})T_s]|n_J - n_{J-1} = i] - \mathbf{x}^T[(n_J - m_J)T_s]P_i\mathbf{x}[(n_J - m_J)T_s] \\
& -[(n_{J+1} - m_{J+1} - n_J + m_J)T_s]\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}^T(s)\hat{Q}\mathbf{x}(s)ds \\
= \; & [A\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds + (\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1)\mathbf{x}[(n_J - m_J)T_s]]^T P_j \\
& [A\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds + (\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1)\mathbf{x}[(n_J - m_J)T_s]] \\
& -\mathbf{x}^T[(n_J - m_J)T_s]P_i\mathbf{x}[(n_J - m_J)T_s] - [(n_{J+1} - m_{J+1} - n_J + m_J)T_s] \\
& \int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}^T(s)\hat{Q}\mathbf{x}(s)ds \\
= \; & [\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds]^T A^T P_j A[\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds] \\
& +2[\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds]^T A^T P_j(\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1)\mathbf{x}[(n_J - m_J)T_s] \\
& +\mathbf{x}[(n_J - m_J)T_s]^T(\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1)^T P_j(\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1) \\
& \mathbf{x}[(n_J - m_J)T_s] - \mathbf{x}^T[(n_J - m_J)T_s]P_i\mathbf{x}[(n_J - m_J)T_s] \\
& -[(n_{J+1} - m_{J+1} - n_J + m_J)T_s]\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}^T(s)\hat{Q}\mathbf{x}(s)ds
\end{aligned}$$

$$\begin{aligned}
&\leq \; [\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds]^T A^T P_j A[\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds] \\
&\quad +2[\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds]^T A^T P_j(\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1)\mathbf{x}[(n_J - m_J)T_s] \\
&\quad +\mathbf{x}[(n_J - m_J)T_s]^T [(\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1)^T P_j(\sum_{j=1}^{S} \pi_{ij} \cdot jT_s \cdot BK + 1) - P_i] \\
&\quad \mathbf{x}[(n_J - m_J)T_s] - [\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds]^T \hat{Q}[\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds] \\
&\leq \; -\beta\mathbf{x}^T[(n_J - m_J)T_s]\mathbf{x}[(n_J - m_J)T_s] \leq 0. \qquad (4.53)
\end{aligned}$$

(4.53) holds if the following inequality is satisfied:

$$\hat{\mathbf{z}}^T \begin{bmatrix} \phi^T P_j \phi - P_i + \beta \cdot I & * \\ A^T P_j \phi & A^T P_j A - \hat{Q} \end{bmatrix} \hat{\mathbf{z}} < 0. \qquad (4.54)$$

where $\phi = (\sum_{j=1}^{S} \pi_{ij} \cdot jTs \cdot BK + 1)$,$\hat{\mathbf{z}} = \begin{bmatrix} \mathbf{x}(n_J T_s - m_J T_s) \\ \int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}(s)ds \end{bmatrix}$, with $P_i = \hat{P}$, using Schur complement, (4.54) is equivalent to

$$\begin{bmatrix} \beta \cdot I - \hat{P} & * & * \\ A^T \hat{P} \phi & A^T \hat{P} A - \hat{Q} & * \\ \hat{P}\phi & 0 & -\hat{P} \end{bmatrix} < 0. \qquad (4.55)$$

Define $X = \hat{P}^{-1}, W = diag(X, A^{-1}X, X), Y = KX$. Then pre-multiplying (4.55) by W^T and post-multiplying by W, using schur complement, we can obtain the inequality (4.42). With suitable values of X and Y, the LMI (4.42) holds, then we have

$$\begin{aligned}
\frac{E[V_2(n_{J+1}T_s - m_{J+1}T_s)|n_J - n_{J-1} = i] - V_2(n_J T_s - m_J T_s)}{V_2(n_J T_s - m_J T_s)} &\leq \\
\frac{-\beta\mathbf{x}^T(n_J T_s - m_J T_s)\mathbf{x}(n_J T_s - m_J T_s)}{\mathbf{x}^T(t)P_i\mathbf{x}(t) + (n_{J+1}T_s - m_{J+1}T_s - t)\int_{(n_J-m_J)T_s}^{(n_{J+1}-m_{J+1})T_s} \mathbf{x}^T(s)\hat{Q}\mathbf{x}(s)ds}&. \qquad (4.56)
\end{aligned}$$

Define $0 < \alpha = 1 - \beta \min\{\frac{1}{\lambda \max(\hat{P}_i)}\} < 1$, it is obvious that

$$\frac{-\beta \mathbf{x}^T(n_J T_s - m_J T_s)\mathbf{x}(n_J T_s - m_J T_s)}{\mathbf{x}^T(n_J T_s - m_J T_s)P_i \mathbf{x}(n_J T_s - m_J T_s)} \leq -\beta \min\{\frac{1}{\lambda \max(\hat{P}_i)}\} = \alpha - 1. \tag{4.57}$$

From (4.56) and (4.57), there exists a γ with $\alpha \leq \gamma < 1$ such that

$$\frac{E[V_2(n_{J+1}T_s - m_{J+1}T_s)|n_J - n_{J-1} = i] - V_2(n_J T_s - m_J T_s)}{V_2(n_J T_s - m_J T_s)} \quad \leq \quad \gamma - 1, \tag{4.58}$$

Then we obtain

$$\begin{cases} E[V_2(n_{J+1}T_s - m_{J+1}T_s)|n_J - n_{J-1} = i] \leq \gamma V_2(n_J T_s - m_J T_s), \\ \qquad \vdots \\ E[V_2(n_2 T_s - m_2 T_s)|\eta(n_1) = i] \leq \gamma V_2(n_1 T_s - m_1 T_s), \\ E[V_2(n_1 T_s - m_1 T_s)|\eta(n_0) = i] \leq \gamma V_2(n_0 T_s - m_0 T_s). \end{cases} \tag{4.59}$$

Taking $E[\cdot|\eta(n_0) = i]$ on both sides of (4.59), we have

$$\begin{aligned} E[V_2(n_2 T_s - m_2 T_s)|\eta(n_0) = i] \quad &\leq \quad \gamma E[V_2(n_1 T_s - m_1 T_s)|\eta(n_0) = i] \\ &\leq \quad \gamma^2 V_2(n_0 T_s - m_0 T_s). \end{aligned} \tag{4.60}$$

By iterative derivation we can obtain

$$E[V_2(n_{J+1}T_s - m_{J+1}T_s)|\eta(n_0) = i] \leq \gamma^{J+1} V_2(n_0 T_s - m_0 T_s). \tag{4.61}$$

Then

$$E[\sum_{J=0}^{N} V_2(n_{J+1}T_s - m_{J+1}T_s)|\eta(n_0) = i] \leq \frac{1-\gamma^N}{1-\gamma} V_2(n_0 T_s - m_0 T_s), \tag{4.62}$$

and the limit of (4.62) is

$$\begin{aligned}
&\lim_{N\to\infty} E[\sum_{J=0}^{N} V_2(n_{J+1}T_s - m_{J+1}T_s)|\eta(n_0) = i] \\
&= \lim_{N\to\infty} E[\sum_{J=0}^{N} \mathbf{x}^T(n_{J+1}T_s - m_{J+1}T_s)P_j\mathbf{x}(n_{J+1}T_s - m_{J+1}T_s)|\eta(n_0) = i] \\
&\le \frac{1}{1-\gamma}V_2(n_0T_s - m_0T_s). \qquad (4.63)
\end{aligned}$$

From (4.63) we obtain that

$$\begin{aligned}
&\lim_{N\to\infty} E[\sum_{J=0}^{N} \mathbf{x}^T(n_{J+1}T_s - m_{J+1}T_s)\mathbf{x}(n_{J+1}T_s - m_{J+1}T_s)|\eta(n_0) = i] \\
&\le \frac{1}{\max\rho(P_j)(1-\gamma)}V_2(n_0T_s - m_0T_s). \qquad (4.64)
\end{aligned}$$

The limit of the expectation in (4.64) is bounded, this completed the proof. Hence the closed loop system (4.40) is stochastically stable. □

Note that the LMI condition in (4.41) is non-convex and hence the following theorem is proposed to be the sufficient condition of (4.41).

Theorem 4.3. *For given scalars θ_i, $i = 1, 2, ..., 8$, and a given upper bound of the time varying delay M_bT_s, if there exist symmetric positive definite matrices $\bar{S} > 0, \bar{P} = \begin{bmatrix} \bar{P}_1 & \bar{P}_2 \\ \bar{P}_2^T & \bar{P}_3 \end{bmatrix} > 0$, $\bar{Q} = \begin{bmatrix} \bar{Q}_{11} & 0 \\ 0 & \bar{Q}_{22} \end{bmatrix} > 0$, $\bar{R} = \begin{bmatrix} \bar{R}_{11} & 0 \\ 0 & \bar{R}_{22} \end{bmatrix} > 0$, matrices $\bar{X}$ and*

$\bar{Y}$ *with appropriate dimensions such that the following inequality holds,*

$$\begin{bmatrix} \bar{H}_{11} & * & * & * & * & * & * & * \\ \bar{H}_{21} & \bar{H}_{22} & * & * & * & * & * & * \\ \bar{H}_{31} & \bar{H}_{32} & \bar{H}_{33} & * & * & * & * & * \\ \bar{H}_{41} & \bar{H}_{42} & \bar{H}_{43} & \bar{H}_{44} & * & * & * & * \\ \bar{H}_{51} & \bar{H}_{52} & \bar{H}_{53} & \bar{H}_{54} & \bar{H}_{55} & * & * & * \\ \bar{H}_{61} & \bar{H}_{62} & \bar{H}_{63} & 0 & 0 & \bar{H}_{66} & * & * \\ \bar{H}_{71} & \bar{H}_{72} & \bar{H}_{73} & \bar{H}_{74} & \bar{H}_{75} & \bar{H}_{76} & \bar{H}_{77} & * \\ \bar{H}_{81} & \bar{H}_{82} & \bar{H}_{83} & \bar{H}_{84} & \bar{H}_{85} & \bar{H}_{86} & \bar{H}_{87} & \bar{H}_{88} \end{bmatrix} < 0, \tag{4.65}$$

where

$$\begin{aligned}
\bar{H}_{11} &= \bar{Q}_{11} - \theta_1 A\bar{X} - \theta_1 \bar{X}^T A^T + M_b T_s \bar{R}_{11} \\
\bar{H}_{21} &= \bar{P}_2^T + \bar{N}_1 - \theta_2 A\bar{X}, \quad \bar{H}_{22} = \bar{N}_2 + \bar{N}_2^T + \bar{S} \\
\bar{H}_{31} &= \bar{P}_1 + \theta_1 \bar{X}^T - \theta_3 A\bar{X}, \quad \bar{H}_{32} = \bar{N}_3^T + \theta_2 \bar{X}^T \\
\bar{H}_{33} &= \bar{Q}_{22} + M_b T_s \bar{R}_{22} + \theta_3 \bar{X}^T + \theta_3 \bar{X}, \quad \bar{H}_{41} = -\theta_4 A\bar{X} - \bar{P}_2^T \\
\bar{H}_{42} &= \bar{N}_4^T, \quad \bar{H}_{43} = \theta_4 \bar{X}, \quad \bar{H}_{44} = -\bar{Q}_{11}, \quad \bar{H}_{51} = -\theta_5 A\bar{X} \\
\bar{H}_{52} &= \bar{P}_3 + \bar{N}_5^T, \quad \bar{H}_{53} = \bar{P}_2^T + \theta_5 \bar{X}, \quad \bar{H}_{54} = -\bar{P}_3, \quad \bar{H}_{55} = \frac{-1}{M_b T_s} \bar{R}_{11} \\
\bar{H}_{61} &= -\theta_6 A\bar{X}, \quad \bar{H}_{62} = \bar{N}_6^T, \quad \bar{H}_{63} = \theta_6 \bar{X}, \quad \bar{H}_{66} = \frac{-1}{M_b T_s} \bar{R}_{22} \\
\bar{H}_{71} &= -\bar{N}_1 - \theta_7 A\bar{X}, \quad \bar{H}_{72} = -\bar{N}_2 + \bar{N}_7^T, \quad \bar{H}_{73} = -\bar{N}_3 + \theta_7 \bar{X} \\
\bar{H}_{74} &= -\bar{N}_4, \quad \bar{H}_{75} = -\bar{N}_5, \quad \bar{H}_{76} = -\bar{N}_6, \quad \bar{H}_{77} = -\bar{N}_7^T - \bar{N}_7 \\
\bar{H}_{81} &= -\bar{N}_1 - \theta_8 A\bar{X} - \theta_1 \bar{Y}^T B^T, \quad \bar{H}_{82} = \bar{N}_8^T - \bar{N}_2 - \theta_2 \bar{Y}^T B^T \\
\bar{H}_{83} &= -\bar{N}_3 + \theta_8 \bar{X} - \theta_3 \bar{Y}^T B^T, \quad \bar{H}_{84} = -\bar{N}_4 - \theta_4 \bar{Y}^T B^T, \\
\bar{H}_{85} &= -\bar{N}_5 - \theta_5 \bar{Y}^T B^T \quad \bar{H}_{86} = -\bar{N}_6 - \theta_6 \bar{Y}^T B^T, \\
\bar{H}_{87} &= -\bar{N}_8^T - \bar{N}_7 - \theta_7 \bar{Y}^T B^T \\
\bar{H}_{88} &= -\bar{S} - \bar{N}_8^T - \bar{N}_8 - \theta_8 B\bar{Y} - \theta_8 \bar{Y}^T B^T
\end{aligned}$$

and (4.42) *is satisfied. Then under the static controller with gain obtained by*

$$K = \bar{Y}\bar{X}^{-1}. \tag{4.66}$$

then the system is stochastically stable.

Proof. In order to transform the nonconvex LMI in (4.41) into a solvable LMI, assume that $M_i = \theta_i M_0$ where θ_i is known and given. Define $\bar{X} = M_0^{-1}$,

$$\hat{W} = diag(\bar{X}, \bar{X}, \bar{X}, \bar{X}, \bar{X}, \bar{X}, \bar{X}, \bar{X})$$

and $\bar{Y} = K\bar{X}$. Then by pre-multiplying the inequality in (4.41) by $\hat{W}^T$ and post-multiplying by $\hat{W}$, we can obtain the inequality (4.65). □

Remark 4.2. *Eq.*4.65 *in Theorem.*4.3 *and Eq.*(4.42) *in Theorem.*4.2 *are used to design K. Eq.*(4.65) *is a sufficient condition of Eq.*(4.41) *in Theorem.*4.2 *due to the simplification of M_i matrices.*

4.5 Summary

This chapter mainly dealt with the stabilization problem of NCSs with bounded time varying delays and Markovian packet loss via sampled-data control approach. The system can be stochastically stabilized according to the proper design of the static feedback controller. Lyapunov method and LMI techniques were applied to ensure the stochastic stability of the networked control systems. A real-time network induced delay and packet loss measurement system has been built and experiments were proceeded to get the real network characteristics.

Chapter 5

Distributed Consensus Formation Control of Networked Multi-agent Robotic Systems with Time Delays

5.1 Introduction

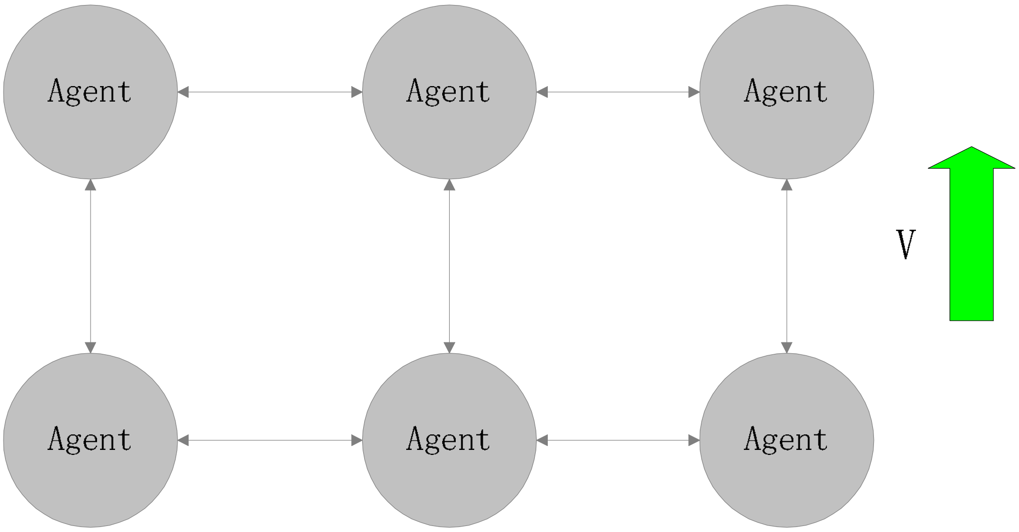


Figure 5.1. Consensus of multiple agent system

Investigation for multi-agent systems begins with studying the behavior of a large number of interacting agents with a common group objective. These agents include fish, ants and bees. Take the fish in a flock as an example, the desired track for each agent in the group has been already decided according to some external elements.

Each agent should track the existent desired trajectory and acquire some information, such as the velocities and relative positions, from its neighbors in order to: *i*) stay in a proper distance to nearby flock-mates; *ii*) avoid collisions; *iii*) match velocity with each other. Then all agents asymptotically move with the same velocity and form a cohesive flock without collisions (See Fig.5.1). This phenomena is called consensus. Notice that in the whole group movement there is no one performs as the"leader" because of a widely accepted opinion by animal behavior scientists that"schools need no leaders" [59].

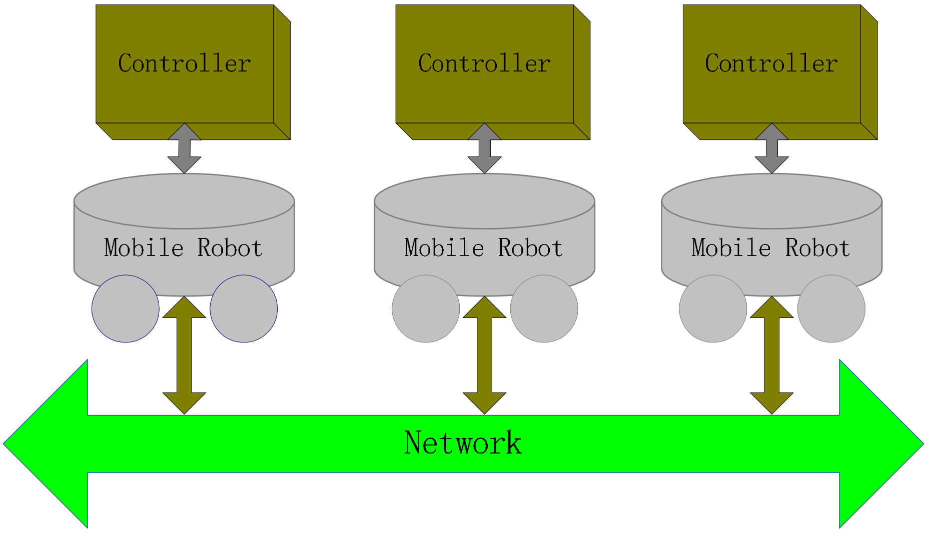


Figure 5.2. Linear multi-agent networked robotic system

In this work, a peer-to-peer architecture without leaders is used to solve a kind of tracking problem of the linear multi-agent networked robotic vehicle systems. Fig.5.2 shows the architecture of the linear multi-agent networked robotic vehicle system. The agents in the systems are connected and share information with their neighbors via networks with constant networked induced time delays. Each robot has a local controller applied. It can sense its own position and orientation in a special coordinate by sensors and send those information to its neighbors via networks. Every agent in the system played the same role as the one in the flock of fish example. It moves at the same velocity as its neighbors and tracks its reference route which is parallel with references of others in the flock. Then the whole flock will start moving from place A to place B. During the movement the flock should reach consensus to present the properties of the "Flock Movement". This model has wide application possibilities, such as the operation of the unpiloted combine harvester in agriculture. The research work done in this chapter is mainly for dealing with this operation and the main results are going to be applied in the real engineering field. The harvester and the truck could be treated as two agents in a multiple system. If using lead-follower architecture to tackle this problem, the task could be done if the leader tracks its own desired route on the field and the follower tracks leader's route with a special angle perfectly. However, people found this architecture has poor disturbance rejection properties [78] which could cause failure for the task. Instead of this way, in this chapter peer-to-peer architecture is chosen to improve the stability of the whole group. In the proposed design the harvester and the truck are supposed to have their own desired trajectories which were made according to the working route on the field so that this architecture could avoid depending heavily on one single agent and avoid being poor performance in adversarial environments. Those two reference trajectories have fixed relative position with each other so the two agents could keep a fixed relative position during the movement. They need to share their state information with each other to make the whole multiple system reach consensus. In [65], the velocity and orientation of the flock movement is random,. The aim is to make the flock reach consensus during the movement. But in this design the flock movement has known origin and destination. Each robot is supposed to exchange information with at least one robot in the group which is treated as its neighbor. In order to tackle the stabilization problem, the system has been modeled by using an error dynamic.

The objective of this work is designing the consensus based formation control algorithm to make the system achieve consensus so that: *i*) each agent tracks its reference trajectory well; *ii*) the agents move at the same velocity; *iii*) the agents keep a proper distance with each other to make the group formations. In order to tackle this problem, a distributed control algorithm which includes a relative position term and a navigational feedback term has been introduced to stabilize the multiple agent system. A new term called collision avoidance has been added to the control algorithm to avoid collisions during the group movement. The Lyapunov analysis method has been used to derive the sufficient conditions for stabilizing controller design. The effects caused by the network induced delay have been considered in this design.

5.2 Problem Formulation

In this section, the multiple coordinate system is introduced. First suppose there are n mobile robots moving on a plane. Each robot is labelled as R_i. To position all the robots in a common coordinate system, a static coordinate system C_0 is defined as the common position reference for all the robots. In order to make group formations each of the mobile robots needs to know its relative position and relative orientation to others. So each robot needs its own coordinate system which is denoted with C_i.

With the help of modern communication techniques such as Global Position Systems (GPS) ot relative positioning systems, C_0 and C_i can be known to R_i. During the group movement, R_i senses its position and velocity in C_0 and sends these information to its neighbor R_j. By using the information, R_j can get the relative position and velocity to R_i in C_j. All robots will sense the relative position to others in this way during the movement. The sketch map of multiple coordinates is as shown in Fig.5.3

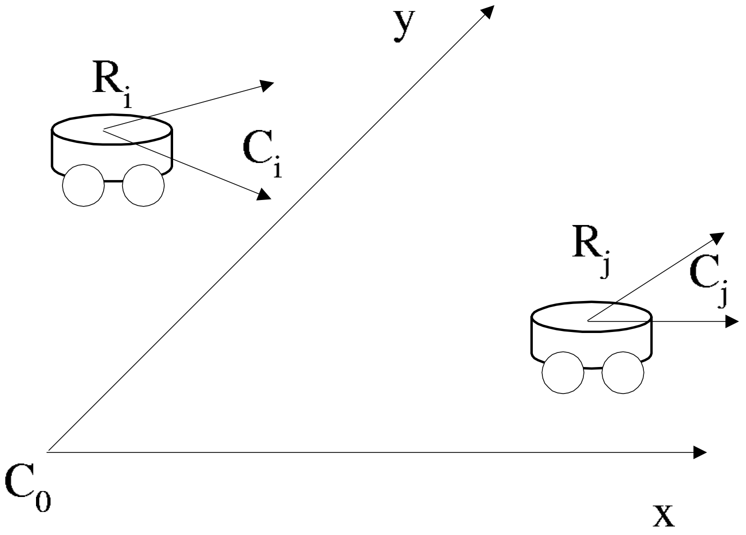


Figure 5.3. Multiple coordinate system

The linear networked multi-agent robotic vehicle system as shown in Fig.5.3 can be expressed in the following equation:

$$\dot{\mathbf{x}}_i(t) = A_{ii}\mathbf{x}_i(t) + \sum_{i \neq j} A_{ij}[\mathbf{x}_j(t-\tau) - \Delta_j] + B\mathbf{u}_i(t), i \in m, j \in n_i \tag{5.1}$$

where $\mathbf{x}_i(t) \in \Re^n$ represents the error of the ith robot between the present states of R_i and its reference at time t. The states of R_i contain the position information of R_i in C_0. $\mathbf{u}_i(t) \in \Re^{m'}$ represents the control input. $\mathbf{x}_j(t-\tau) \in \Re^n$ represents the difference of states between R_i and R_j, which means the position of R_j in C_i. τ is the known constant time delay. Δ_j is the desired relative position of R_j to R_i, m is the total corpora of all the robots, n_i is the subset of all the neighbors of R_i. A_{ii},A_{ij} and B are known constant matrices with appropriate dimensions. A_{ij} is the attraction coefficient matrix of R_i to R_j, physically it means that R_i is attracted to R_j. A_{ij} could be zero in which case the linear dynamic is the same as the one used in [65]. The objective is to design the controller to stabilize the linear multi-agent networked robotic systems so that the system could reach consensus.

5.3 Distributed Control Design

Now consider the following distributed control algorithm:

$$\begin{aligned}\mathbf{u}_i(t) &= \sum_{i\neq j, j\in m_i} K_{ij}[\mathbf{x}_j(t-\tau)-\Delta_j] + K_{ii}\mathbf{x}_i(t) \\ &\quad + \sum_{i\neq j, j\in m_c} L_{ij}[1-\frac{\Delta}{\|\mathbf{x}_j(t-\tau)\|}]\mathbf{x}_j(t-\tau) \\ L_{ij} &= \{ \begin{array}{ll} L, & \|\mathbf{x}_j(t-\tau)\| \leq \Delta, j\in m_c \\ 0, & \|\mathbf{x}_j(t-\tau)\| > \Delta, j\in m_i \end{array} \end{aligned} \tag{5.2}$$

where Δ is the collision warning indicating distance between two robots. It means that R_i is repulsed from R_j as an artificial potential field exists [11]. When the relative distance of R_i to R_j is shorter than Δ, there is going to be a possible collision between them and the repulsion works. L_{ij} is the repulsion coefficient of R_i from R_j, the scalar L is positive. m_i is the subset of all the R_js which stays in a distance farther than Δ to R_i; m_c is the subset of all the R_js which stays in a distance shorter than Δ to R_i. Note that $m_i \bigcap m_c = 0$ and $m_i \cup m_c = n_i$. K_{ii} and K_{ij} are the control gain to be designed. Let $\gamma = 1 - \frac{\Delta}{\|\mathbf{x}_j(t-\tau)\|} < 0$ and

$$\bar{\mathbf{x}}_j(t-\tau) = \begin{bmatrix} \gamma \mathbf{x}_{\hat{j}}(t-\tau) \\ \vdots \\ \mathbf{x}_{\check{j}}(t-\tau) - \Delta_j \end{bmatrix}_{(m-1)\times 1} \tag{5.3}$$

where m is the total corpora of all the robots, $\hat{j} \in m_c$ and $\check{j} \in m_i$. Then apply the control law and (5.3) on (5.1) the closed loop system becomes

$$\dot{\mathbf{x}}_i(t) = (A_{ii} + K_{ii})\mathbf{x}_i(t) + [\ldots, L_{ij}, \ldots, A_{ij} + K_{ij}, \ldots]_{j \neq i, j \in n_i} \bar{\mathbf{x}}_j(t-\tau), \tag{5.4}$$

Lemma 5.1. - *Jensen Inequality For any constant matrix $E \in R^{n\times n}$, $E = E^T > 0$, vector function $\boldsymbol{\omega} : [0, \tau] \to R^n$ such that the integrations concerned are well defined, then,*

$$\tau \int_0^\tau \boldsymbol{\omega}^T(s) E \boldsymbol{\omega}(s) ds \geq \left[\int_0^\tau \boldsymbol{\omega}(s) ds\right]^T E \left[\int_0^\tau \boldsymbol{\omega}(s) ds\right]. \tag{5.5}$$

Theorem 5.1. *Consider the linear multi-agnet robotic system* (5.1)*, for a given time delay τ, if there exist symmetric positive definite matrices $diag[Q_1, \ldots, Q_{m-1}] > 0$, $diag[R_1, \ldots, R_{m-1}] > 0$, matrices $M_i, N_i, i \in m$ with appropriate dimensions and a scalar $L > 0$, such that the following inequality holds*

$$H = \begin{bmatrix} H_{11} & * & * & * & * & * \\ H_{21} & H_{22} & * & * & * & * \\ 0 & 0 & H_{33} & * & * & * \\ H_{41} & H_{42} & H_{43} & H_{44} & * & * \\ 0 & 0 & 0 & 0 & H_{55} & * \\ 0 & 0 & H_{63} & H_{64} & 0 & H_{66} \end{bmatrix} < 0, \tag{5.6}$$

where

$$\begin{aligned}
H_{11} &= M_1^T A_{ii} + A_{ii}^T M_1 + M_1^T B K_{ii} + K_{ii}^T B^T M_1, \\
H_{21} &= P + M_2^T A_{ii} + M_2^T B K_{ii} - M_1, \\
H_{22} &= -M_2^T - M_2, \quad H_{33} = diag[Q_1 + \tau R_1, \ldots, Q_{m-1} + \tau R_{m-1}] \\
&\quad + diag[N_{11}^T + N_{11}, \ldots, N_{1(m-1)}^T + N_{1(m-1)}], \\
H_{41} &= \begin{bmatrix} M_{31}^T \\ \vdots \\ M_{3(m-1)}^T \end{bmatrix} \cdot (A_{ii} + B K_{ii}) + \begin{bmatrix} L_{ij}^T \\ \vdots \\ A_{ij}^T + K_{ij}^T B^T \end{bmatrix} \cdot M_1, \\
H_{42} &= -\begin{bmatrix} M_{31}^T \\ \vdots \\ M_{3(m-1)}^T \end{bmatrix} + \begin{bmatrix} L_{ij}^T \\ \vdots \\ A_{ij}^T + B^T K_{ij}^T \end{bmatrix} \cdot M_2, \\
H_{43} &= \begin{bmatrix} N_{31}^T - N_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{3(m-1)}^T - N_{1(m-1)} \end{bmatrix}, \\
H_{44} &= -\begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{m-1} \end{bmatrix} + \begin{bmatrix} M_{31}^T \\ \vdots \\ M_{3(m-1)}^T \end{bmatrix} \cdot [L_{ij}, \ldots, A_{ij} + B K_{ij}] \\
&\quad + \begin{bmatrix} L_{ij}^T \\ \vdots \\ A_{ij}^T + K_{ij}^T B^T \end{bmatrix} \cdot [M_{31}, \ldots, M_{3(m-1)}] \\
&\quad - \begin{bmatrix} N_{31}^T + N_{31} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{3(m-1)}^T + N_{3(m-1)} \end{bmatrix},
\end{aligned}$$

$$H_{55} = -\frac{1}{\tau}\begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{m-1} \end{bmatrix} \quad H_{63} = \begin{bmatrix} N_{21}^T - N_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{2(m-1)}^T - N_{1(m-1)} \end{bmatrix},$$

$$H_{64} = -\begin{bmatrix} N_{21}^T + N_{31} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{2(m-1)}^T + N_{3(m-1)} \end{bmatrix},$$

$$H_{66} = -\begin{bmatrix} N_{21}^T + N_{21} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{2(m-1)}^T + N_{2(m-1)} \end{bmatrix}, \tag{5.7}$$

then system (5.1) *is asymptotically stable, e.g.* $\boldsymbol{x}_i(t)$ *tends to zero asymptotically which means* R_i *tracks its desired trajectory well.*

Proof. Now consider the following Lyapunov functional candidate:

$$\begin{aligned} V_i(t) &= \mathbf{x}_i^T(t)P\mathbf{x}_i(t) + \int_{t-\tau}^{t} \bar{\mathbf{x}}_j^T(s)\begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(s)ds \\ &\quad + \int_{-\tau}^{0}\int_{t+\theta}^{t} \bar{\mathbf{x}}_j^T(s)\begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(s)ds, \end{aligned} \tag{5.8}$$

Take the derivative of (5.8) and use Lemma 5.1

$$\begin{aligned} \dot{V}_i(t) &= \dot{\mathbf{x}}_i^T(t)P\mathbf{x}_i(t) + \mathbf{x}_i^T(t)P\dot{\mathbf{x}}_i(t) + \bar{\mathbf{x}}_j^T(t)\begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(t) \\ &\quad -\bar{\mathbf{x}}_j^T(t-\tau)\begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(t-\tau) + \tau\bar{\mathbf{x}}_j^T(t) \\ &\quad \begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(t) - \int_{t-\tau}^{t} \bar{\mathbf{x}}_j^T(s)\begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{m-1} \end{bmatrix} \\ &\quad \bar{\mathbf{x}}_j(s)ds \end{aligned}$$

$$\begin{aligned}
\leq\ & \dot{\mathbf{x}}_i^T(t)P\mathbf{x}_i(t) + \mathbf{x}_i^T(t)P\dot{\mathbf{x}}_i(t) + \bar{\mathbf{x}}_j^T(t)\begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(t) \\
& -\bar{\mathbf{x}}_j^T(t-\tau)\begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(t-\tau) + \tau\bar{\mathbf{x}}_j^T(t) \\
& \begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{m-1} \end{bmatrix}\bar{\mathbf{x}}_j(t) - \frac{1}{\tau}\int_{t-\tau}^{t}\bar{\mathbf{x}}_j^T(s)ds\begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{m-1} \end{bmatrix} \\
& \int_{t-\tau}^{t}\bar{\mathbf{x}}_j(s)ds, \qquad (5.9)
\end{aligned}$$

Now with proper dimensions, the following equations holds:

$$\begin{aligned}
\Phi_1 &= 2Z_1^T M^T\{-\dot{\mathbf{x}}_i(t) + (A_{ii} + K_{ii})\mathbf{x}_i(t) + \\
&\quad [\ldots, L_{ij}, \ldots, A_{ij} + K_{ij}, \ldots]_{j\neq i, j\in n_i}\bar{\mathbf{x}}_j(t-\tau)\} = 0 \\
\Phi_2 &= 2Z_2^T[N_1^T, N_2^T, N_3^T]^T[\bar{\mathbf{x}}_j^i(t) - \int_{t-\tau}^{t}\dot{\bar{\mathbf{x}}}_j^i(s)ds - \bar{\mathbf{x}}_j^i(t-\tau)] = 0, \\
Z_1 &= [\mathbf{x}_i(t), \dot{\mathbf{x}}_i(t), \bar{\mathbf{x}}_j(t-\tau)]^T, \\
Z_2 &= [\bar{\mathbf{x}}_j(t), \int_{t-\tau}^{t}\dot{\bar{\mathbf{x}}}_j(s)ds, \bar{\mathbf{x}}_j(t-\tau)]^T, \\
M &= \{M_1, M_2, [M_{31}, \ldots, M_{3(m-1)}]\}, \\
N_1 &= \begin{bmatrix} N_{11}, & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{1(m-1)} \end{bmatrix}, \\
N_2 &= \begin{bmatrix} N_{21}, & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{2(m-1)} \end{bmatrix}, \\
N_3 &= \begin{bmatrix} N_{31}, & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & N_{3(m-1)} \end{bmatrix}, \qquad (5.10)
\end{aligned}$$

Then

$$\dot{V}_i(t) + \Phi_1 + \Phi_2 \leq \mathbf{Z}^T H\mathbf{Z} < 0, \qquad (5.11)$$

where $\mathbf{Z} = [\mathbf{x}_i(t), \dot{\mathbf{x}}_i(t), \bar{\mathbf{x}}_j(t), \bar{\mathbf{x}}_j(t-\tau), \int_{t-\tau}^{t}\bar{\mathbf{x}}_j(s)ds, \int_{t-\tau}^{t}\dot{\bar{\mathbf{x}}}_j(s)ds]^T$ and H is as shown in (5.6). □

The LMI condition in (5.6) is non-convex and hence the following theorem is proposed to be the equivalent sufficient condition.

Theorem 5.2. *For given scalars θ_i, $i \in M$, and a given time delay constant τ, if there exist symmetric positive definite matrices $diag[\bar{Q}_1, \ldots, \bar{Q}_{m-1}]$, $diag[\bar{R}_1, \ldots, \bar{R}_{m-1}]$, matrices $\bar{M}_i, \bar{N}_i, i \in m$ with appropriate dimensions and a scalar $L > 0$, nonsingular matrix X with appropriate dimensions such that the following inequality holds,*

$$\bar{H} = \begin{bmatrix} \bar{H}_{11} & * & * & * & * & * \\ \bar{H}_{21} & \bar{H}_{22} & * & * & * & * \\ 0 & 0 & \bar{H}_{33} & * & * & * \\ \bar{H}_{41} & \bar{H}_{42} & \bar{H}_{43} & \bar{H}_{44} & * & * \\ 0 & 0 & 0 & 0 & \bar{H}_{55} & * \\ 0 & 0 & \bar{H}_{63} & \bar{H}_{64} & 0 & \bar{H}_{66} \end{bmatrix} < 0, \tag{5.12}$$

where

$$\begin{aligned}
\bar{H}_{11} &= \theta_1^T A_{ii} X + \theta_1 X^T A_{ii}^T + \theta_1^T B Y_{ii} + \theta_1 Y_{ii}^T B^T, \\
\bar{H}_{21} &= \bar{P} + \theta_2 A_{ii} X + \theta_2 B Y_{ii} - \theta_1 X^T \\
\bar{H}_{22} &= -\theta_2 X^T - \theta_2 X, \quad \bar{H}_{33} = diag[\bar{Q}_1 + \tau \bar{R}_1, \ldots, \bar{Q}_{m-1} + \tau \bar{R}_{m-1}] \\
&\quad + diag[\bar{N}_{11}^T + \bar{N}_{11}, \ldots, \bar{N}_{1(m-1)}^T + \bar{N}_{1(m-1)}], \\
\bar{H}_{41} &= \begin{bmatrix} \theta_{31} \\ \vdots \\ \theta_{3(m-1)} \end{bmatrix} \cdot (A_{ii} X + B Y_{ii}) + \begin{bmatrix} \theta_1 X^T L_{ij}^T \\ \vdots \\ \theta_1 X^T A_{ij}^T + Y_{ij}^T B^T \end{bmatrix} \\
\bar{H}_{42} &= - \begin{bmatrix} \theta_{31} X \\ \vdots \\ \theta_{3(m-1)} X \end{bmatrix} + \begin{bmatrix} \theta_2 X^T L_{ij}^T \\ \vdots \\ \theta_2 X^T A_{ij}^T + Y_{ij}^T B^T \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
\bar{H}_{43} &= \begin{bmatrix} \bar{N}_{31}^T - \bar{N}_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{N}_{3(m-1)}^T - \bar{N}_{1(m-1)} \end{bmatrix} \\
\bar{H}_{44} &= -\begin{bmatrix} \bar{Q}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{Q}_{m-1} \end{bmatrix} + \begin{bmatrix} \theta_{31} \\ \vdots \\ \theta_{3(m-1)} \end{bmatrix} \cdot [L_{ij}X, \ldots, A_{ij}X + BY_{ij}] \\
&\quad + \begin{bmatrix} X^T L_{ij}^T \\ \vdots \\ X^T A_{ij}^T + Y_{ij}^T B^T \end{bmatrix} \cdot [\theta_{31}, \ldots, \theta_{3(m-1)}] \\
&\quad - \begin{bmatrix} \bar{N}_{31}^T + \bar{N}_{31} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{N}_{3(m-1)}^T + \bar{N}_{3(m-1)} \end{bmatrix} \\
\bar{H}_{55} &= -\frac{1}{\tau} \begin{bmatrix} \bar{R}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{R}_{m-1} \end{bmatrix} \\
\bar{H}_{63} &= \begin{bmatrix} \bar{N}_{21}^T - \bar{N}_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{N}_{2(m-1)}^T - \bar{N}_{1(m-1)} \end{bmatrix} \\
\bar{H}_{64} &= -\begin{bmatrix} \bar{N}_{21}^T + \bar{N}_{31} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{N}_{2(m-1)}^T + \bar{N}_{3(m-1)} \end{bmatrix} \\
\bar{H}_{66} &= -\begin{bmatrix} \bar{N}_{21}^T + \bar{N}_{21} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{N}_{2(m-1)}^T + \bar{N}_{2(m-1)} \end{bmatrix} \qquad (5.13)
\end{aligned}
$$

then the K_{ii} and K_{ij} matrices in Theorem 1 are obtained as

$$K_{ii} = Y_{ii} X^{-T}, \qquad K_{ij} = Y_{ij} X^{-T}. \tag{5.14}$$

As a result, the system (5.1) *is asymptotically stable, e.g.* $\boldsymbol{x}_i(t)$ *tends to zero asymptotically which means R_i tracks its desired trajectory well.*

Proof: In order to transform the nonconvex LMI in (4.41) into a solvable LMI, assume that there are some relations in M_i's, $i \in M$. One possibility is that $M_i = \theta_i M_0$ where M_0 is nonsingular and θ_i is known and given. Define $X = M_0^{-1}$, $W = diag(X, X, X, X, X, X)$ and $Y_{ii} = K_{ii} X^T$, $Y_{ij} = K_{ij} X^T$. Then pre-multiplying the inequality in (5.6) by W and post-multiplying by W^T, the inequality in (5.12) can be obtained. Note that the inequality in (5.12) is only a sufficient condition for the solvability of (5.7) based on these derivations and $\bar{Q}_i = X^T Q_i X$, $\bar{R}_i = X^T R_i X$, $\bar{N}_{ij} = X^T N_{ij} X$.

Remark 5.1. *i,j might be in a bound. If i,j is too big, there will be too many given scalars θ_i in 5.12 and it will be difficult to solve 4.42. The interaction topology was affected by the interactive weight matrix A_{ij}, if $A_{ij} = 0$ it means there is no connection between agent i and j. If 5.12 can not be solved with relative weight matrices, then the system can not reach consensus.*

5.4 Summary

This work dealt with the consensus problem for networked multi-agent robotic vehicle systems. The advantage of the peer-to-peer architecture which was used to model the system has been discussed. The multi-coordinate system has also been introduced to explain the system model and communications in the group. Each robotic vehicle had its own coordinate system, and sent its position and orientation to its neighbors.

The chapter presents a novel distributed feedback control algorithm that guarantees the stability of the system by feedback, such as relative position to others. Sufficient conditions for controller design were given by analyzing the Lyapunov functional candidate. The system can be stabilized by the distributed feedback controller via the interaction among robotic vehicles. The future work will focus on testing this approach into real robotic vehicles located in the research lab. Each robotic vehicle is capable of sensing its relative position and orientation in C_0 and sending those information to others via wireless network.

Chapter 6

Consensus Formation Control of Multiple Mobile Robots

In this chapter, the problem for a kind of consensus formation control of networked multi-agent robotic systems has been tackled. The outline of this chapter is as follows: in Section.6.1, the basic idea and main background information about formation control on networked multi-robotic systems have been introduced; in section.6.2, the background information of Pioneer 3 mobile robots used to test the control design methodologies in this work has been introduced; in Section.6.3, the kinematic model equations of Pioneer 3 robots have been studied; in Section.6.4, the proposed formation control design methodologies based on consensus algorithms have been introduced. The summary of this work is made in Section.6.5.

Notations: $\|\mathbf{x}\|$ is the norm defined as $\| \cdot \| = \sqrt{\mathbf{x}^T\mathbf{x}}$, where $\mathbf{x}$ is a vector. The information exchange among robots is usually modelled by graphs. Suppose that a team consists of n mobile robots. $Ç_n = (\nu_n, \varepsilon_n)$ is a graph, where $\nu_n = 1, ..., n$ is a finite nonempty node set and $\varepsilon_n \subseteq \nu_n \times \nu_n$ is an edge set of ordered pairs of nodes, called edges. The adjacency matrix $A_n = [a_{ij}] \in R^{n \times n}$ of a graph $Ç_n = (\nu_n, \varepsilon_n)$ is defined such that a_{ij} is a positive weight if $(j, i) \in \varepsilon_n$ is ture, and $a_{ij} = 0$ if $(j, i) \in \varepsilon_n$

is false.

6.1 Introduction

As the embedded computational resources in autonomous robotic vehicles become abundance, the enhanced operational effectiveness through cooperative teamwork in civilian and military applications have been enabled. Compared to autonomous robotic vehicles that operate single tasks, cooperative teamwork has greater efficiency and operational capability. Multi-robotic vehicle systems have many potential applications, such as platooning of vehicles in urban transportation, the operation of the multiple robots, autonomous underwater vehicles and formation of aircrafts in military affairs.

For multi-robotic vehicle systems, the group cooperative behavior is the essential study objective. Group cooperative behavior signifies that individuals in the group share a common objective and action according to the interest of the whole group. Group cooperation can be efficient if individuals in the group coordinate their actions well. Each individual can coordinate with other individuals in the group to facilitate group cooperative behavior in two ways which are local coordination and global coordination. For local coordination, individuals react only to other individuals that are close, such as fish engaged in a school. For global coordination, each individual can directly coordinate its act with every other individual in the group. Due to communication constraints, most researchers are interested primarily in group cooperation problems where the coordination occurs locally.

Cooperative control of multi-robotic vehicle systems bring us significant theoretical and practical challenges, such as the research objective is defined based on a system of some subsystems rather than a single system, the effects caused by the communication constraints should be considered, and how to design coordination strategies so that coordination will result in group cooperation.

As a concrete example of cooperative control, formation control of multiple autonomous vehicles receives significant interest in recent years. It requires that autonomous robotic vehicles collectively maintain a prescribed geometric shape during movement. Maintaining an accurate geometric configuration among multiple robotic vehicles moving in formation can result in less expensive and more capable systems that can accomplish objectives impossible for a single vehicle. The advantages of formation control for multi-robotic systems are summarized as follows: good feasibility, accuracy, robustness, flexibility, lower cost, energy efficiency and probability of success. For example, a group of robotic vehicles can be used for large objective transferring, terrain model reconnaissance, unknown area exploration and path obstruction.

Various strategies and approaches, which can be roughly categorized as leader-follower, behavioral, and virtual leader approaches, have been proposed for formation control. In leader-follower approach, one of the vehicles is appointed as the leader, other vehicles in the group are appointed as the followers. When the group cooperative behavior occurs, the followers should track the trajectory of the leader with some prescribed offset. The basic idea about the behavioral approach is to prescribe several desired behaviors for each vehicle and to make the control action to each vehicle according to each behavior. In the virtual leader structure approach, the entire formation of the group is treated as a single structure. The virtual leader's dynamic or trajectory is converted as the desired action of each vehicle in the group. Tracking control based on consensus algorithm is then needed to tackle this problem.

In this work, the consensus-based design scheme has been applied to formation control of multiple wheeled mobile robot group with a virtual leader. The group communication configuration is assumed to be a fully coupled system which means decisions made by each robot in the group affect the cost and outcomes of all other members of the group. In this case, what a single robot is going to do is affected by what all other robots in the group are going to do. The distributed formation control architecture has been defined to accommodate an arbitrary number of subgroup leaders and arbitrary information flow among the robots. This architecture requires neighbor-to-neighbor information exchange. On the group level the consensus tracking algorithm is applied to guarantee consensus on the time-varying group reference trajectory in a distributed manner. A consensus-based formation control strategy developed based on the group level consensus tracking algorithm is applied for vehicle level control. A novel delay-dependent multiple Lyapunov functional candidate related to LKF has been constructed to investigate the convergence of the tracking error. The null sums have been added to the new multiple LKF with free weighting matrices introduced to reduce the conservatism in the derivation of the stability conditions. The matrices and the sufficient conditions for stabilization of the proposed control approach are determined by solving LMIs. The effects caused by the group communication delay also have been considered in the proposed approach. The proposed control strategy is experimentally implemented for multiple wheeled mobile robots under neighbor-to-neighbor information exchange with group communication delay involved.

6.2 Pioneer 3 Mobile Robots

Pioneer mobile robots are durable, differential-drive robots for academic researchers. The most famous advantages of Pioneer 3 robot series are good versatility, reliability and durability. In this work two kinds in the series named Pioneer 3-DX ($P3 - DX$) and Pioneer 3-AT ($P3 - AT$) have been used for the experimental implementation.

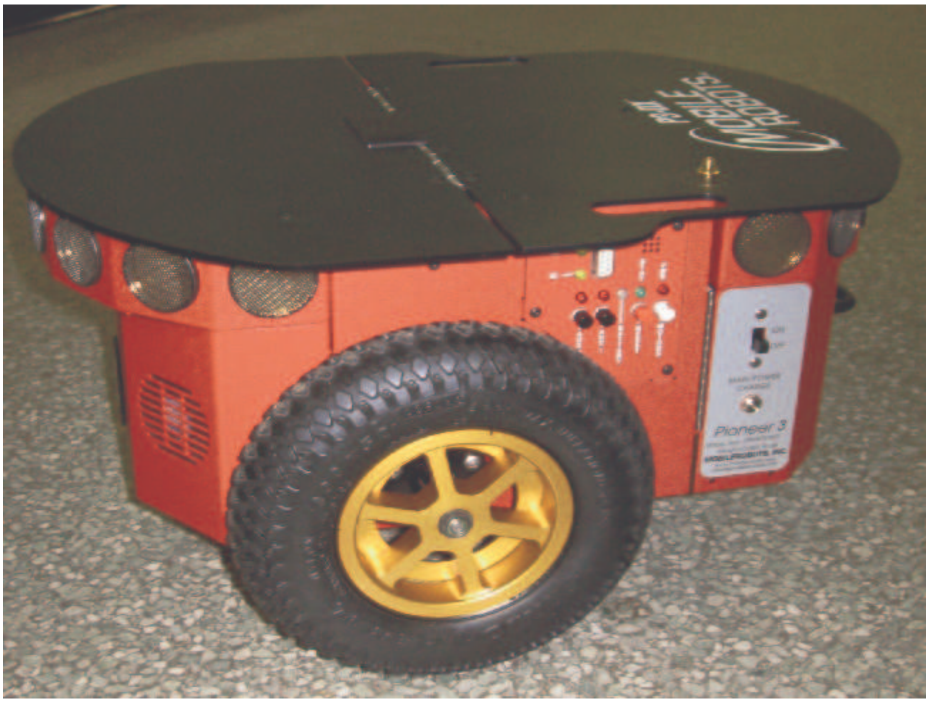


Figure 6.1. Pioneer 3 DX mobile robot

The $P3 - DX$ used in the lab is shown in the Fig.6.1 It has assembled motors with 500-tick encoders, 19cm wheels, 8 forward-facing ultrasonic sensors, 8 real-facing sonar, 1, 2 or 3 rechargeable batteries, and a micro-controller which can communicate with a laptop through the serial port. $P3 - DX$ can reach the maximum speed of 1.6

meters per second and carry a payload of up to 23 kg. It is an all-purpose base and can be used for research and applications involving mapping, teleoperation, localization, monitoring, reconnaissance,vision capture, cooperation and other behaviors. $P3-DX$ runs best on hard surfaces. It can traverse low sills and household power cords, it can also climb most wheelchair ramps.

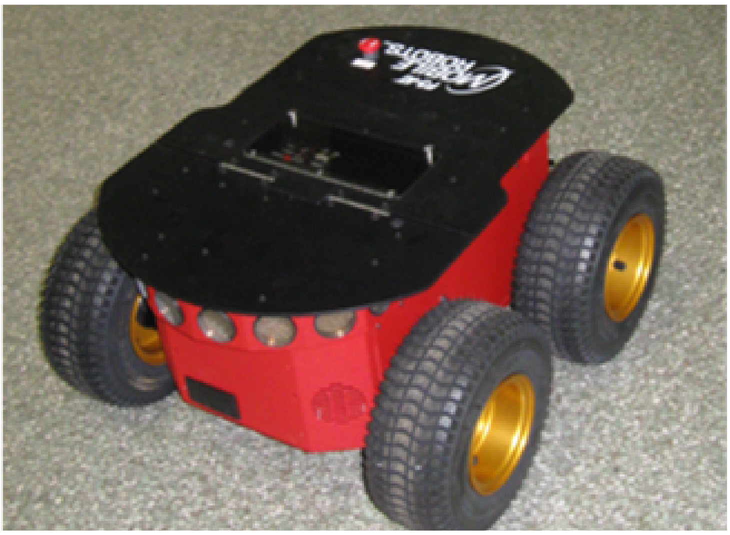


Figure 6.2. Pioneer 3 AT mobile robot

Another robot $P3 - AT$ used in this work is shown in Fig.6.2. It is a highly versatile four wheel drive robotic platform, which is software-compatible with other Pioneer 3 robots. $P3 - AT$ is a popular team performer for outdoor or rough-terrain projects. It has powerful motors and four knobby wheels that can reach the maximum speed of 0.8 meters per second and carry a payload of up to 12 kg. $P3 - AT$ uses 100 tick encoders with inertial correction recommended for dead reckoning to compensate for skid steering. Similar with $P3 - DX$, $P3 - AT$ also has 8 forward and 8 rear sonar, a micro-controller which can be connected with a laptop through serial port and batteries. It can be used for all the applications of $P3 - DX$.

Both of $P3-DX$ and $P3-AT$ have the same kinematic model which can be expressed by the following equation:

$$\begin{aligned} \dot{x} &= v\cos(\theta), \\ \dot{y} &= v\sin(\theta), \\ \dot{\theta} &= \omega, \end{aligned} \tag{6.1}$$

where $[x, y]$ is the inertial position of the $P3$ mobile robot, θ is the orientation of the robot and $[v, \omega]$ denote the linear and angular speeds of the robot. Since the $P3$ mobile robots used in this work has nonholonomic constraints, the coordination problem becomes more complicated. Since nonholonomic systems cannot be stabilized with continuous static state feedback, so the difficulty of the coordination problem for differentially driven mobile robots is that the position and orientation of the center of the robot cannot be simultaneously stabilized with a time-invariant feedback control strategy. Some researchers successfully used discontinuous control laws and time varying control laws to stabilize the center of a single differentially driven mobile robot, however, the multiple robot case is more complicated.

The most popular way to simplify this complex case is to define a hand position for each robot. As shown in Fig.6.3, the hand position of the robot usually has been defined at the point $h = [h_x, h_y]^T$ which lies a distance L along the line that is normal to the wheel axis and intersects the wheel axis at the center point $r = [r_x, r_y]^T$. The kinematic model of the hand position are holonomic for $L \neq 0$. Instead of considering the coordinating problem at the center of the robot, the problem at the hand position has been considered. Another important advantage of defining a hand position for

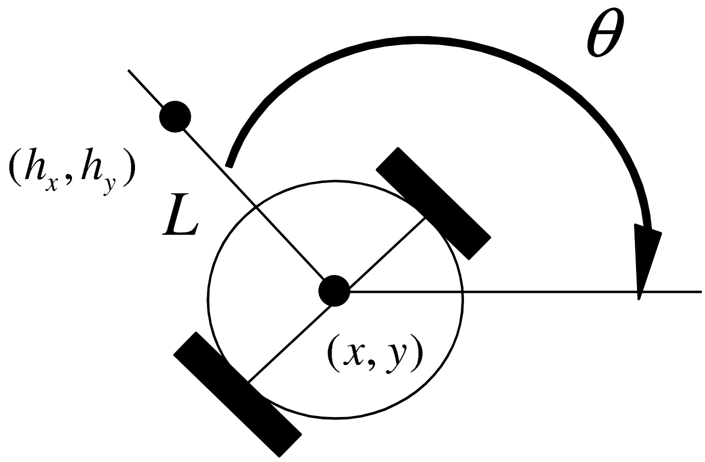


Figure 6.3. Hand position for P3 mobile robot

the robot is that the hand position has practical interest. For example, if the task of the robot group is to move an object from one place to another by using the gripper which has been installed at the hand position of each robot, then the control objective for this task is to move the gripper locations in a coordinated fashion.

Now let's find the kinematic model for the hand position of each robot. First the hand position can be represented by the following equation:

$$\begin{aligned} h_x &= x + L\cos(\theta) \\ h_y &= y + L\sin(\theta). \end{aligned} \tag{6.2}$$

where $\mathbf{h} = [h_x, h_y]^T$ is the hand position in x and y plane. Now let's differentiate (6.2) with respect to time and substitute (6.1), then

$$\begin{aligned} \dot{h}_x &= \cos(\theta)\upsilon - L\sin(\theta)\omega \\ \dot{h}_y &= \sin(\theta)\upsilon + L\cos(\theta)\omega. \end{aligned} \tag{6.3}$$

Define

$$\begin{aligned} \mathbf{h} &= [h_x, h_y]^T \\ \mathbf{u} &= [u_x, u_y]^T, \end{aligned} \tag{6.4}$$

and let

$$\begin{aligned} v &= \cos(\theta)u_x + \sin(\theta)u_y \\ \omega &= -\frac{1}{L}\sin(\theta)u_x + \frac{1}{L}\cos(\theta)u_y, \end{aligned} \tag{6.5}$$

then

$$\dot{\mathbf{h}} = \mathbf{u}, \tag{6.6}$$

which is the kinematic model of the robot's hand position.

6.3 Problem Formulation

As described in the above section, the nonlinear kinematic model of the center position of the robot has been simplified and linearized into the form of single-integrator dynamics shown by (6.6) by defining a hand position for each robot. The control interest has been converted from the center position of the robot to its hand position.

As discussed in subsection.6.1, the virtual leader and virtual structure approach is one solution to formation control. Fig..6.4 shows the example of the virtual leader and virtual structure approach with a formation composed of two vehicles with planar motions. In Fig.6.4, C_0 is the global coordinate system, $r_i(t) = [x_i(t), y_i(t)]^T$ is the

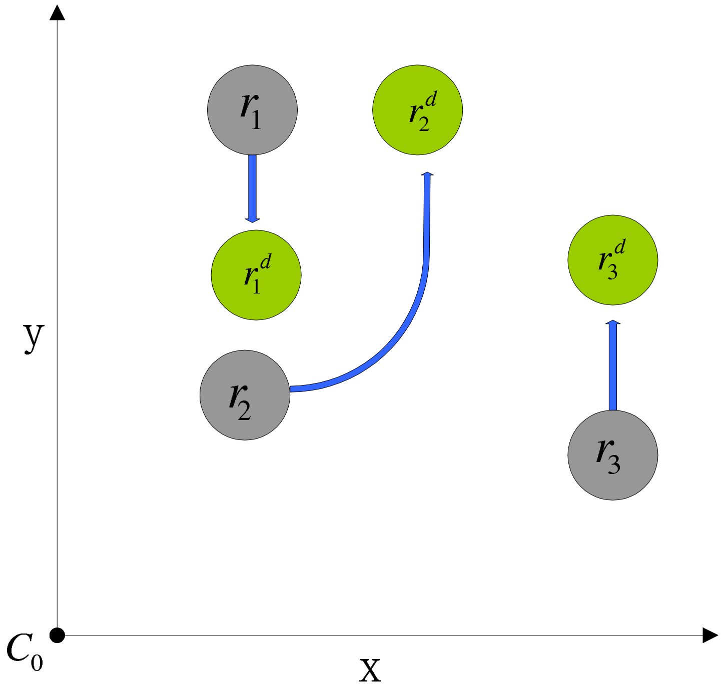


Figure 6.4. Framework for P3 mobile robot team with virtual leaders

ith vehicle's actual position at time t and $r_i^d(t) = [x_i^d(t), y_i^d(t)]^T$ is the ith $P3$ mobile robotic vehicle's desired position at time t. Both its actual and desired position at time t are relative to C_0. As shown in the figure, the group can move with the desired formation shape only if each vehicle can track its desired position accurately.

In this work suppose each $P3$ mobile robot knows the state of its virtual subgroup leader versus time. If each vehicle has inconsistent knowledge of its virtual subgroup leader's states, then the desired formation cannot be maintained.

The linearized model of each robot in the group can be represented by the following equation:

$$\dot{\mathbf{r}}_i(t) = \mathbf{u}_i(t), i = 1, ..., n. \tag{6.7}$$

where $\mathbf{r}_i(t) = [x_i(t), y_i(t)]^T \in \mathbb{R}^m$ is the state of the ith robot in the group which includes its position and velocity information. i is the index denoting the number of the robot in the group. n is the total number of all robots in the group. $\mathbf{u}_i(t) \in \mathbb{R}^m$ is the control input signal.

Now define the virtual subgroup leader's state, which is also the desired state for each $P3$ mobile robot in the group, as $\mathbf{r}_i^d(t) = [x_i^d(t), y_i^d(t)]^T$. If $\mathbf{r}_i(t) \rightarrow \mathbf{r}_i^d(t)$, $i = 1, \ldots, n$, as $t \rightarrow \infty$, then the desired formation shape is maintained and the group movement follows the desired reference.

In the next section, a distributed formation control approach which accommodates an arbitrary number of subgroup leaders and ensures accurate formation maintenance by sharing information between group members will be introduced.

6.4 A Novel Consensus Control Approach

Apply a distributed consensus tracking control algorithm on the control level as

$$\mathbf{u}_i(t) = \dot{\mathbf{r}}_i^d(t) - \mathbf{k}_i(\mathbf{r}_i(t) - \mathbf{r}_i^d(t)) - \sum_{j=1, i\neq j}^{n} \mathbf{a}_{ij}(\mathbf{r}_j(t-\tau) - \mathbf{r}_j^d(t-\tau)), \tag{6.8}$$

where $\mathbf{k}_i$ is the control gain need to be designed, notice that each robotic vehicle in the group has the same group communication coupling and kinematic model, $\mathbf{k}_i$ can be represented as $\mathbf{k}$ instead; $\mathbf{a}_{ij}$ is the (i, j) entry of adjacency matrix $A_n^v \in \mathbb{R}^{n\times n}$ according to the interaction topology $Ç_n^v = (\nu_n^v, \varepsilon_n^v)$ for $\mathbf{r}_i - \mathbf{r}_i^d$, $\mathbf{r}_j - \mathbf{r}_j^d$ is the information from neighbors of the *ith* $P3$ mobile robot, this term also can be treated as the coupling between the *ith* $P3$ mobile robot and its neighbors, τ is the network induced group communication delay. Submit (6.8) into (6.7) then

$$\dot{\mathbf{r}}_i(t) = \dot{\mathbf{r}}_i^d(t) - \mathbf{k}(\mathbf{r}_i(t) - \mathbf{r}_i^d(t)) - \sum_{j=1, i\neq j}^{n} \mathbf{a}_{ij}(\mathbf{r}_j(t-\tau) - \mathbf{r}_j^d(t-\tau)), \tag{6.9}$$

if the error vector has been defined as

$$\mathbf{e}_i(t) = \mathbf{r}_i(t) - \mathbf{r}_i^d(t), \tag{6.10}$$

then (6.9) can be rewritten as

$$\dot{\mathbf{e}}_i(t) = -\mathbf{k}\mathbf{e}_i(t) - \sum_{j=1, i\neq j}^{n} \mathbf{a}_{ij}\mathbf{e}_j(t-\tau). \tag{6.11}$$

It is obvious that if $\mathbf{e}_i \rightarrow 0, i = 1, \ldots, n$ as $t \rightarrow \infty$ which means $\mathbf{r}_i \rightarrow \mathbf{r}_i^d, i = 1, \ldots, n$ as $t \rightarrow \infty$, then the desired formation shape is maintained and the group movement follows the desired reference. Next is to find the sufficient conditions for the design of the control gain $\mathbf{k}$ which can stabilize the error dynamics represented by (6.11).

Lemma 6.1. - *Jensen Inequality For any constant matrix $E \in \mathcal{R}^{n\times n}$, $E = E^T > 0$, vector function $\boldsymbol{\omega} : [0, \tau] \rightarrow \mathcal{R}^n$ such that the integrations concerned are well defined, then,*

$$\tau \int_0^{\tau} \boldsymbol{\omega}^T(s)E\boldsymbol{\omega}(s)ds \geq \left[\int_0^{\tau} \boldsymbol{\omega}(s)ds\right]^T E \left[\int_0^{\tau} \boldsymbol{\omega}(s)ds\right]. \tag{6.12}$$

Theorem 6.1. *Consider the error dynamics model represented by* (6.11)*, for a given time delay τ and the number of members in one group n, if there exist symmetric positive definite matrices $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} > 0$, $Q = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{22} \end{bmatrix} > 0$, $R = \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix} > 0$, matrices $M_i, N_i, i = 1, \ldots, 5$ with appropriate dimensions, such* *that the following inequality holds*

$$H = \begin{bmatrix} H_{11} & * & * & * & * \\ H_{21} & H_{22} & * & * & * \\ H_{31} & H_{32} & H_{33} & * & * \\ H_{41} & H_{42} & H_{43} & H_{44} & * \\ H_{51} & H_{52} & H_{53} & H_{54} & H_{55} \end{bmatrix} < 0, \tag{6.13}$$

where

$$\begin{aligned}
H_{11} &= \hat{P}_{12} + \hat{P}_{12}^T + (n-1)\hat{Q}_{11} + (n-1)\tau\hat{R}_{11} \\
&\quad + N_1 + N_1^T + M_1K + K^TM_1^T \\
H_{21} &= -\hat{P}_{12}^T + N_2 - N_1^T + A^TM_1^T + M_2K \\
H_{22} &= -(n-1)\hat{Q}_{11} - N_2 - N_2^T + M_2A + A^TM_2^T \\
H_{31} &= (n-1)\hat{P}_{11} + N_3 + M_1^T + M_3K \\
H_{32} &= -N_3 + M_3A + M_2^T \\
H_{33} &= (n-1)\hat{Q}_{22} + (n-1)\tau\hat{R}_{22} + M_3 + M_3^T \\
H_{41} &= (n-1)\hat{P}_{22} + N_4 + M_4K \\
H_{42} &= -(n-1)\hat{P}_{22} - N_4 + M_4A \\
H_{43} &= \hat{P}_{12}^T + M_4 \quad H_{52} = -N_2^T - N_5 + M_5A \\
H_{51} &= -N_1^T + N_5 + M_5K \quad H_{44} = -\frac{(n-1)\hat{R}_{11}}{\tau} \\
H_{53} &= -N_3^T + M_5 \quad H_{54} = -N_4^T \\
H_{55} &= -\frac{(n-1)\hat{R}_{22}}{\tau} - N_5 - N_5^T,
\end{aligned} \tag{6.14}$$

then system (6.11) *is asymptotically stable, e.g.* $\mathbf{e}_i(t)$ *tends to zero asymptotically which means the ith robot in the group tracks its desired trajectory well.*

Proof. For each robot in the group take the Lyapunov krasovskii functional candidate as:

$$\begin{aligned} V_i &= (n-1)\mathbf{e}_i^T(t)P_{11}\mathbf{e}_i(t) + 2\mathbf{e}_i^T(t)P_{12}\sum_{j=1,j\neq i}^{n}\int_{t-\tau}^{t}\mathbf{e}_j(s)ds \\ &+ \sum_{j=1,j\neq i}^{n}\int_{t-\tau}^{t}\mathbf{e}_j^T(s)dsP_{22}\int_{t-\tau}^{t}\mathbf{e}_j(s)ds \\ &+ \sum_{j=1,j\neq i}^{n}\int_{t-\tau}^{t}[\mathbf{e}_j^T(s),\dot{\mathbf{e}}_j^T(s)]Q\begin{bmatrix}\mathbf{e}_j(s)\\ \dot{\mathbf{e}}_j(s)\end{bmatrix}ds \\ &+ \sum_{j=1,j\neq i}^{n}\int_{-\tau}^{0}\int_{t+\theta}^{t}[\mathbf{e}_j^T(s),\dot{\mathbf{e}}_j^T(s)]R\begin{bmatrix}\mathbf{e}_j(s)\\ \dot{\mathbf{e}}_j(s)\end{bmatrix}dsd\theta, \end{aligned} \tag{6.15}$$

now it is defined that

$$\hat{\mathbf{e}}^T(t) = [\mathbf{e}_1^T(t),\ldots,\mathbf{e}_n^T(t)]_{1\times n}, \tag{6.16}$$

and take the multiple Lyapunov krasovskii functional candidate as:

$$\begin{aligned} V &= \sum_{i=1}^{n}V_i = (n-1)\hat{\mathbf{e}}^T(t)\hat{P}_{11}\hat{\mathbf{e}}(t) + 2\hat{\mathbf{e}}^T(t)\hat{P}_{12}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds \\ &+(n-1)\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\hat{P}_{22}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds \\ &+(n-1)\int_{t-\tau}^{t}[\hat{\mathbf{e}}^T(s),\dot{\hat{\mathbf{e}}}^T(s)]\begin{bmatrix}\hat{Q}_{11} & 0\\ 0 & \hat{Q}_{22}\end{bmatrix}_{2n\times 2n}\begin{bmatrix}\hat{\mathbf{e}}(s)\\ \dot{\hat{\mathbf{e}}}(s)\end{bmatrix}ds \\ &+(n-1)\int_{-\tau}^{0}\int_{t+\theta}^{t}[\hat{\mathbf{e}}^T(s),\dot{\hat{\mathbf{e}}}^T(s)]\begin{bmatrix}\hat{R}_{11} & 0\\ 0 & \hat{R}_{22}\end{bmatrix}_{2n\times 2n} \\ &\begin{bmatrix}\hat{\mathbf{e}}(s)\\ \dot{\hat{\mathbf{e}}}(s)\end{bmatrix}dsd\theta, \end{aligned} \tag{6.17}$$

where

$$\begin{aligned}
\hat{P}_{11} &= \begin{bmatrix} P_{11} & & \\ & \ddots & \\ & & P_{11} \end{bmatrix}_{n\times n}, \hat{P}_{22} = \begin{bmatrix} P_{22} & & \\ & \ddots & \\ & & P_{22} \end{bmatrix}_{n\times n} \\
\hat{P}_{12} &= \begin{bmatrix} 0 & P_{12} & \cdots & P_{12} \\ P_{12} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & P_{12} \\ P_{12} & \cdots & P_{12} & 0 \end{bmatrix}_{n\times n}
\end{aligned}$$

$$\begin{aligned}
\hat{Q}_{11} &= \begin{bmatrix} Q_{11} & & \\ & \ddots & \\ & & Q_{11} \end{bmatrix}_{n\times n}, \hat{Q}_{22} = \begin{bmatrix} Q_{22} & & \\ & \ddots & \\ & & Q_{22} \end{bmatrix}_{n\times n} \\
\hat{R}_{11} &= \begin{bmatrix} R_{11} & & \\ & \ddots & \\ & & R_{11} \end{bmatrix}_{n\times n}, \hat{R}_{22} = \begin{bmatrix} R_{22} & & \\ & \ddots & \\ & & R_{22} \end{bmatrix}_{n\times n}, \qquad (6.18)
\end{aligned}$$

With appropriate dimensions, the following two zero equations hold:

$$\begin{aligned}
\phi_1 &= 2\mathbf{z}^T N \left[\hat{\mathbf{e}}(t) - \int_{t-\tau}^{t} \dot{\hat{\mathbf{e}}}(s)ds - \hat{\mathbf{e}}(t-\tau)\right] = 0 \\
\phi_2 &= 2\mathbf{z}^T M \left[\dot{\hat{\mathbf{e}}}(t) + K\hat{\mathbf{e}}(t) + A\hat{\mathbf{e}}(t-\tau)\right] = 0, \qquad (6.19)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{z} &= \left[\hat{\mathbf{e}}(t), \hat{\mathbf{e}}(t-\tau), \dot{\hat{\mathbf{e}}}(t), \int_{t-\tau}^{t} \hat{\mathbf{e}}(s)ds, \int_{t-\tau}^{t} \dot{\hat{\mathbf{e}}}(s)ds\right]^T \\
N &= [N_1, N_2, N_3, N_4, N_5]^T, N_{i=1,\ldots,5} = diag[n_i, \ldots, n_i]_{n\times n} \\
M &= [M_1, M_2, M_3, M_4, M_5]^T, M_{i=1,\ldots,5} = diag[m_i, \ldots, m_i]_{n\times n}
\end{aligned}$$

$$\mathbf{K} = diag[\mathbf{k}, \ldots, \mathbf{k}]_{n\times n}. \tag{6.20}$$

Then the derivative of the multiple Lyapunov function candidate is as follows:

$$\begin{aligned}
\dot{V} &= (n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{P}_{11}\hat{\mathbf{e}}(t) + (n-1)\hat{\mathbf{e}}^T(t)\hat{P}_{11}\dot{\hat{\mathbf{e}}}(t) \\
&\quad +2\dot{\hat{\mathbf{e}}}^T(t)\hat{P}_{12}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds + 2\hat{\mathbf{e}}^T(t)\hat{P}_{12}\hat{\mathbf{e}}(t) \\
&\quad -2\hat{\mathbf{e}}^T(t)\hat{P}_{12}\hat{\mathbf{e}}(t-\tau) + (n-1)\hat{\mathbf{e}}^T(t)\hat{P}_{22}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds \\
&\quad -(n-1)\hat{\mathbf{e}}^T(t-\tau)\hat{P}_{22}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds \\
&\quad +(n-1)\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\hat{P}_{22}\hat{\mathbf{e}}(t) \\
&\quad -(n-1)\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\hat{P}_{22}\hat{\mathbf{e}}(t-\tau) \\
&\quad +(n-1)\hat{\mathbf{e}}^T(t)\hat{Q}_{11}\hat{\mathbf{e}}(t) + (n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{Q}_{22}\dot{\hat{\mathbf{e}}}(t) \\
&\quad -(n-1)\hat{\mathbf{e}}^T(t-\tau)\hat{Q}_{11}\hat{\mathbf{e}}(t-\tau) - (n-1)\dot{\hat{\mathbf{e}}}^T(t-\tau)\hat{Q}_{22}\dot{\hat{\mathbf{e}}}(t-\tau) \\
&\quad +\tau(n-1)\hat{\mathbf{e}}^T(t)\hat{R}_{11}\hat{\mathbf{e}}(t) + \tau(n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{R}_{22}\dot{\hat{\mathbf{e}}}(t) \\
&\quad -(n-1)\int_{t-\tau}^{t}[\hat{\mathbf{e}}^T(s), \dot{\hat{\mathbf{e}}}^T(s)]\begin{bmatrix}\hat{R}_{11} & 0 \\ 0 & \hat{R}_{22}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{e}}(s) \\ \dot{\hat{\mathbf{e}}}(s)\end{bmatrix}ds,
\end{aligned} \tag{6.21}$$

then

$$\begin{aligned}
\dot{V} &= \dot{V} + \phi_1 + \phi_2 \\
&= (n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{P}_{11}\hat{\mathbf{e}}(t) + (n-1)\hat{\mathbf{e}}^T(t)\hat{P}_{11}\dot{\hat{\mathbf{e}}}(t) \\
&\quad +2\dot{\hat{\mathbf{e}}}^T(t)\hat{P}_{12}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds + 2\hat{\mathbf{e}}^T(t)\hat{P}_{12}\hat{\mathbf{e}}(t) \\
&\quad -2\hat{\mathbf{e}}^T(t)\hat{P}_{12}\hat{\mathbf{e}}(t-\tau) + (n-1)\hat{\mathbf{e}}^T(t)\hat{P}_{22}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds \\
&\quad -(n-1)\hat{\mathbf{e}}^T(t-\tau)\hat{P}_{22}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds \\
&\quad +(n-1)\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\hat{P}_{22}\hat{\mathbf{e}}(t)
\end{aligned}$$

$$\begin{aligned}
&-(n-1)\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\hat{P}_{22}\hat{\mathbf{e}}(t-\tau)\\
&+(n-1)\hat{\mathbf{e}}^T(t)\hat{Q}_{11}\hat{\mathbf{e}}(t)+(n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{Q}_{22}\dot{\hat{\mathbf{e}}}(t)\\
&-(n-1)\hat{\mathbf{e}}^T(t-\tau)\hat{Q}_{11}\hat{\mathbf{e}}(t-\tau)-(n-1)\dot{\hat{\mathbf{e}}}^T(t-\tau)\hat{Q}_{22}\dot{\hat{\mathbf{e}}}(t-\tau)\\
&+\tau(n-1)\hat{\mathbf{e}}^T(t)\hat{R}_{11}\hat{\mathbf{e}}(t)+\tau(n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{R}_{22}\dot{\hat{\mathbf{e}}}(t)\\
&-(n-1)\int_{t-\tau}^{t}[\hat{\mathbf{e}}^T(s),\dot{\hat{\mathbf{e}}}^T(s)]\begin{bmatrix}\hat{R}_{11} & 0\\ 0 & \hat{R}_{22}\end{bmatrix}_{2n\times 2n}\begin{bmatrix}\hat{\mathbf{e}}(s)\\ \dot{\hat{\mathbf{e}}}(s)\end{bmatrix}ds\\
&+2\mathbf{z}^TN\left[\hat{\mathbf{e}}(t)-\int_{t-\tau}^{t}\dot{\hat{\mathbf{e}}}(s)ds-\hat{\mathbf{e}}(t-\tau)\right]\\
&+2\mathbf{z}^TM\left[\dot{\hat{\mathbf{e}}}(t)+K\hat{\mathbf{e}}(t)+A\hat{\mathbf{e}}(t-\tau)\right].
\end{aligned}\tag{6.22}$$

Using Lemma 6.1, it can be obtained that

$$\begin{aligned}
\dot{V}\ \leqslant\ & (n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{P}_{11}\hat{\mathbf{e}}(t)+(n-1)\hat{\mathbf{e}}^T(t)\hat{P}_{11}\dot{\hat{\mathbf{e}}}(t)\\
&+2\dot{\hat{\mathbf{e}}}^T(t)\hat{P}_{12}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds+2\hat{\mathbf{e}}^T(t)\hat{P}_{12}\hat{\mathbf{e}}(t)\\
&-2\hat{\mathbf{e}}^T(t)\hat{P}_{12}\hat{\mathbf{e}}(t-\tau)+(n-1)\hat{\mathbf{e}}^T(t)\hat{P}_{22}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds\\
&-(n-1)\hat{\mathbf{e}}^T(t-\tau)\hat{P}_{22}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds+(n-1)\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\hat{P}_{22}\hat{\mathbf{e}}(t)\\
&-(n-1)\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\hat{P}_{22}\hat{\mathbf{e}}(t-\tau)\\
&+(n-1)\hat{\mathbf{e}}^T(t)\hat{Q}_{11}\hat{\mathbf{e}}(t)+(n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{Q}_{22}\dot{\hat{\mathbf{e}}}(t)\\
&-(n-1)\hat{\mathbf{e}}^T(t-\tau)\hat{Q}_{11}\hat{\mathbf{e}}(t-\tau)\\
&+\tau(n-1)\hat{\mathbf{e}}^T(t)\hat{R}_{11}\hat{\mathbf{e}}(t)+\tau(n-1)\dot{\hat{\mathbf{e}}}^T(t)\hat{R}_{22}\dot{\hat{\mathbf{e}}}(t)\\
&-\frac{n-1}{\tau}\int_{t-\tau}^{t}\hat{\mathbf{e}}^T(s)ds\widehat{R_{11}}\int_{t-\tau}^{t}\hat{\mathbf{e}}(s)ds-\frac{n-1}{\tau}\int_{t-\tau}^{t}\dot{\hat{\mathbf{e}}}^T(s)ds\widehat{R_{22}}\\
&\int_{t-\tau}^{t}\dot{\hat{\mathbf{e}}}(s)ds+2\mathbf{z}^TN\left[\hat{\mathbf{e}}(t)-\int_{t-\tau}^{t}\dot{\hat{\mathbf{e}}}(s)ds-\hat{\mathbf{e}}(t-\tau)\right]\\
&+2\mathbf{z}^TM\left[\dot{\hat{\mathbf{e}}}(t)+K\hat{\mathbf{e}}(t)+A\hat{\mathbf{e}}(t-\tau)\right]\\
=\ & \mathbf{z}^TH\mathbf{z}.
\end{aligned}\tag{6.23}$$

where H was shown in (6.13). If (6.13) holds, then from (6.23), it shows that $\dot{V} \leq 0$ which means the system (6.11) is asymptotically stable, e.g. $\mathbf{e}_i(t)$ tends to zero asymptotically which means the *ith* robot in the group tracks its desired trajectory well. □

Note that the LMI condition in (6.13) is non-convex and hence the following theorem is proposed to be the sufficient condition of (6.13).

Theorem 6.2. *Consider the error dynamics model represented by* (6.11)*, for a given time delay* τ*, given scalars* θ_i*,* $i = 1, \ldots, 5$ *and the number of members in one group* n*, if there exist matrices* $\bar{P}_{11}$*,* $\bar{P}_{12}$*,* $\bar{P}_{22}$*,* $\bar{Q}_{11}$*,* $\bar{Q}_{22}$*,* $\bar{R}_{11}$*,* $\bar{R}_{22}$*,* $\bar{N}_i, i = 1, \ldots, 5$ *with appropriate dimensions, such that the following inequality holds*

$$\begin{bmatrix} \bar{H}_{11} & * & * & * & * \\ \bar{H}_{21} & \bar{H}_{22} & * & * & * \\ \bar{H}_{31} & \bar{H}_{32} & \bar{H}_{33} & * & * \\ \bar{H}_{41} & \bar{H}_{42} & \bar{H}_{43} & \bar{H}_{44} & * \\ \bar{H}_{51} & \bar{H}_{52} & \bar{H}_{53} & \bar{H}_{54} & \bar{H}_{55} \end{bmatrix} < 0, \tag{6.24}$$

where

$$\begin{aligned}
\bar{H}_{11} &= \bar{P}_{12} + \bar{P}_{12}^T + (n-1)\bar{Q}_{11} + (n-1)\tau\bar{R}_{11} \\
&\quad + \bar{N}_1 + \bar{N}_1^T + \theta_1 Y + \theta_1 Y^T \\
\bar{H}_{21} &= -\bar{P}_{12}^T + \bar{N}_2 - \bar{N}_1^T + \theta_2 X A^T + \theta_2 Y \\
\bar{H}_{22} &= -(n-1)\bar{Q}_{11} - \bar{N}_2 - \bar{N}_2^T + \theta_2 A X^T + \theta_2 X A^T \\
\bar{H}_{31} &= (n-1)\bar{P}_{11} + \bar{N}_3 + \theta_1^T X + \theta_3 Y \\
\bar{H}_{32} &= -\bar{N}_3 + \theta_3 A X^T + \theta_2 X \\
\bar{H}_{33} &= (n-1)\bar{Q}_{22} + (n-1)\tau\bar{R}_{22} + \theta_3 X^T + theta_3 X \\
\bar{H}_{41} &= (n-1)\bar{P}_{22} + \bar{N}_4 + \theta_4 Y
\end{aligned}$$

$$
\begin{aligned}
\bar{H}_{42} &= -(n-1)\bar{P}_{22} - \bar{N}_4 + \theta_4 A X^T \\
\bar{H}_{43} &= \bar{P}_{12}^T + \theta_4 X^T \quad \bar{H}_{52} = -\bar{N}_2^T - \bar{N}_5 + \theta_5 A X^T \\
\bar{H}_{51} &= -\bar{N}_1^T + \bar{N}_5 + \theta_5 Y \quad \bar{H}_{44} = -\frac{(n-1)\bar{R}_{11}}{\tau} \\
\bar{H}_{53} &= -\bar{N}_3^T + \theta_5 X^T \quad \bar{H}_{54} = -\bar{N}_4^T \\
\bar{H}_{55} &= -\frac{(n-1)\bar{R}_{22}}{\tau} - \bar{N}_5 - \bar{N}_5^T, \qquad (6.25)
\end{aligned}
$$

then system (6.11) *is asymptotically stable with control gain* $K = YX^{-1}$*, e.g.* $\mathbf{e}_i(t)$ *tends to zero asymptotically which means the ith robot in the group tracks its desired trajectory well.*

Proof. In order to transform the nonconvex LMI in (6.13) into a solvable LMI, assume that $M_i = \theta_i M_0, i = 1, \ldots, 5$ where θ_i is known and given. Define $X = M_0^{-1}$, $\hat{W} = diag(X, X, X, X, X)$ and $Y = KX$. Then by pre-multiplying the inequality in (6.13) by $\hat{W}^T$ and post-multiplying by $\hat{W}$, the inequality (6.24) can be obtained. □

Remark 6.1. *Eq.*(6.24) *in Theorem.*(6.2) *and Eq.*(6.13) *in Theorem.*(6.1) *are used to design control gain* K*. Eq.*(6.24) *is a sufficient condition of Eq.*(6.13) *in Theorem.*(6.1) *due to the simplification of* M_i *matrices.*

6.5 Summary

In this chapter, a consensus based tracking control strategy has been proposed for a kind of virtual leader formation control approach of multiple mobile robot group with group communication delay. The kinematic model of $P3$ mobile robot has been introduced. The formation control algorithm has been applied to the linearized dynamic model of $P3$ mobile robot. A novel multiple Lyapunov functional candidate has been proposed to give the sufficient conditions of the control gain design. Theorems have been provided to list the sufficient conditions as well.

Chapter 7

Simulation Results

7.1 Numerical Example of Stochastic Stabilization of Sampled-data NCSs

7.1.1 Numerical Example *I*

Real network conditions were applied to the simulation. The delays and packet loss that were recorded during the network experiment were replayed for the simulation, reproducing the exact performance that a real network control system would have experienced.

Consider the following nominal continuous-time system which is controlled through networks with packet losses and time varying delays recorded in the experiment:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -3 & -0.001 \\ -1 & 0.001 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}(t - d_2(t)), \tag{7.1}$$

with $\mathbf{x}(0) = [0.5, -0.5]^T$. The continuous system is open loop unstable with eigenvalues of A as -3.0013 and 0.0013. The plant is sampled with a sampling period $T_s = 0.01$ *seconds*. The packet-loss upper bound is $S = 5$. The transition probability matrix (8.4) was used to represent the packet losses in this example. The time varying delay is set to be $m_J T_s,\ m_J \in \{1, 2, 3, 4, 5\}$. The sampled-data controller is designed

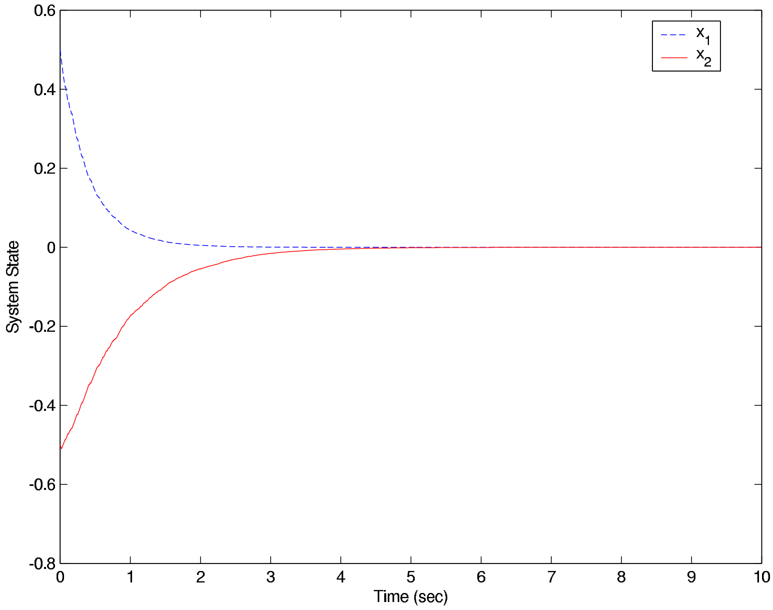


Figure 7.1. The state response with bounded delays and packet losses

as in (4.39), applying Theorem 4.3 with $\beta = 3$, a networked controller gain is designed as:

$$K = YX^{-1} = \begin{bmatrix} 1.0829 & -3.5142 \end{bmatrix} \begin{bmatrix} -0.0068 & -0.0027 \\ -0.6352 & 2.0597 \end{bmatrix}^{-1} = \begin{bmatrix} 0.1078 & -1.7060 \end{bmatrix}.$$

Fig.7.1 shows the state response of the system. Since the control gain was well designed, the system can be stochastically stabilized in around 5 seconds.

7.1.2 Numerical Example *II*

In this section, the numerical example of the nominal continuous system which has been used in [21] and [51] was considered and simulated to illustrate the effectiveness of the proposed approach. The system is as follows:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -1 & 0 & -0.5 \\ 1 & -0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{u}(t). \tag{7.2}$$

The maximum continuing packet loss amounts was assumed to be five, which means that up to 80% of the packets could be lost during the network transmissions. Furthermore, the packet-loss process is governed by a Markov Chain. The transition probability matrix in [11] is applied here:

$$\prod = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.3 & 0 & 0 \\ 0 & 0.2 & 0.5 & 0.3 & 0 \\ 0 & 0 & 0.2 & 0.5 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.2 & 0.5 \end{bmatrix}. \tag{7.3}$$

Applying Theorem. 4.1, the control gain is obtained as $K = [0.0412, -0.0117, -0.3172]$. The initial state $\mathbf{x}_0$ has been chosen as $\mathbf{x}_0 = [5 \;\; 0 \; -5]^T$. The sampling time $T_s = 0.5s$. Fig.7.2 depicts the trajectory of the system state when the packet-loss process is a Markovian process. The successfully arrived control inputs are marked with circles on the time axis. As shown in the picture that only 17 control inputs arrived at the system during the first 20 seconds, 57.5% of the packets were lost. The system can

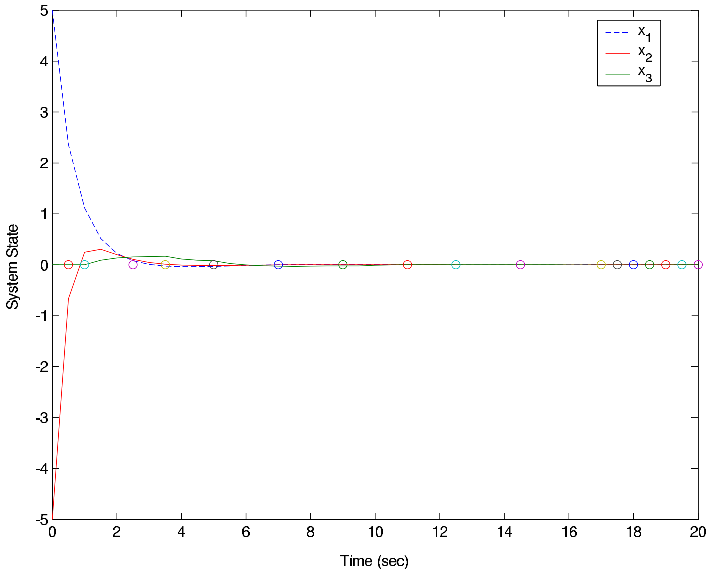


Figure 7.2. State response (Markovian packet loss)

be stabilized by the controller in 8 seconds which is shorter than the one in [21].

If the time varying delays used in Section. 8.1 are considered in the system (7.2), the new control gain can be obtained as $K = [0.0379, -0.0637, -0.8173]$ by applying Theorem. 4.3. Fig.7.3 shows the state response of the system. From the figure, 72.5% of the packets were lost. The closed-loop system with both Markovian packet losses and time varying delays can be stabilized in 15 seconds by the feedback controller.

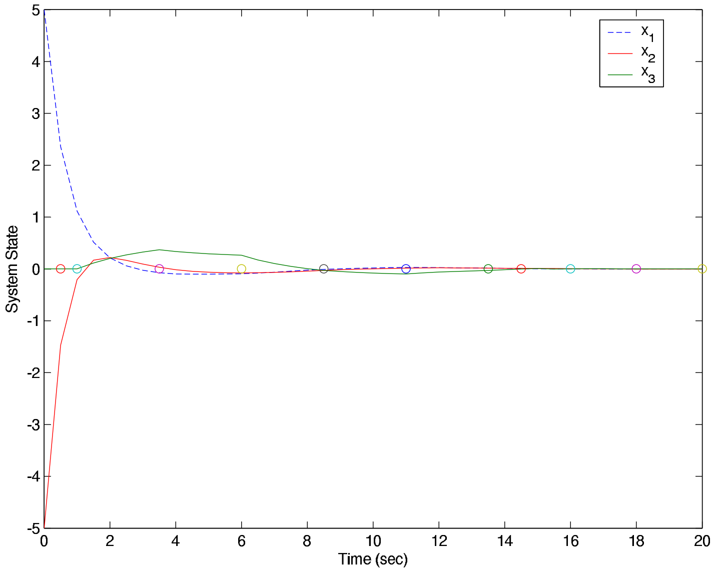


Figure 7.3. State response (Time varying delays)

7.2 Numerical Example of Distributed Consensus Formation Control of Networked Multi-agent Robotic Systems with Time Delays

In this section, the results of the simulation work which had been done to support the theoretic results in Chapter 5 have been shown. The simulation work include two parts.

In the first part, three-robot group movement has been simulated. The task is asking the three-robot group to track the trajectory of the virtual leader. In other words, the desired trajectory of the three-robot group has been existent already at the beginning of the group movement. Two different types of group coupling configuration,called closed chain and open chain respectively, have been considered.

In the second part, three-robot group movement with disturbance has been studied and simulated. Two different cases were considered in the simulation work. For the first case there is group communication happening between each of the robots in the group during the group movement. For the second case there is not any group communication happening during the group movement. For both cases no matter there is or no group communication, the disturbance has been input to one of the three robots in the group at a fixed time point during the whole group movement. The performance of the group movement for both cases has been studied to investigate the meaning of the group communication of multi-agent systems.

7.2.1 Group Performance Comparisons: Closed and Open Chain Configuration

In this subsection, the three-robot group is asked to track the trajectory of the virtual leaders and ensure accurate formation maintenance through information exchange between neighbors. The two group coupling configurations have been shown by Fig.7.4.

The group in Fig.7.4(a) has a configuration called “closed chain group”. The group in Fig.7.4(b) has a configuration called “open chain group”. The distributed control scheme is independent of the configurations and it is applicable in groups which have other configurations. However, only closed chain and open chain configurations are considered in the simulation since the groups can be easily tangled in

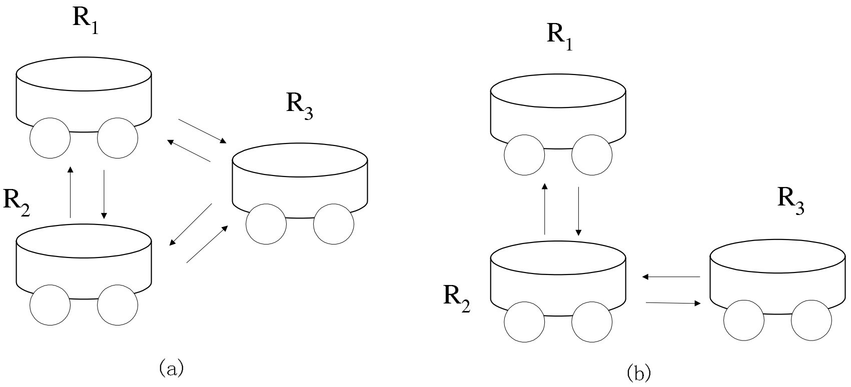


Figure 7.4. A three-robot group configuration: (*a*) closed chain; (*b*) open chain

other configurations.

A. Closed Chain Group

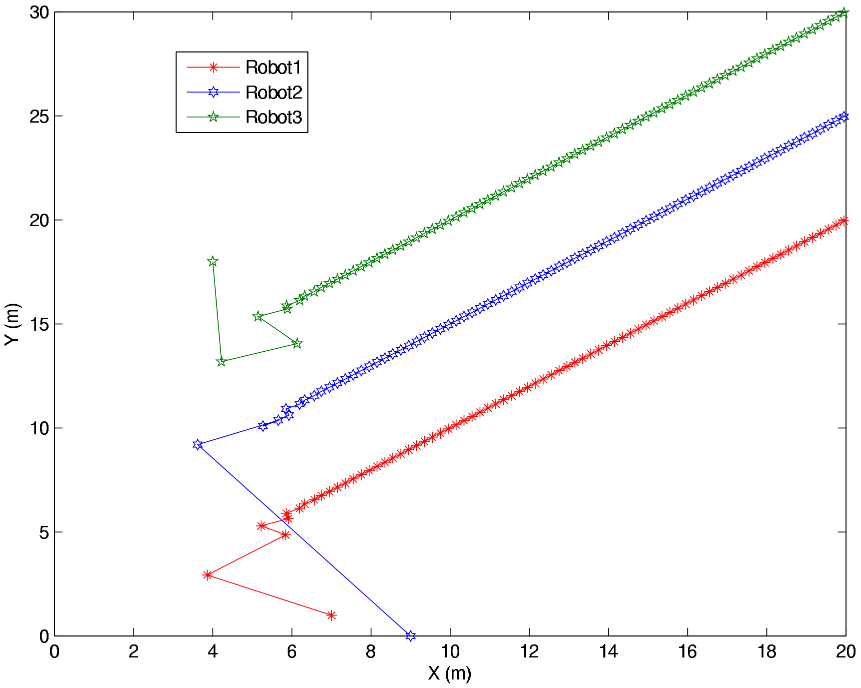


Figure 7.5. Evolution of closed chain group movement

The case of a closed chain group as shown in Fig.7.4 (*a*) has been simulated. The robot group consists of three mobile robots free to move on a plane. The dynamics for each robot was given in Chapter 5. The element values in the matrices are given as:

$$\begin{aligned} A_{ii} &= \begin{bmatrix} 0.2 & 0 \\ 0.1 & 0.2 \end{bmatrix} \\ A_{ij} &= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \\ B &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \end{aligned} \tag{7.4}$$

where $i, j \in [1, 2, 3], i \neq j$. The distributed control scheme in Chapter 5 was applied in the simulation. The control gain was designed as:

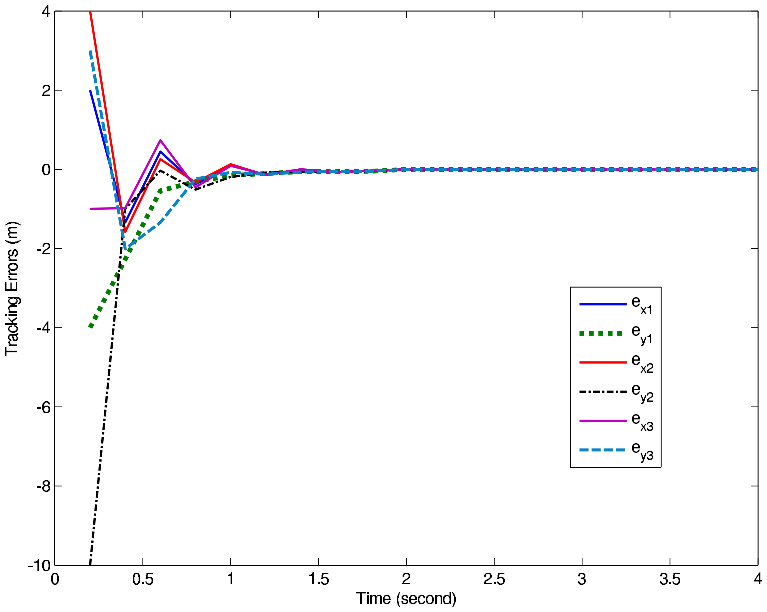


Figure 7.6. Tracking error of closed chain group movement

$$\begin{aligned} K_{ii} &= \begin{bmatrix} -4.7293 & 0 \\ 0 & -4.7293 \end{bmatrix} \\ K_{ij} &= \begin{bmatrix} -0.6524 & 0 \\ 0 & -0.6524 \end{bmatrix}, \end{aligned} \tag{7.5}$$

to satisfy the conditions in Theorem 2 with $L = 0.1$, $\tau = 0.2sec$. The three robots start moving at different initial points which are located in a plane as $[x_0, y_0]_{R_1} = [7, 1]$;

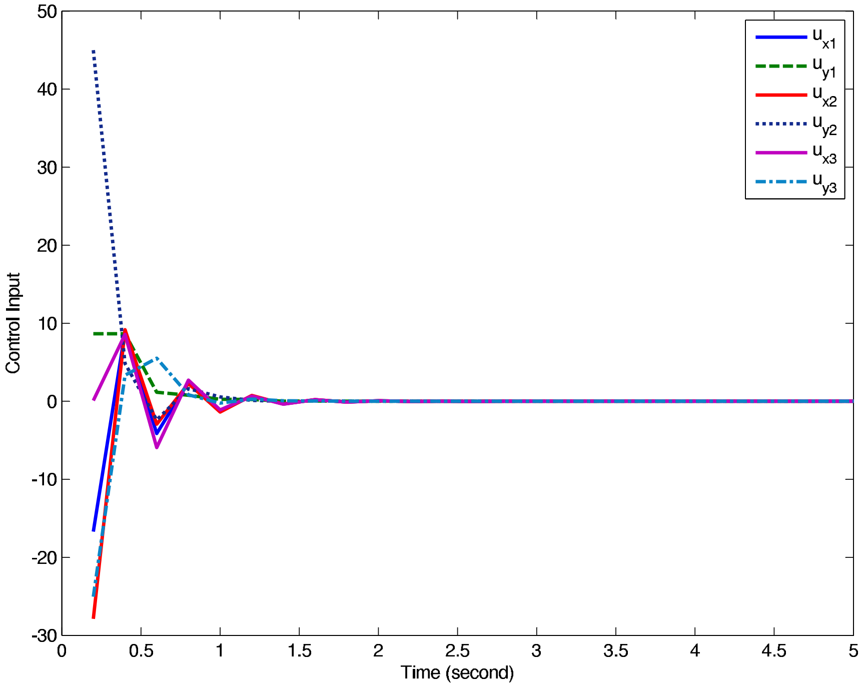


Figure 7.7. Control input signals of closed chain robot group

$[x_0, y_0]_{R_2} = [9, 0]$; $[x_0, y_0]_{R_3} = [4, 18]$. The desired trajectories for each robot are $R_1 : y_d = x_d$ with initial point at $x_d(0) = 5, y_d(0) = 5$; $R_2 : y_d = x_d + 5$ with initial point at $x_d(0) = 5, y_d(0) = 10$; $R_3 : y_d = x_d + 10$ with initial point at $x_d(0) = 5, y_d(0) = 15$. Fig.7.5 gives the evolution of the closed chain group movement. From the figure, at the beginning the three robots start moving at three different initial positions and try to get close to their own desired trajectories without any collision. In the first 1.2 seconds, the three robots move in different velocities based on both of their desired trajectories and initial positions. After two seconds the group track the desired trajectory very well and keep the accurate formation maintenance.

The tracking error between closed chain group movement trajectory and its own desired trajectory has been shown by Fig.7.6. The control input signals sent by the controller on each of the three robots has been recorded in Fig.7.7. From the figures,

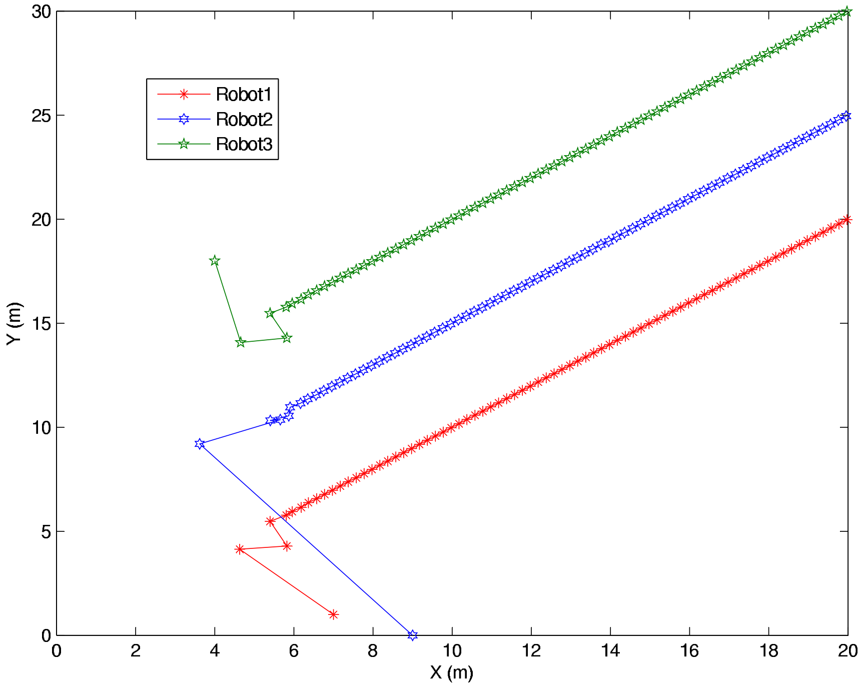


Figure 7.8. Evolution of open chain group movement

at the beginning of the movement, the controller on each of the robots sends control input signals with different values to the robot based on how big the tracking error between the robot and the desired position is. Since the controller has been designed well, in the two second transitional time, the controller adjusts the robot to its desired position without any collision. After two seconds, the tracking errors between group trajectory and desired trajectory converge to zero which means the effect caused by the communication delay between each robot has been reduced by the controller and do not affect group movement seriously. Control signals converge to zero as the tracking errors converge to zero.

B. Open Chain Group

Second we simulated the case of a open chain group as shown in Fig.7.4 (b). The same situation as in closed chain case, the robot group consists of three mobile robots

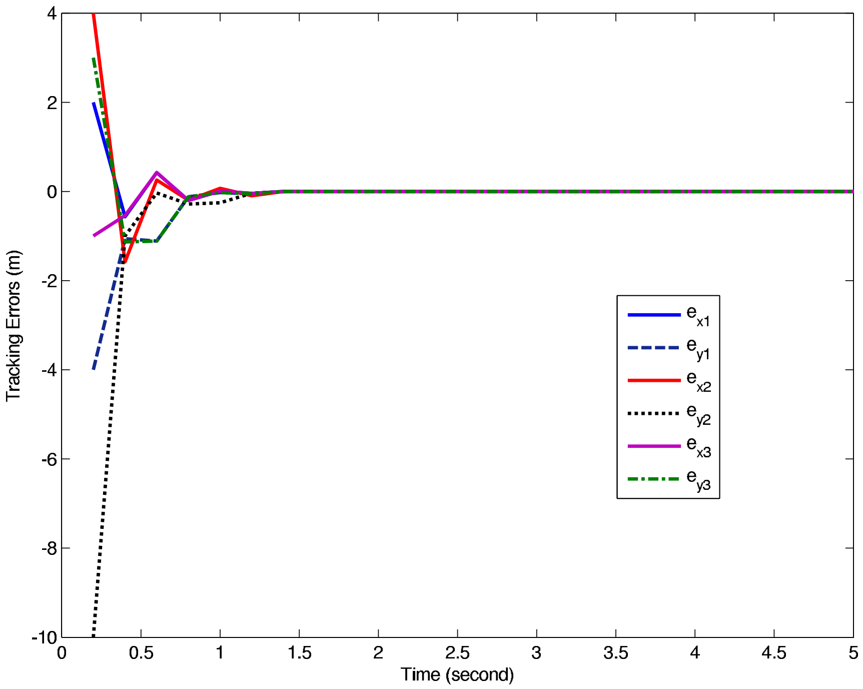


Figure 7.9. Tracking error of open chain group movement

free to move on a plane. The dynamics for each robot and most element values in the matrices are the same as in closed chain case except

$$
\begin{aligned}
A_{13} = A_{31} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
K_{11} = K_{33} &\begin{bmatrix} -4.6392 & 0 \\ 0 & -4.6392 \end{bmatrix} \\
K_{13} = K_{31} &\begin{bmatrix} -0.6173 & 0 \\ 0 & -0.6173 \end{bmatrix},
\end{aligned}
\tag{7.6}
$$

where $i, j \in [1, 2, 3], i \neq j$.

Fig.7.8 gives the evolution of the open chain group movement. From the figure, as same in the case of closed chain case, the three robots start moving at three different initial positions and try to get closer to their own desired trajectories without any collision. After two seconds the group tracks the desired trajectory and keep the

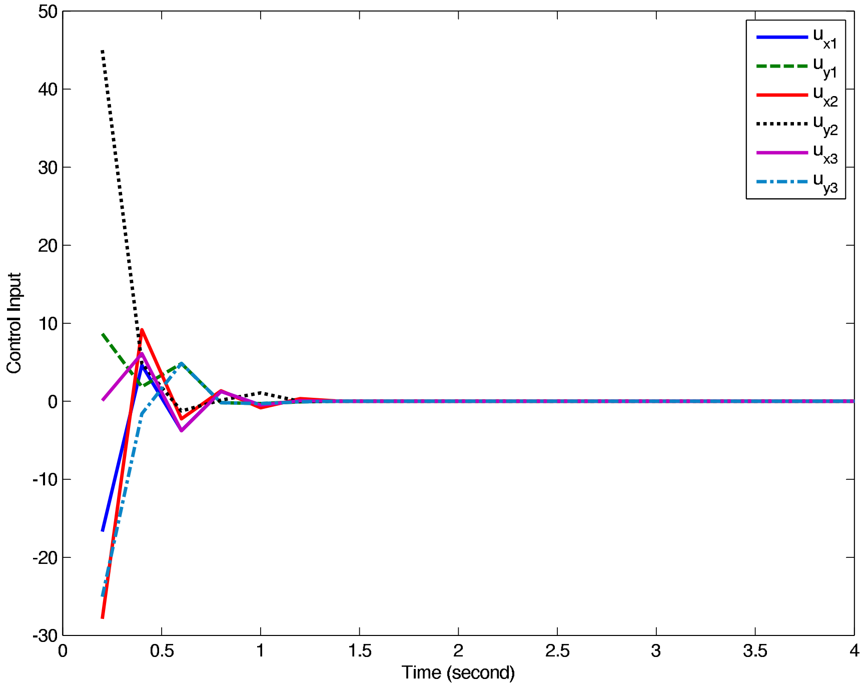


Figure 7.10. Control input signals of open chain robot group

accurate formation maintenance.

The tracking error between closed chain group movement trajectory and its own desired trajectory has been shown by Fig.7.9. The control input signals sent by the controller on each of the three robots has been recorded in Fig.7.10. From the figures, the same as in the case of closed chain group, the controller on each of the robots sends control input signals with different values to the robot based on how big the tracking error between the robot and the desired position is. Since the controller has been designed well, in the two second transition time, the controller adjusts the robot to its desired position without any collision. After two seconds, control signals converge to zero as the tracking errors converge to zero.

The results discussed above show that since the distributed consensus control algorithm and the control gain are designed properly, for both configurations the control objective of the whole group can be implemented and the effect caused by the networked induced time delay can be reduced as good as possible. In the next subsection, the performance of the group movement with group communication and without communication will be discussed and compared. The comparison shows why and how group communication plays an important role in consensus formation control of networked multi-agent robotic system.

7.2.2 Group Performance Comparisons: With and Without Group Communication

In this subsection, two cases of robot group movement have been investigated. In the first case, one three-robot group with group communication which takes place under the closed chain configuration has been asked to track the virtual leader's trajectory and keep accurate group formation. The dynamics of each robot is the same as the one of the closed chain case in subsection.7.2.1. In the second case, another three-robot group without group communication has been asked to track the same trajectory and keep the same group formation. For both cases one disturbance signal has been added to one of the three robots at the same time during the movement process to disturb the tracking performance of the robot. The dynamics of each robot in this group is the same as the previous one except

$$A_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{7.7}$$

where $i, j \in [1, 2, 3], i \neq j$.

The tracking performance of the three-robot group with & without group communication has been shown in Fig.7.11 and Fig.7.12 respectively. By comparing these

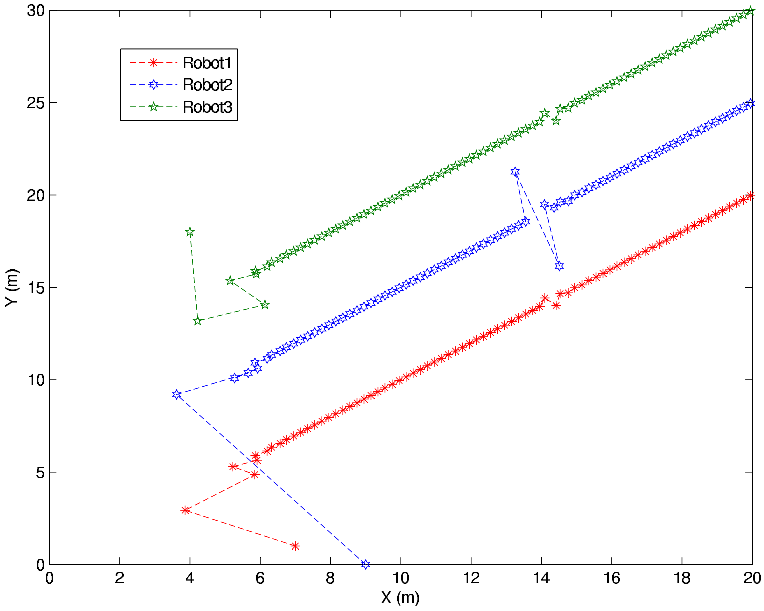


Figure 7.11. Evolution of closed chain group movement with disturbance

two group movement evolutions two differences have been found and discussed.

The first difference between the two cases is the transitional time (the time difference between the beginning time of the system movement and the time when the system reaches consensus) movement evolution. In the transitional time, the three robots in the first group start moving from different initial positions and exchanging information which includes their relative position to their neighbors and relative distance between each of them. So during the movement each of them can “feel” the relative position and velocity of its neighbors and set its own velocity and orientation based on the information from itself and its neighbors. To do so can make each of them stay in a safe distance from each other and bring the whole group a better robustness, and then the three robot can react with each other well and perform as a real group. On the other hand, the three robots in the second group cannot exchange those information. They set their velocities and orientations only depending on the

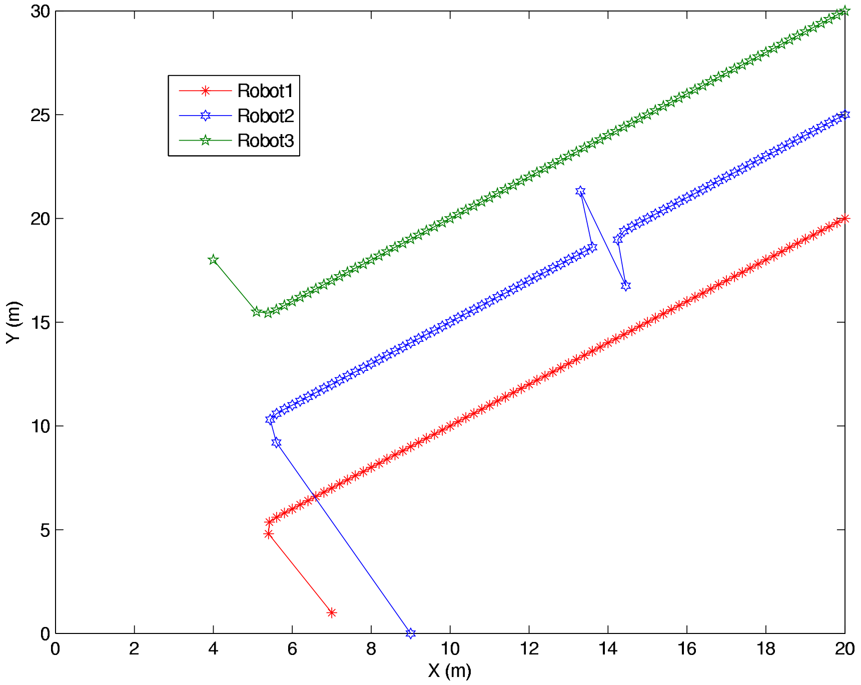


Figure 7.12. Evolution of no neighbor communication group movement with disturbance

information of themselves. So each of them moves independently and the whole movement cannot be treated as “group movement”. Ignoring its neighbors’ existence also results in a higher collision possibility during the movement.

The second difference between the two cases is the group anti-disturbance ability. As shown in the figures, a disturbance signal has been input to one of the robots at the same time during the movements for both two cases. When robot 2 in the first group receives the disturbance, its position gets a sudden change as shown in Fig.7.11. Since the group has the communication coupling between neighbors, the other two robots get this information and react to this change as soon as they receive the information from robot 2. So the whole group obtains a good formation maintenance during the disturbance. The Fig.7.12 shows that the robot 2 in the second group performs the same as the one in the first group when it gets the disturbance signal. However, since

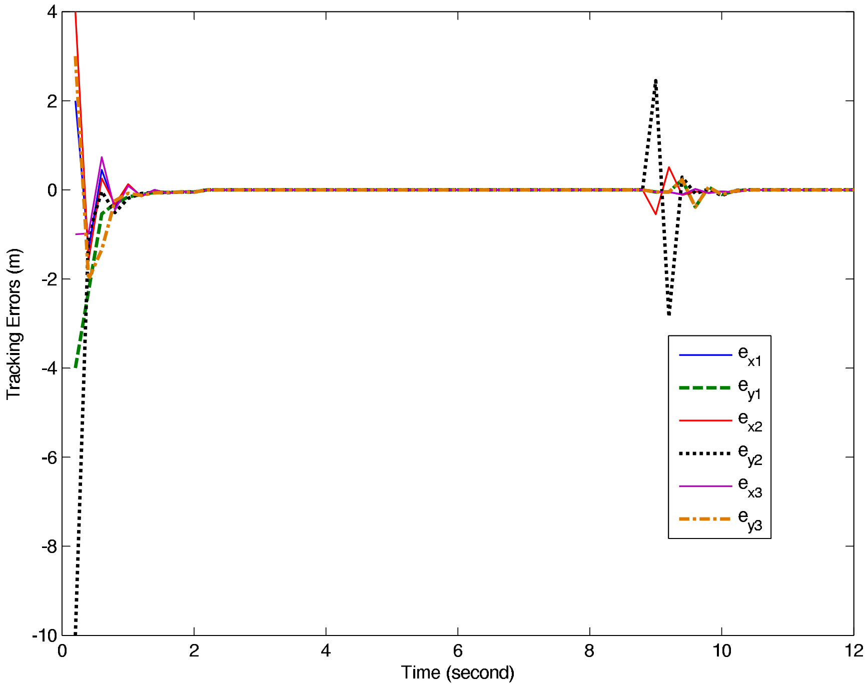


Figure 7.13. Tracking Error of closed chain group movement with disturbance

there is no information exchange between the three robots, the other two robots have no reaction to the position change of their neighbor and this results in not only a higher collision possibility, but also a bad group formation maintenance.

Another interesting comparison between the two cases has been done towards the tracking errors between robot group states and the desired states. The tracking errors for both cases have been represented in Fig.7.13 and Fig.7.14. As shown in Fig.7.13, in the first two second transitional time, the state tracking errors of the three robots react and affect with each other. When robot 2 receives the disturbance, its tracking errors get a jump from zero. The tracking errors of the other two also react this jump after they received this information from robot 2. While for case two which has been shown in Fig.7.14, the three robots do not affect with each other, and the disturbance received by robot 2 cause the same jump on its tracking errors as in case one but did

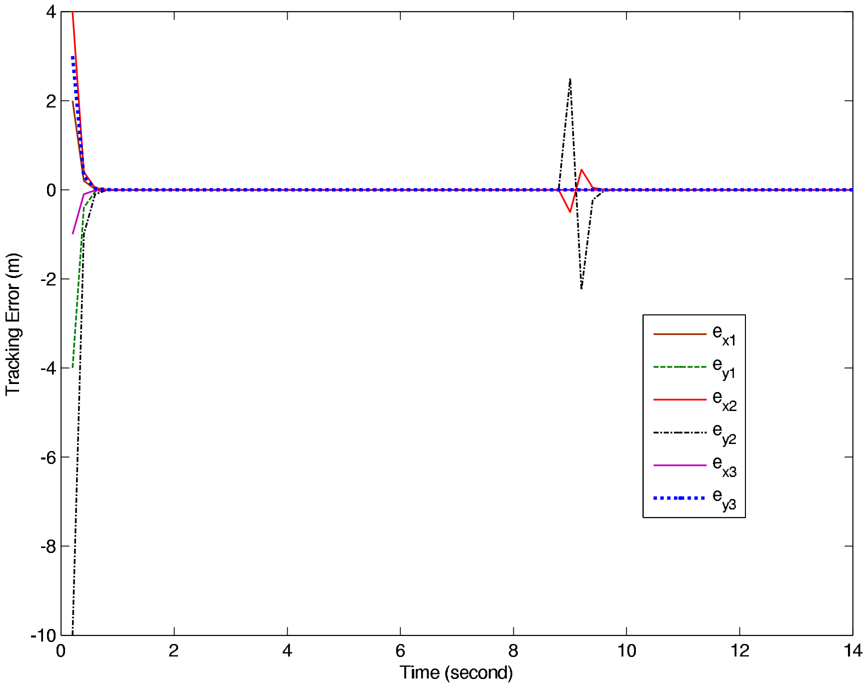


Figure 7.14. Tracking Error without neighbor communication and group movement with disturbance at 8.8 second

nothing to the tracking errors of the other two.

The disturbance area in Fig.7.11 has been zoomed in and represented in Fig.7.15 to get more details. As shown in the figure, before 8.8 seconds the three-robot group is moving on its desired trajectory with a stable group formation. At 8.8 seconds during the movement robot 2 receives the disturbance signal and its position has been pulled off its desired trajectory by this disturbance. Since there is a networked induced time delay in the group communication channel, the other two robots do not react to this position change. So during the short 0.2 seconds time delay period, robot 2 gets close to robot 3 and far from robot 1. This causes a potential collision and a damage to the group formation. After the 0.2 seconds delay the other two robots receive this information from robot 2 and adjust their speeds and orientations according to this information. This adjustent shown in the Fig.7.15 begins at 9 second, robot 3 starts

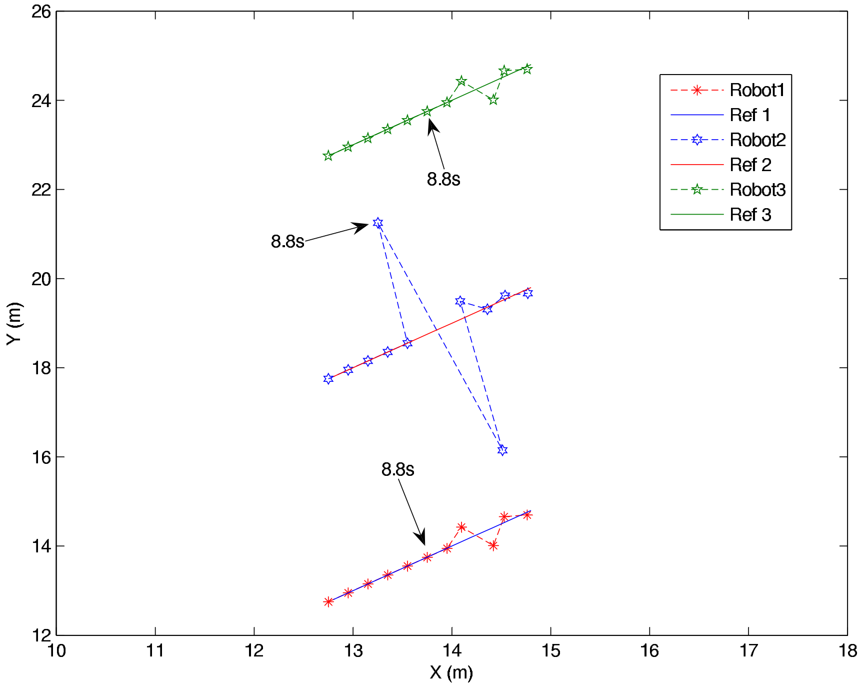


Figure 7.15. Evaluation of disturbance area in closed chain group movement

getting far away from robot 2 and robot 1 starts getting close to robot 2. It just looks like that the robot 3 and robot 1 has been pushed away and pulled in by the robot 2 during the process. When robot 2 trys to adjust its position back to the desired position at 9 second, the whole process has been reversed again until the whole group reaches consensus. The whole process shows that information exchange through group agents can avoid potential collisions and maintain the group formation during any unknown happening as well as possible.

In order to get more details from the group movement in the first case, more figures have been drawn. Fig.7.16 shows the control input signals sent by the controller of the three robots. The velocities and accelerations of the three robots have been shown in Fig.7.17and Fig.7.18; Fig.7.19 and Fig.7.20 show us the angular velocities and angular accelerations of the robots.

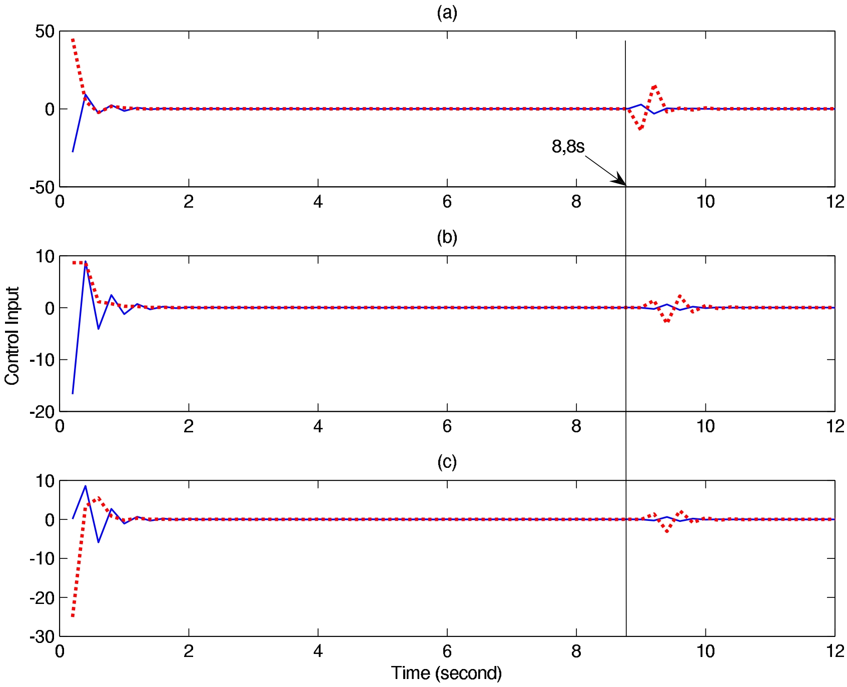


Figure 7.16. Control signals of closed chain group with disturbance on Robot 2: (a) Robot 2; (b) Robot 1; (c) Robot 3

From these figures, during the first two second transitional time, the controller attached on each robot in the group sends control input signals to make the whole group track the virtual leaders' trajectories and reach consensus. Each robot in the group exchanges information. The information includes the relative positions to its neighbors. Both the relative distance to its desired trajectory and the relative positions to its neighbors have been considered in the control input calculating process. Fig.7.16 shows that the control input signals sent by the three robots' controllers own a kind of "match" characteristic. This characteristic is more visible in Fig.7.17, Fig.7.18, Fig.7.19 and Fig.7.20 which show us the velocities, acceleration, angular velocities and angular acceleration. From these figures, at the beginning of the movement, each robot gets different accelerations and velocities since they stay at different initial positions and have different virtual leaders to follow. Then they start moving with group

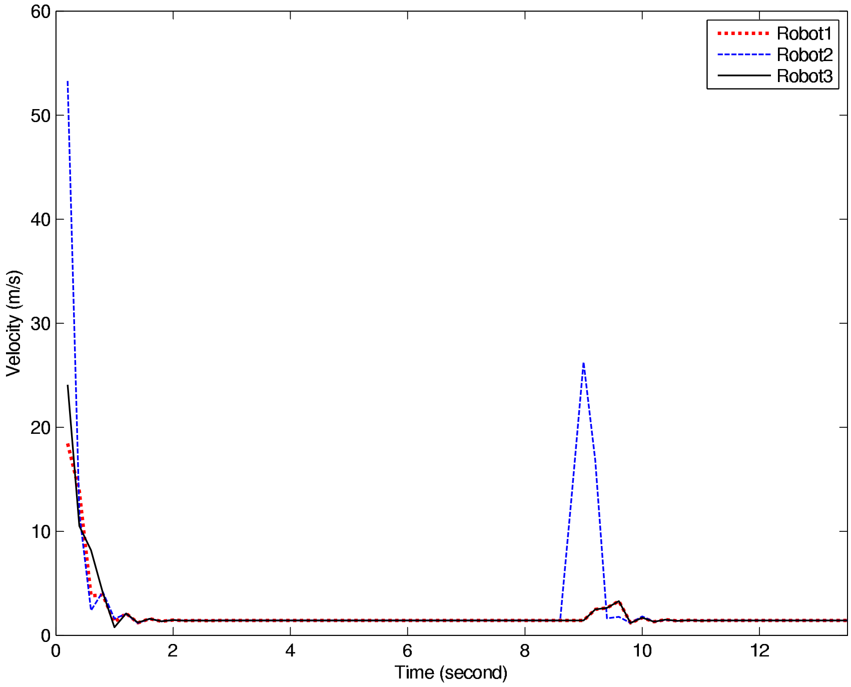


Figure 7.17. Speeds of closed chain group movement with disturbance at 8.8 second

communication so that their velocities and accelerations can match with each other and avoid potential collisions. As shown in Fig.7.18 and Fig.7.20, it takes the whole group approximate one second after moving to make the acceleration of each member match with each other well. This means the speed and orientation change trends of all members meet an agreement, and then the whole group will move together until it reaches consensus. Fig.7.17 and Fig.7.19 show that from one second after the group starts the movement the whole group reaches consensus and from two seconds after the movement each robot moves with the same constant speed and zero angular velocity.

During the disturbance period starts since 8.8 seconds, as shown in these figures, the speed and orientation of robot 2 has been changed by the disturbance (Fig.7.17; Fig.7.19). Fig.7.16 shows that the controller attached on robot 2 reacts to this change

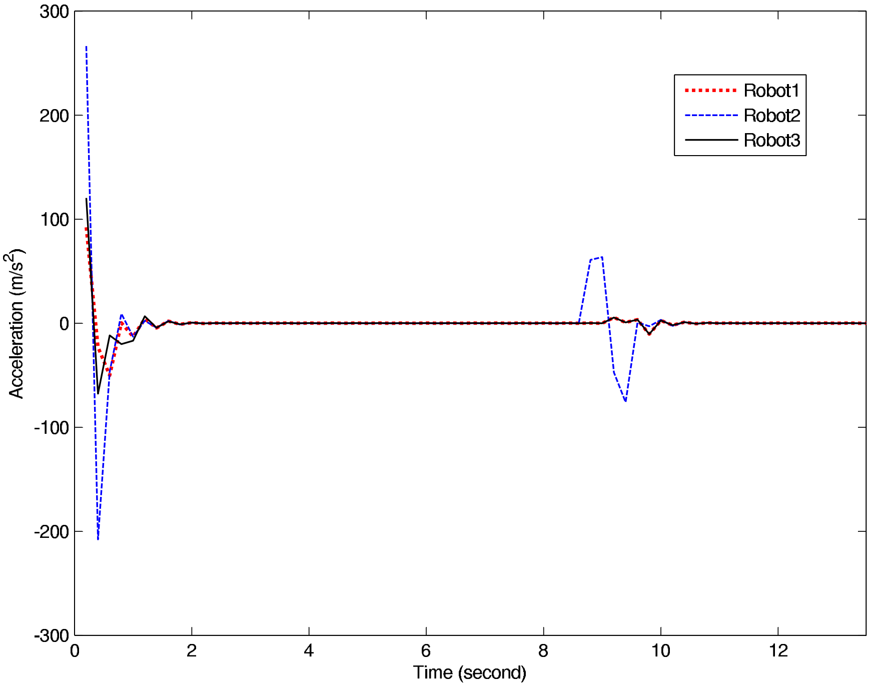


Figure 7.18. Accelerations of closed chain group movement with disturbance

and adjusts the control input values to fix this position migration. When these information from robot 2 has been received by the other two robots, the controllers attached on them also adjust the control input values to make these two robots match the position and velocity change of robot 2. Then a potential collision and maintain the group formation can be avoided as well as possible. Finally the group reaches consensus and each robot moves with the same constant speed and zero angular velocity again.

The above discussion and comparison shows that information exchange plays an important role in multi-agent systems. In other words, even a control law with only local information is sufficient to guarantee a good tracking performance, however, the coupling between neighbors improves group robustness and reduces formation maintenance error.

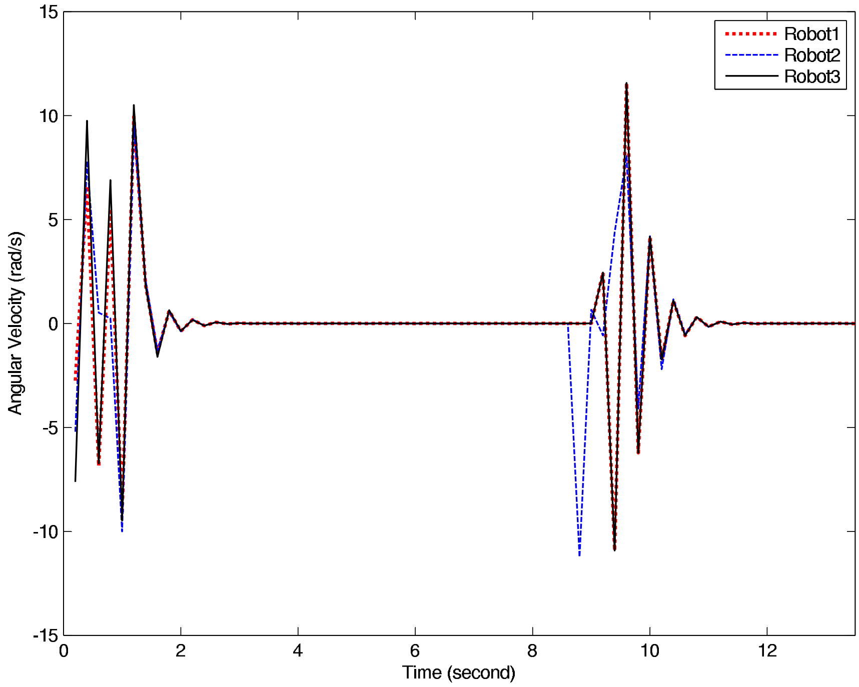


Figure 7.19. Angular Velocities of closed chain group movement with disturbance

7.3 Numerical Example of Consensus Formation Control of Multiple Mobile Robots

In this section, the simulation results regarding the work described in Chapter 6 have been shown. The main results show that the proposed distributed consensus control algorithm can 1): make the two-wheel robot group, which has communication delays inside, track the desired trajectory as well as possible; and 2): reduce the formation maintenance error as well as possible. The same with section.7.2, this section has been divided into two parts.

In the first part, two robot groups with the same dynamics and different group communication delays have been asked to track the same virtual leaders' trajectory and maintain the same group formation. The simulation results show how different communication delays will affect the group performance and how the controllers

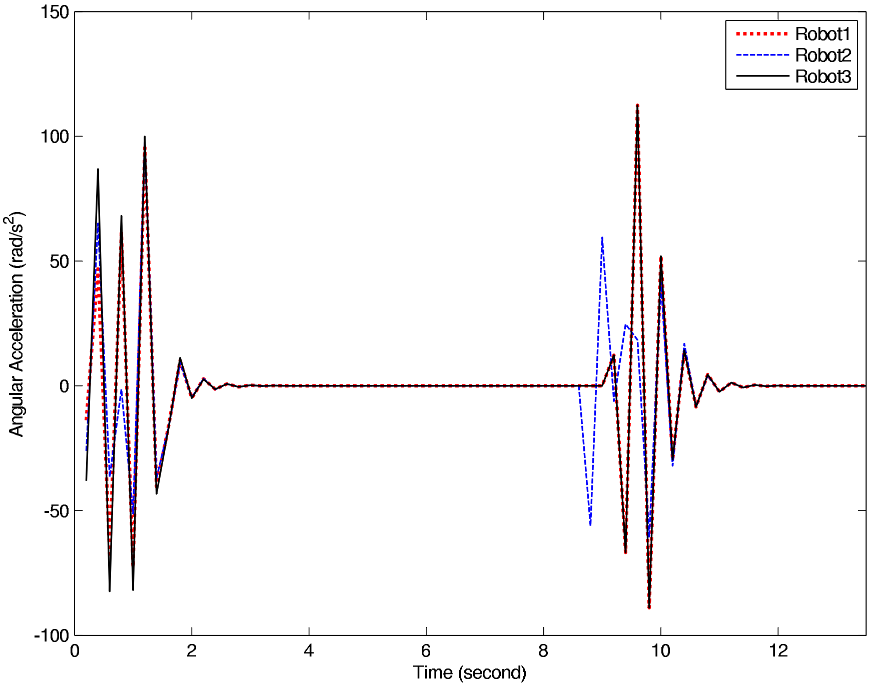


Figure 7.20. Angular Accelerations of closed chain group movement with disturbance

attached on the robots adjust the group movement evolutions.

In the second part, the same group has been asked to track the desired trajectory under two cases: in the first case there is group communication during the group movement; in the second case there is no group communication at all. Comparing the two cases shows us the success of the consensus control algorithm designed in this work and the importance of group communication in multi-robotic vehicle systems.

7.3.1 Group Performance Comparisons: Short and Long Time Delays

In this subsection, two robotic vehicle groups are asked to track the virtual leaders' trajectories with desired group formation and different time delays. The robotic vehicle's dynamic equation has been represented in Chapter 6. The element values in

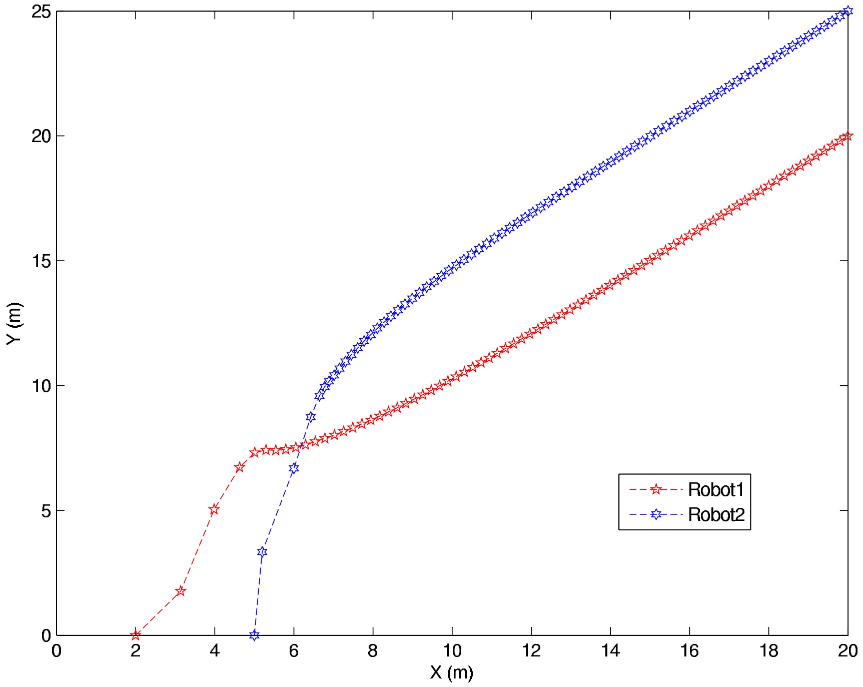


Figure 7.21. Evolution of the two robot group movement with short time communication delay ($\tau = 0.2second$)

the matrices are given as:

$$A_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.8}$$

where $i, j \in [1, 2], i \neq j$.

The two robots start moving at different initial points which are located in a plane as $[x_0, y_0]_{R_1} = [2, 0]$; $[x_0, y_0]_{R_2} = [5, 0]$. The desired trajectories for each robot are $R_1 : y_d = x_d$ with initial point at $x_d(0) = 5, y_d(0) = 5$; $R_2 : y_d = x_d + 5$ with initial point at $x_d(0) = 5, y_d(0) = 10$. The group movements include two cases: group movement with short time delay and group movement with long time delay.

A. Short Time Delay Case

In this case, the two robotic vehicles in the group move with a 0.2 second group communication delay. The consensus control algorithm in Chapter 6 has been applied

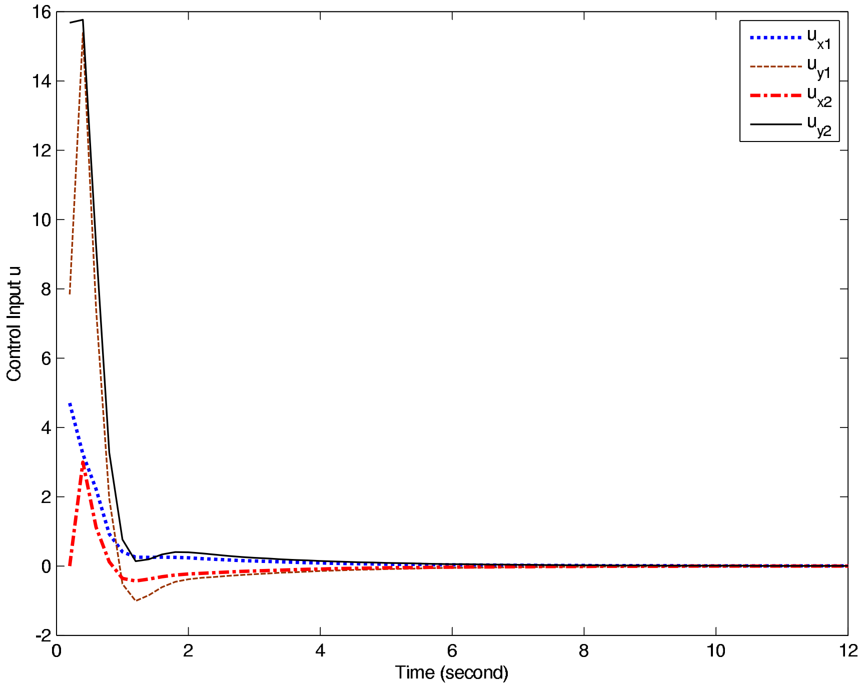


Figure 7.22. Control signals of the two robot group with short time communication delay ($\tau = 0.2 second$)

to adjust the whole group movement performance. The control gain has been designed based on sufficient conditions in the theorem as

$$K = YX^{-1} = \begin{bmatrix} 1.2026 & 0 \\ 0 & 1.2026 \end{bmatrix} \tag{7.9}$$

where

$$\begin{aligned} X &= \begin{bmatrix} -5.6453 & 0 \\ 0 & -5.6453 \end{bmatrix} \\ Y &= \begin{bmatrix} -6.1877 & 0 \\ 0 & -6.1877 \end{bmatrix}, \end{aligned} \tag{7.10}$$

$i \in [1, 2]$. The evolution of the group movement and the control input signal sent by the controllers have been shown in Fig.7.21 and Fig.7.22.

Since the connections between initial positions and desired trajectories of two robots cross with each other, after the group starts moving as shown in Fig.7.21

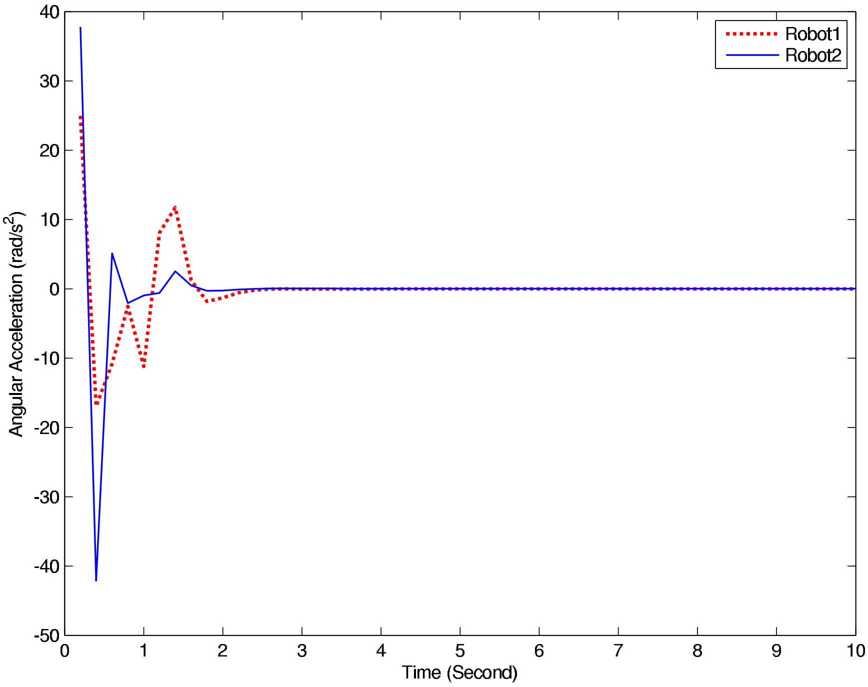


Figure 7.23. Evolution of group angular acceleration with short time communication delay ($\tau = 0.2 second$)

robot2 needs to cross robot1 at a proper time and a proper position so that they would not collide each other. After that time each robot keeps the desired distance with the other one to form the group formation and track the desired trajectory as well as possible. Note that the time difference between two index in Fig.7.21 is 0.2 second.

The controllers attached on two robots play an important role in the whole process. As shown in Fig.7.22, before the whole group reaches consensus at 6 seconds the controller of each robot keep sending signals, which are calculated based on the position information of the robot and its neighbor, to the robot for adjusting its relative position to its desired trajectory and neighbor. But at the beginning of the movement, because of time delays each robot has not received the information from its neighbor yet, the controller calculates the control input value which will be used to adjust

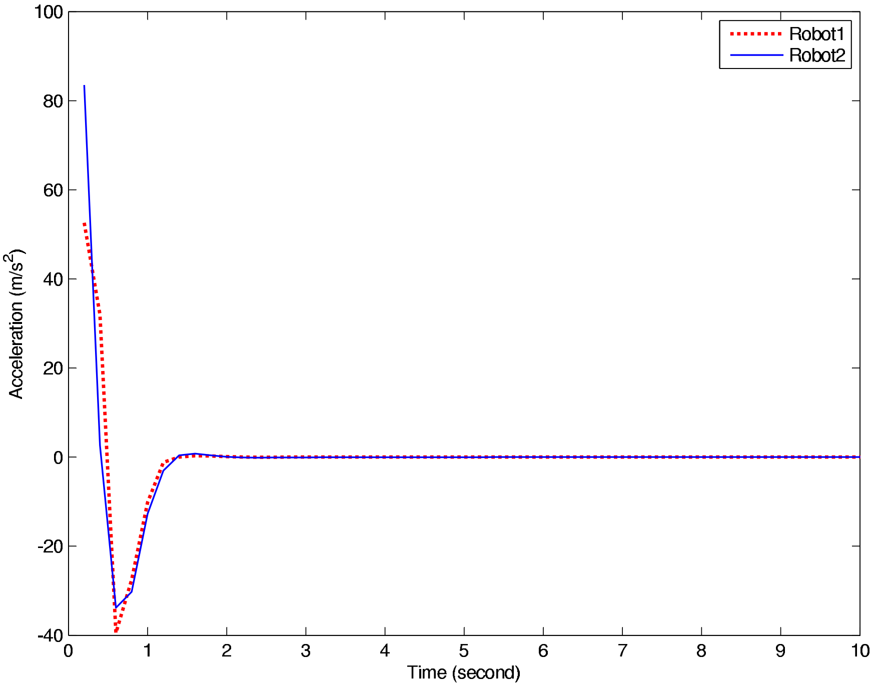


Figure 7.24. Evolution of group acceleration with short time communication delay ($\tau = 0.2 second$)

the velocity and orientation of the robot depending on the distance between robot's initial position and initial desired position only. The evolutions of the velocity and orientation of two robots have been recorded down and shown by Fig.7.23-Fig.7.26. At $t = 0.2$ seconds each robot has received the information from its neighbor, robot 1 reduces its angular acceleration in order to reduce its angular velocity toward its desired position as shown in Fig.7.21, Fig.7.23 and Fig.7.25. However, robot 1 does not reduce its angular acceleration too much so that it would not bump with robot 2. While for robot 2, it reduces its angular acceleration much enough to reverse its angular velocity to avoid getting too close to robot 1. There is only a small difference between their linear acceleration and speed until the group reaches consensus at about eighth seconds after movement as shown in Fig.7.24 and Fig.7.26. After 0.6 seconds robot 2 has passed by the desired trajectory of robot 1 and starts tracking on

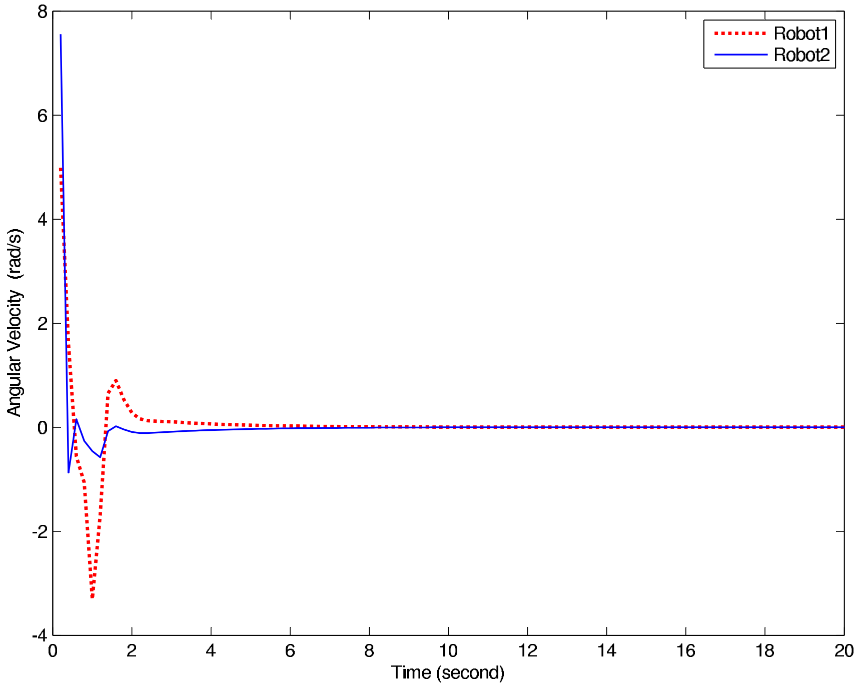


Figure 7.25. Evolution of group angular velocity with short time communication delay ($\tau = 0.2 second$)

its own desired trajectory. At the same time robot 1 receives this information from robot 2 and adjusts its moving orientation toward its desired trajectory by setting new angular accelerations. It takes about eight seconds for the two robots to adjust their positions and track the desired trajectory without any possible collision. The tracking errors shown in Fig.7.27 converge to zero in 8 seconds from when the whole group reaches consensus. In the second case, the group performance with long time delay case will be discussed such as how the long time communication delay will affect the group performance and how the controller will reduce this effect caused by the long time delay.

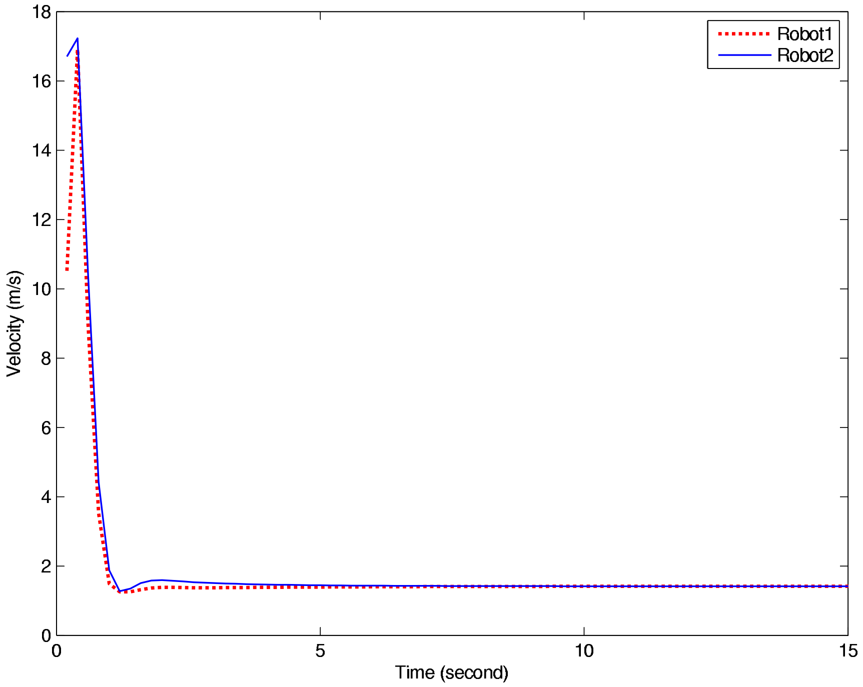


Figure 7.26. Evolution of group velocity with short time communication delay ($\tau = 0.2 second$)

B. Long Time Delay Case

In the pervious case, the robot group performance with short group communication delay has been discussed. In this case, two robotic vehicles in the group move with a 1 second group communication delay. The consensus control algorithm in Chapter.6 has been applied to adjust the whole group movement performance. The control gain has been designed based on sufficient conditions in the theorem as

$$K = YX^{-1} = \begin{bmatrix} 1.5675 & 0 \\ 0 & 1.5675 \end{bmatrix} \tag{7.11}$$

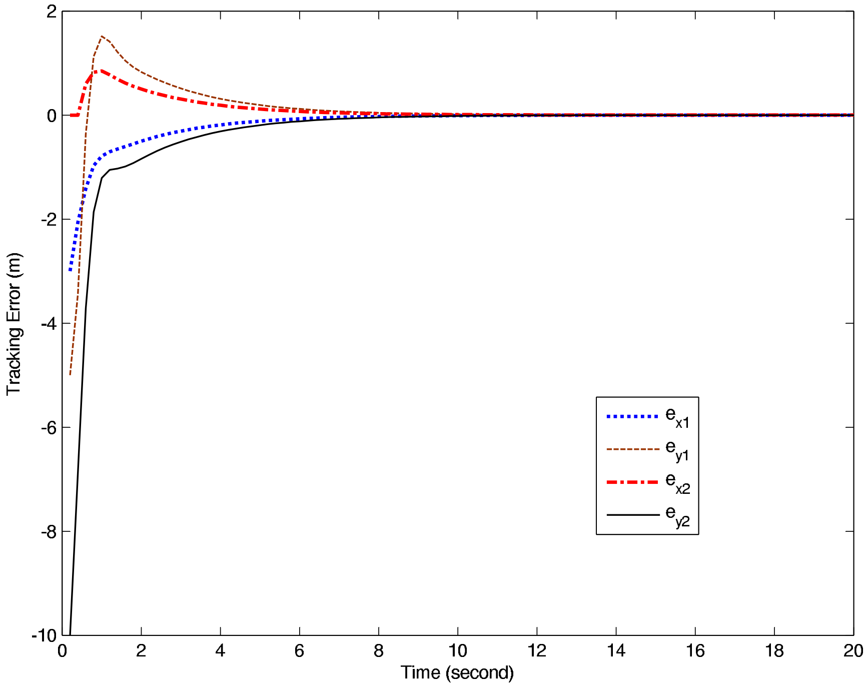


Figure 7.27. Evolution of group tracking errors with short time communication delay ($\tau = 0.2second$)

where

$$\begin{aligned} X &= \begin{bmatrix} -6.7636 & 0 \\ 0 & -6.7636 \end{bmatrix} \\ Y &= \begin{bmatrix} -10.6019 & 0 \\ 0 & -10.6019 \end{bmatrix}, \end{aligned} \tag{7.12}$$

$i \in [1, 2]$. The evolution of the group movement and the control input signal sent by the controllers have been shown in Fig.7.28 and Fig.7.29. Compared with Fig.7.21, the main differences in Fig.7.28 focus on the $10m$ by $20m$ transitional domain. So this domain has been zoomed in and shown in Fig.7.30. As shown in Fig.7.30 from 0 to 0.4 second (the time difference between two index is 0.2 second) each of them adjusts its orientation toward its desired position by adjusting its angular acceleration as shown in Fig.7.31. From 0.4 to 1 second the value of both angular velocity becomes close

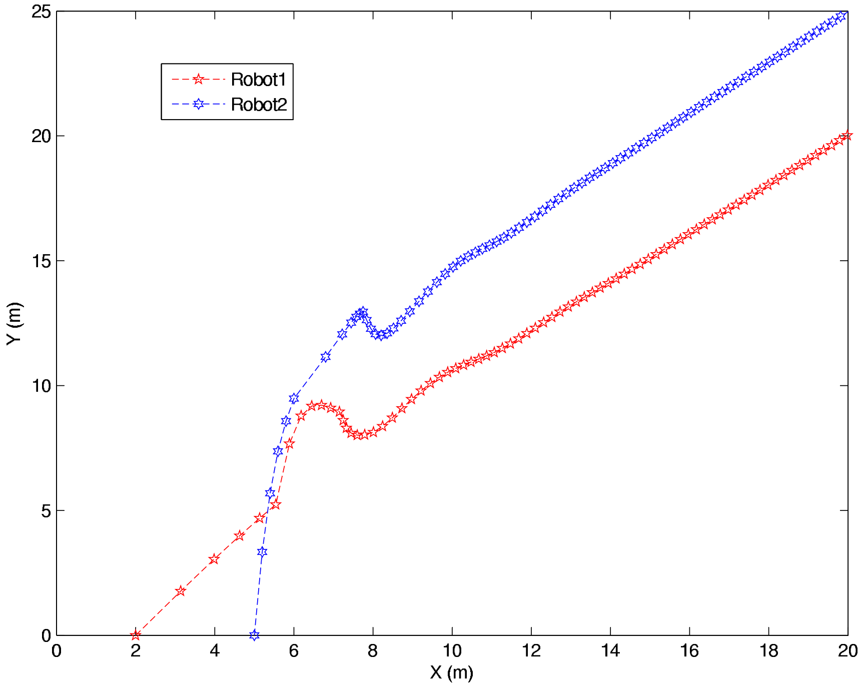


Figure 7.28. Evolution of the robot group movement with long time communication delay ($\tau = 1 second$)

to zero as shown in Fig.7.32, which means the orientation of each robot keeps fixed toward the desired trajectory during this period. The linear acceleration and speed of the robot group shown in Fig.7.33 and Fig.7.34 shows during the first second the acceleration and speed of each robot is reduced as much as the distance between the robot and its desired position, which is shown in Fig.7.35, is reduced. It is because of the group communication delay that each robot cannot receive any information from its neighbor in the first second and runs ignoring its neighbor's existence.It is not hard to see that there is a risk for a possible collision of the two robots if one of them changes its initial position and the group formation does not appear or it is not even going to appear in the first second.

As shown in the Fig.7.30 at 1 second after the group starts moving, the position information for 0 second of each robot has not been received by each other. Since the

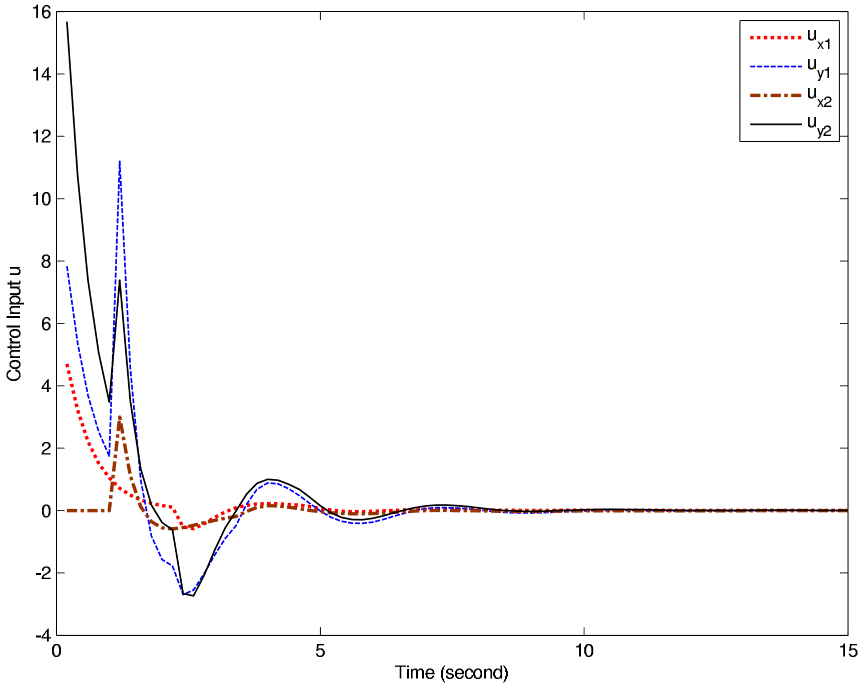


Figure 7.29. Control signals of the robot group with long time communication delay ($\tau = 1second$)

delayed communication cannot accurately provide the current position information of the neighbor for each robot, the speed and orientation of robots have been affected by this delayed signal greatly as shown in Fig.7.31 and Fig.7.33. The tracking error in Fig.7.35 also shows before 1 second the tracking error on x-axis direction has been close to zero but affected greatly by the delayed signal from its neighbor at 1 second. Since the control gain has been designed well to reduce the effect caused by the delay, the tracking errors gradually converges to zero in the next 15 seconds.

For Fig.7.30, at 1.2 second, the controller starts working to adjust the speed and orientation of the robot. From 1.2 second to 1.6 robot 2 reduces its speed by adjusting its control input shown in Fig.7.29 to change its acceleration shown in Fig.7.33 because it “feels” it is getting too far from its neighbor’s “1 second ago position” and tries to keep the desired team formation with robot 1. For robot 1, it changes its direction by

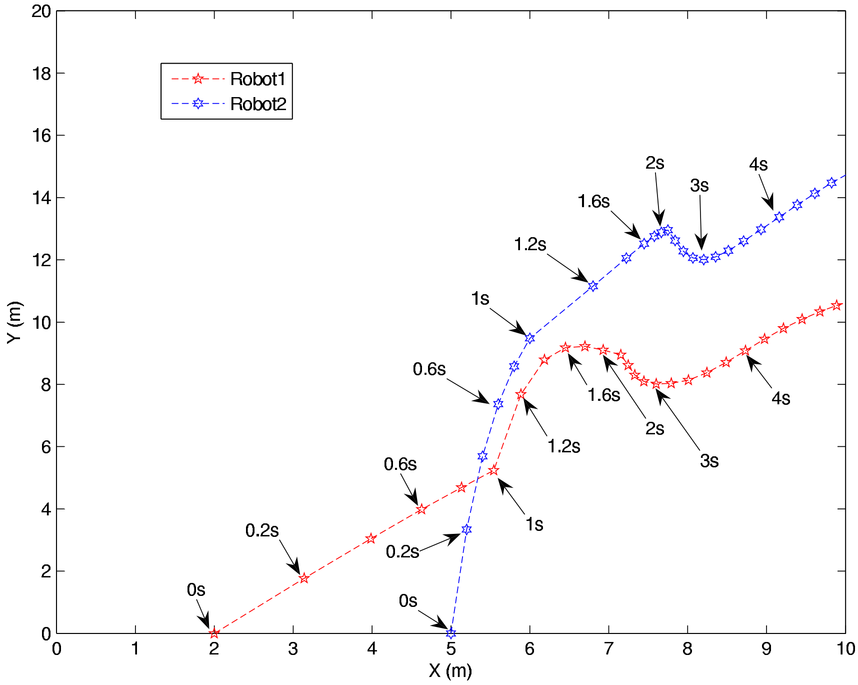


Figure 7.30. Evolution of the robot group movement in transitional domain with long time communication delay

reducing its angular acceleration and increases its speed since it “feels” it is getting close to the 1 second ago position of robot 2. The good stabilization ability of the controller can be seen specially from 1.6 second to 3 second, the two robots start matching its neighbor’s movement and the whole team formation starts appearing. At 2 second, robot 2 changes its direction greatly to avoid getting to far with robot 1’s position at one second ago, while for robot 1, it also changes its direction with the same trend as robot 2 to avoid getting too close to robot 2’s position at one second ago. After 3 second, as the tracking errors converge to zero gradually and the team formation is being maintaining, the effect caused by the communication delays has been reduced as well as possible. More details for this adjusting process, such as angular velocity and speed, are shown in figures discussed above.

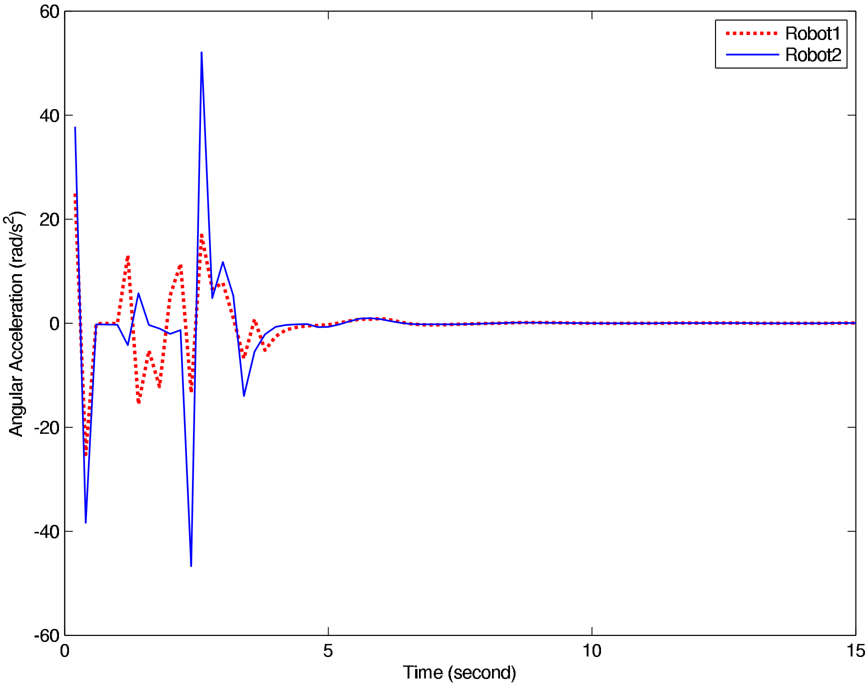


Figure 7.31. Angular acceleration of the robot group with long time communication delay ($\tau = 1 second$)

Compared with short communication delay case, two things should be paid attention: 1. long time communication delays may cause a higher possibility of collision than short time delay; 2. long group communication time delay can result in a longer transition time for the system than short time delay case. Because of the well designed control gain, the effects caused by long time delay are reduced well as discussed in this section. In next subsection, the transition time performance of the group movement with and without group communication will be discussed. By analyzing the difference of the two cases can tell us why the group communication is so important in consensus formation control of networked multi-agent robotic vehicle systems.

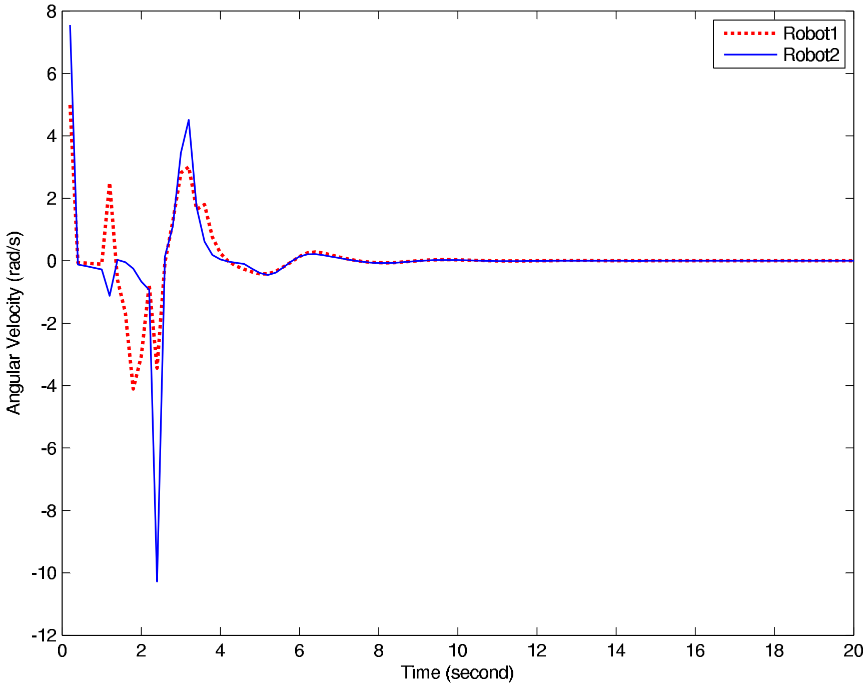


Figure 7.32. Angular velocity of the robot group with long time communication delay ($\tau = 1 second$)

7.3.2 Group Performance in the Transition Time: With and Without Group Communication

In the previous subsection, the performance of the group movement with short and long group communication delay have been discussed and compared. From the results in previous subsection, how the group communication delay could affect group performance has been learnt. How can the controller designed based on the theoretic work in Chapter.6 reduce the effects caused by the delay and implement the control objective as well as possible also has been learnt.

In this subsection two robot groups are asked to track the same virtual leaders' trajectories with the same desired group formation and 0.2 second group communication delay. The control gain is the same as the one used in the short time delay case. The only difference between the two groups is during the movement one group

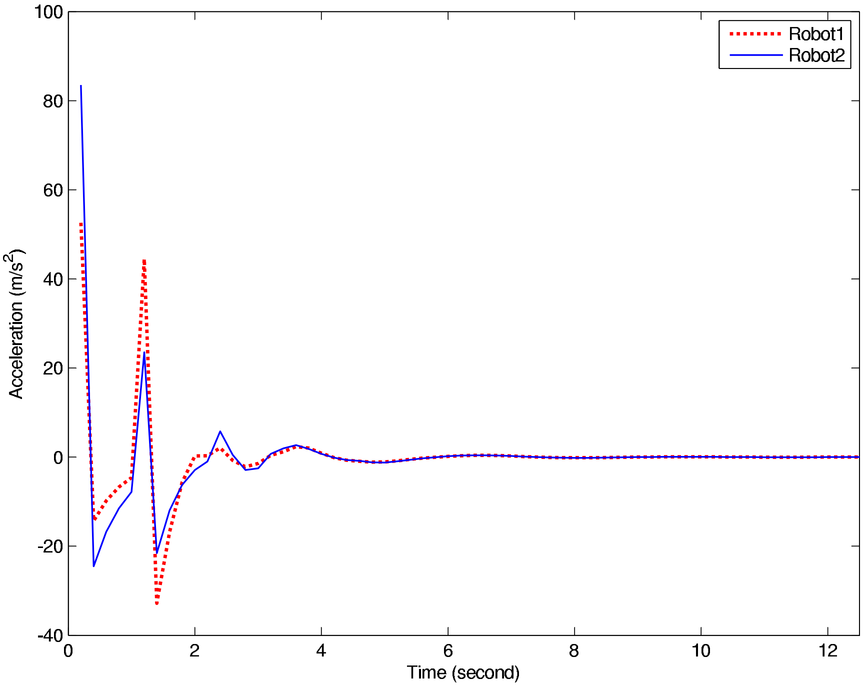


Figure 7.33. Acceleration of the robot group with long time communication delay ($\tau = 1second$)

has group communication and the other does not. In order to test the stabilizing ability of the proposed control approach, the virtual leaders' trajectories are designed in a more complicated case. For robot 1, the desired trajectory is $x_{d1} = 1.5cos(t)$; $y_{d1} = 1.5sin(t) + 10$. For robot 2, it is $x_{d1} = cos(t)$; $y_{d1} = sin(t) + 10$.

The evolution of the group movement without and with group communications have been shown in Fig.7.36 and Fig.7.37 respectively. These two figures show that finally both of groups can track the virtual leaders' trajectory and can maintain the desired group formation. However, there is an obvious difference between these two figures which focuses on the transition time area. In order to get more details regarding this, the transition time area in the figures, which includes the evolution of the group movement during the time period from the beginning of the movement to when the group starts getting in the elliptic orbit, has been magnified and shown in Fig.7.38

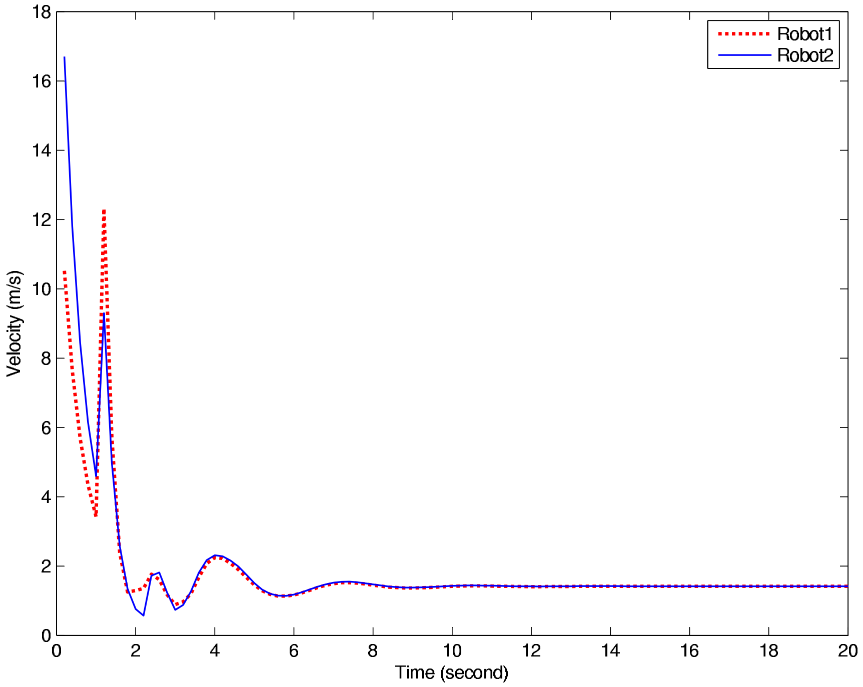


Figure 7.34. Speed of the robot group with long time communication delay ($\tau = 1 second$)

and Fig.7.39.

As shown in Fig.7.38, the group without group communication starts moving and getting close to the elliptic orbit during the first second. For the two robots in the group, there is not any group communication during the movement, they start getting close to each other gradually after they start moving. From 0.8 to 1 second, as shown in the figure, the running trajectory of each robot crosses with each other and then a high possibility of collision has appeared during this very short period. Compared with the group movement without group communication, the group movement with group communication shows the different sight. As shown in Fig.7.39 the two robots in this group start moving and getting close to the elliptic orbit while keeping a safe distance with each other. It shows that group communication plays a very important

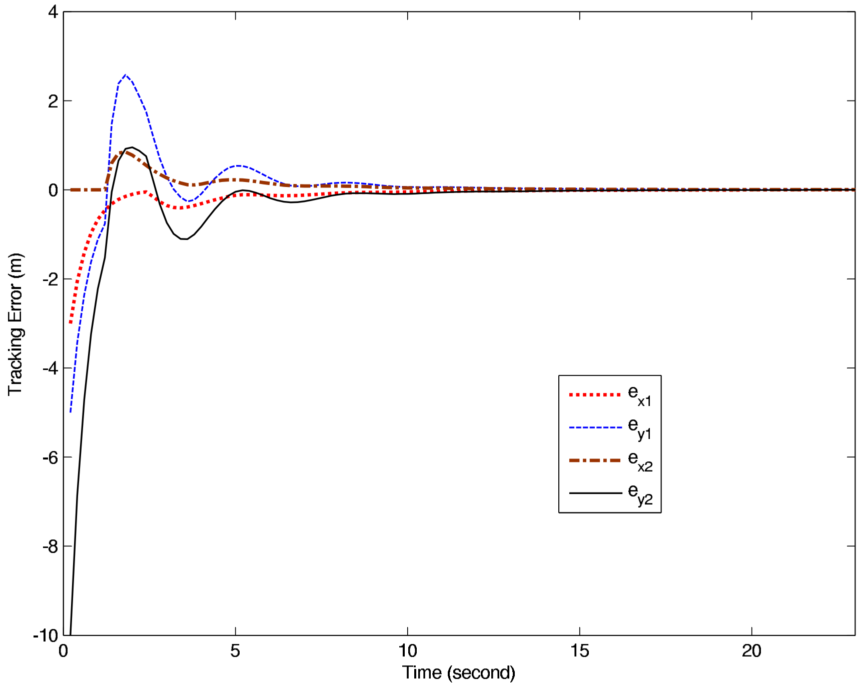


Figure 7.35. Tracking error of the robot group with long time communication delay ($\tau = 1 second$)

role in multi-agent system group movement by exchanging the position velocity information in group members. Each robot can "feel" its neighbor's position and velocity by receiving those information from the neighbor and adjust its own position and velocity based on not only its own states but also its neighbor's. Sharing information can avoid conflicting with its neighbor and drive the whole group to the orbit in a more safe way. Fig.7.39 shows that at 0.8 seconds each robot in this group starts getting in the elliptic orbit from its current position which has a safe distance with its neighbor's. Comparing Fig.7.36 with Fig.7.37 shows it takes the group without communication less time than the group with communication to get into the elliptic orbit, however, the group with communication can move in a safer way than the one without communication.

Drawing and comparing their control input signals in Fig.7.40 and Fig.7.41, their

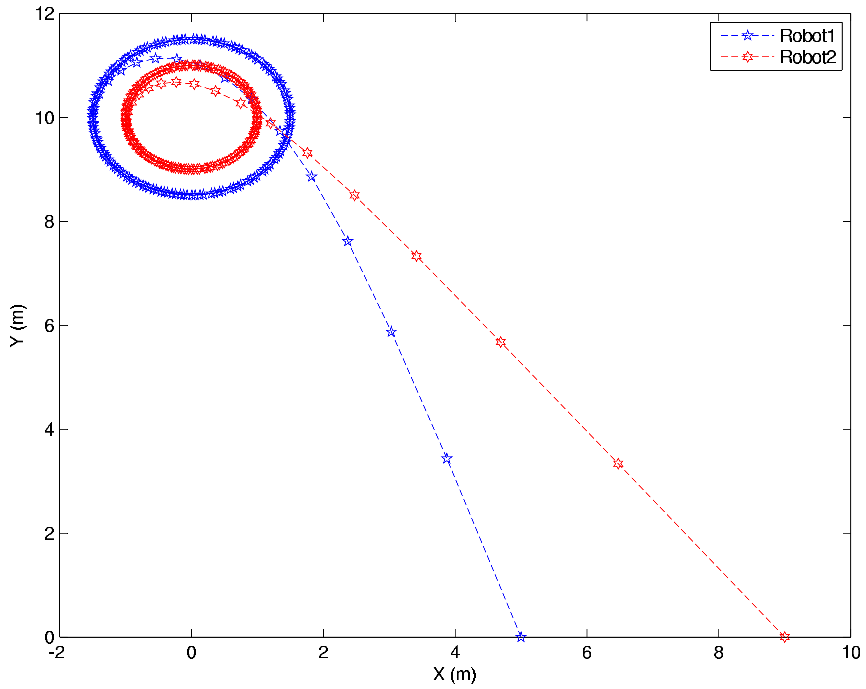


Figure 7.36. Evolution of the group movement without group communication

angular acceleration in Fig.7.42 and Fig.7.43 and their angular velocity in Fig.7.44 and Fig.7.45 show the same story. As shown in Fig.7.40 all control inputs of this group converge to zero in 4 seconds and during this transition period the control input values of each robot are similar with each other. This is because two robots have the same dynamic equation and they do not exchange information with each other. They cannot be treated as a whole “group”. The whole movement process just looks like two independent single system running respectively. So it is not hard to understand why there is a high possibility of collision appearing in this movement. While for the other group in Fig.7.41 the control inputs converge to zero in 8 seconds. That is because the controller on each robot of this group calculates the control inputs based on not only the states of itself, but also the states of its neighbor. It takes the group longer time than the other one to make each robot get into the elliptic orbit in the

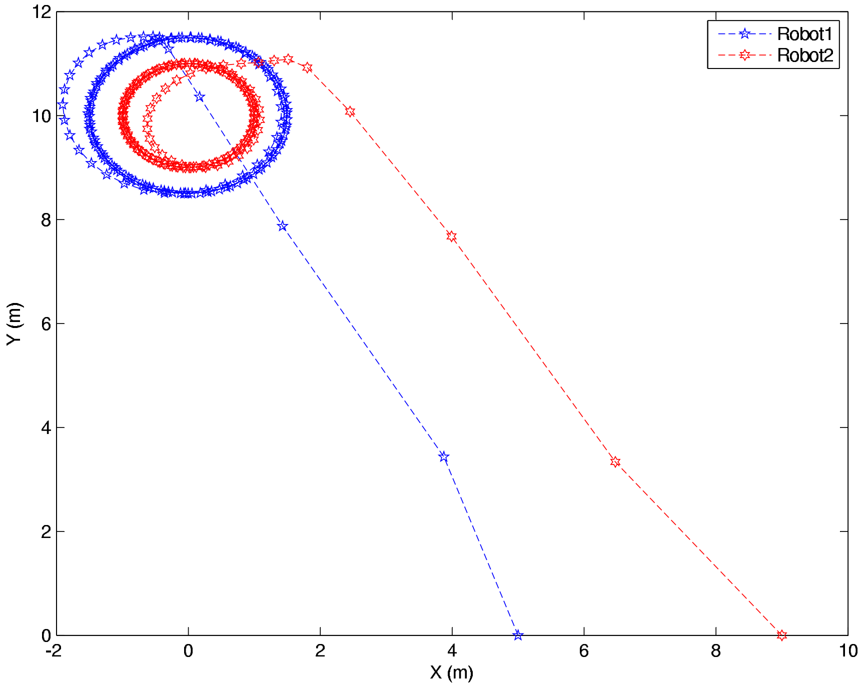


Figure 7.37. Evolution of the group movement with group communication

right way. By adjusting the control inputs in this way, each robot in this group gets different angular acceleration, different angular velocity and speed shown in Fig.7.43, Fig.7.45 and Fig.7.46 in the 8 second transition time. These differences can make sure that each robot can keep a safe distance with its neighbor when the whole group gets close to the elliptic orbit. For the group shown in Fig.7.42 and Fig.7.44, the angular acceleration and angular velocity are synchronized with each other even in the transition period. These two pictures show a high collision possibility during the movement.

For the group with communication the tracking errors have been represented in Fig.7.47. As shown in the figure, the tracking errors converge to zero in 8 seconds which supports that the consensus control algorithm in Chapter 6 has been designed properly. It can make the whole networked multi-agent robotic system reach consensus

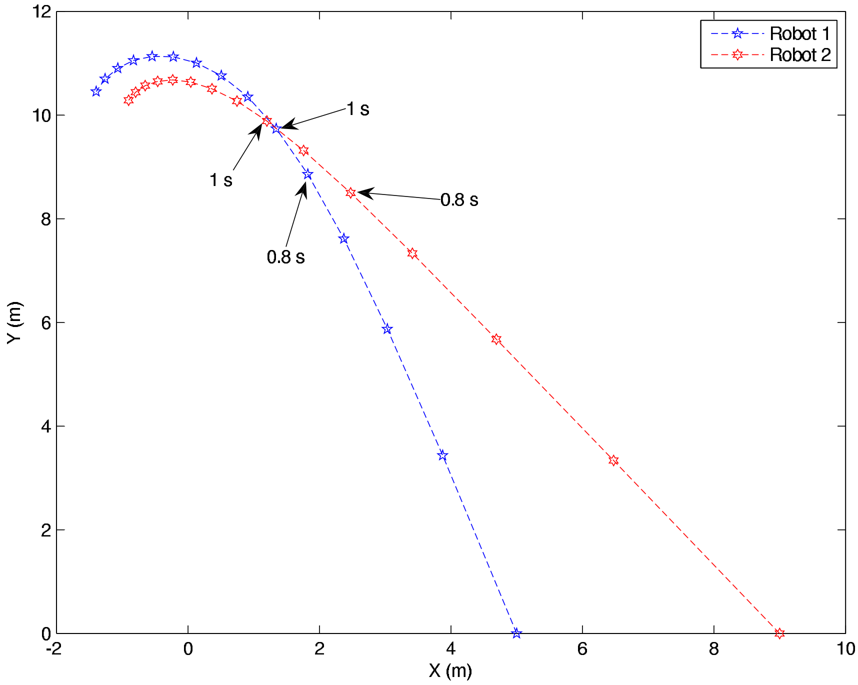


Figure 7.38. Group movement in transition time without group communication

and maintain the desired group formation. It also can reduce the effects caused by the communication delay.

As discussed in this subsection, it can be concluded that for consensus formation control of multi-agent system, even a control law which does not include the interact elements from neighbors is sufficient to guarantee the convergence of the tracking errors for the whole system; however, the coupling between neighbors can effectively improve group robustness and reduce formation maintenance error.

7.4 Summary

In subsection.7.1, the simulation results show the effectiveness and feasibility of the proposed approach. The sampled-data controller can stabilize the NCSs with random packet loss and varying time delay properly due to the proper design of the control

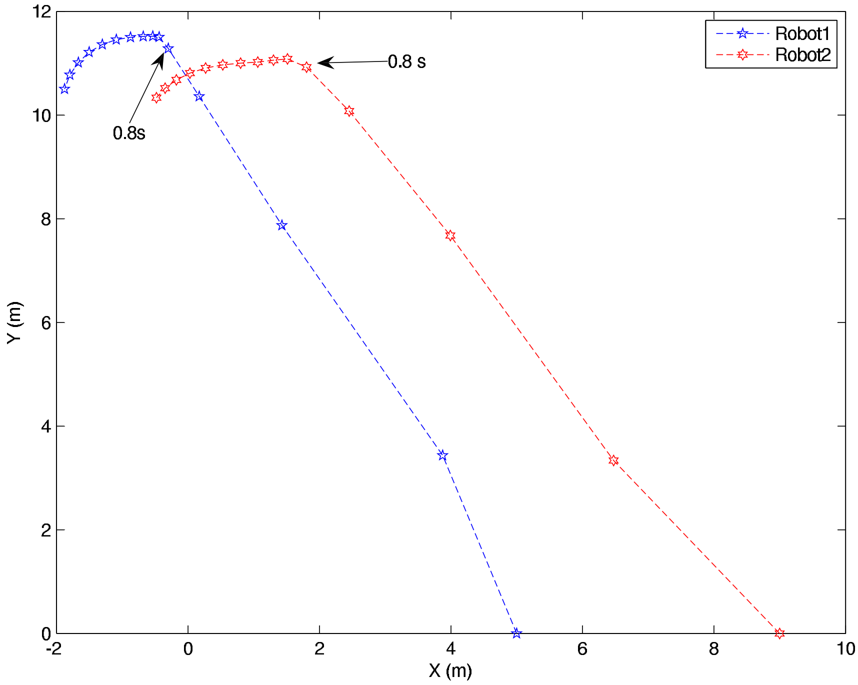


Figure 7.39. Group movement in transition time with group communication

gain. In subsection.7.2, the group performance with different configurations has been shown by simulations. The results support the theoretic design in this work. The importance of the group communication also has been shown. In subsection.7.3, the group performance with long ($\tau = 1 second$) and short delay ($\tau = 0.2 second$) have been compared. The effects caused by different delays have been shown. The results show the feasibility of the proposed strategy with different delays and group communication topologies. The importance of the group communication in group movement have been shown by the simulation results.

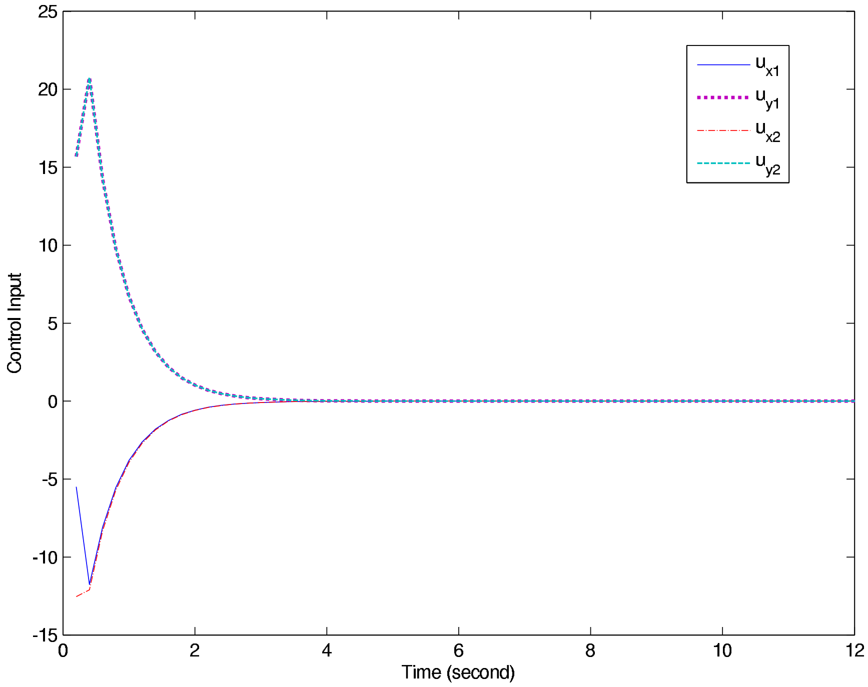


Figure 7.40. Evolution of control input for the group without communication

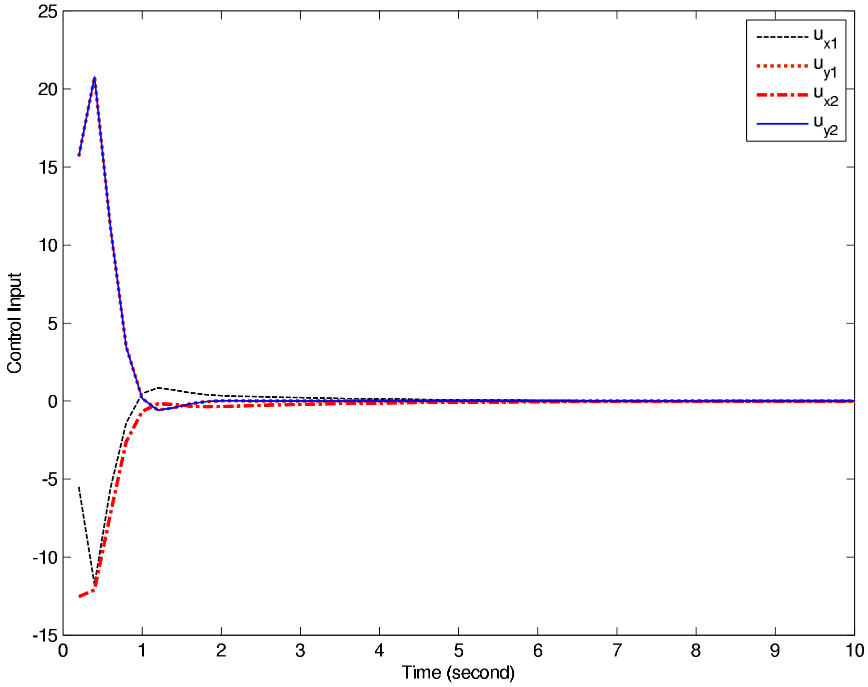


Figure 7.41. Evolution of control input for the group with communication

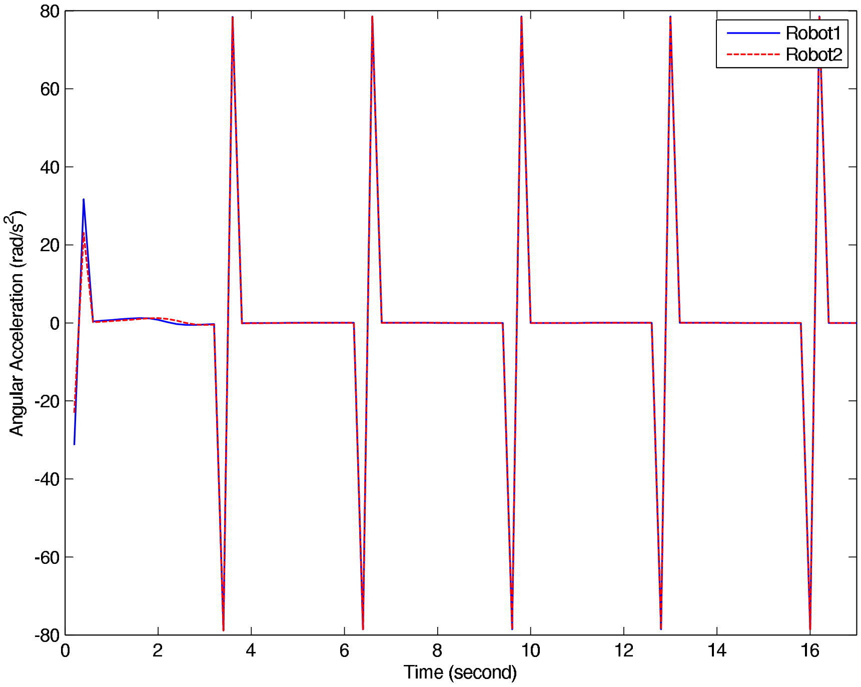


Figure 7.42. Angular acceleration of the group without group communication

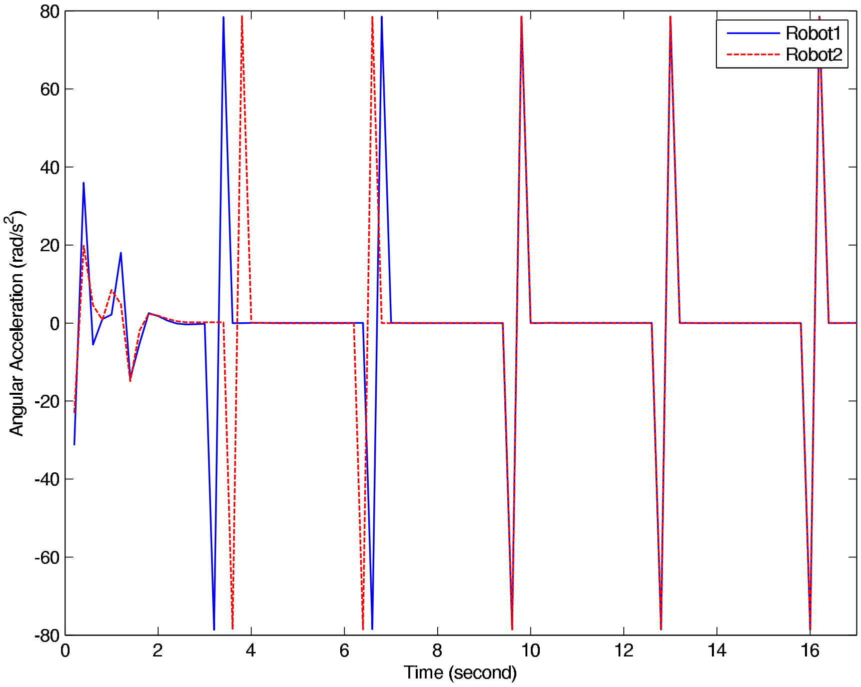


Figure 7.43. Angular acceleration of the group with group communication

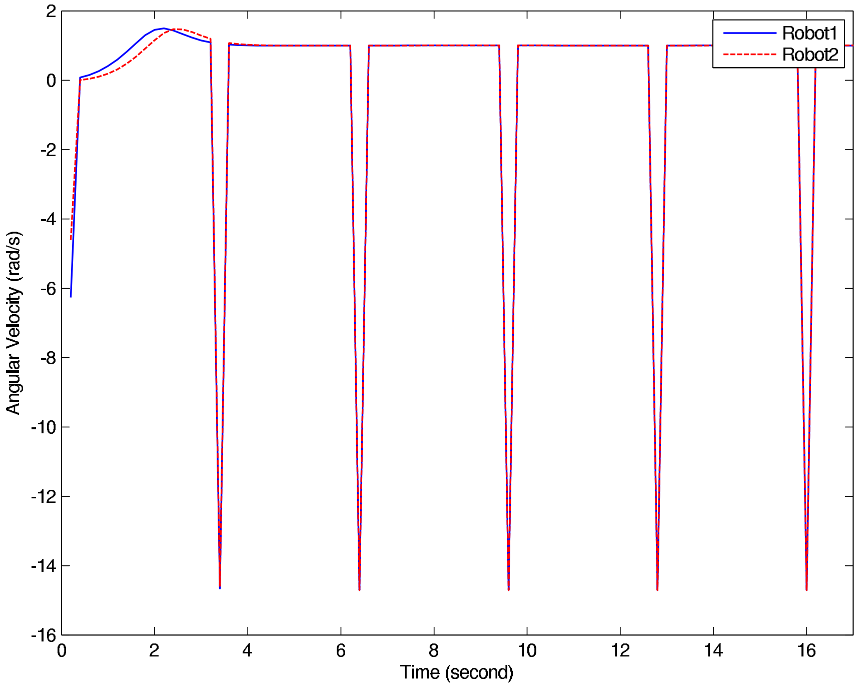


Figure 7.44. Angular velocity of the group without group communication

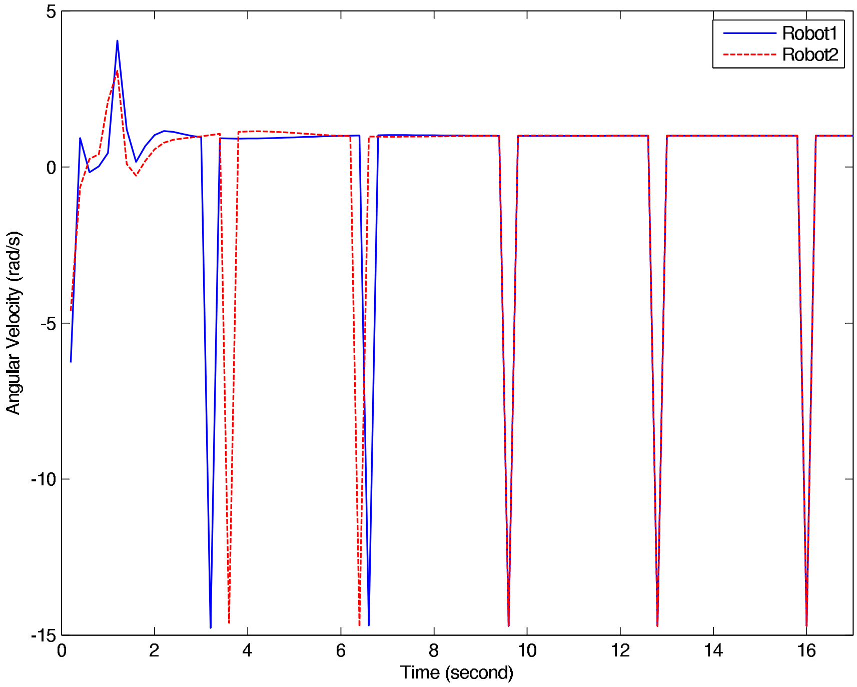


Figure 7.45. Angular velocity of the group with group communication

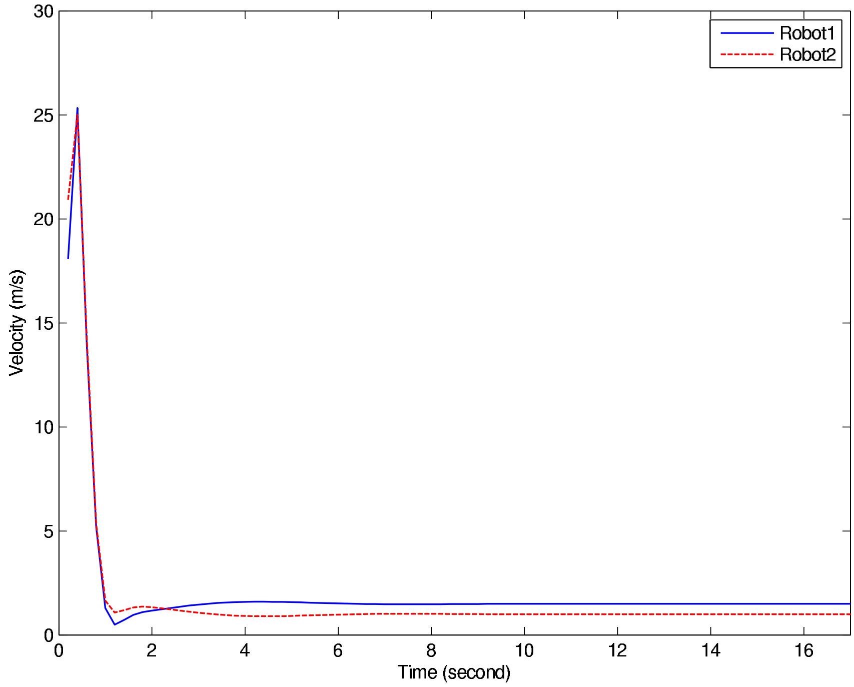


Figure 7.46. Speed of the group with group communication

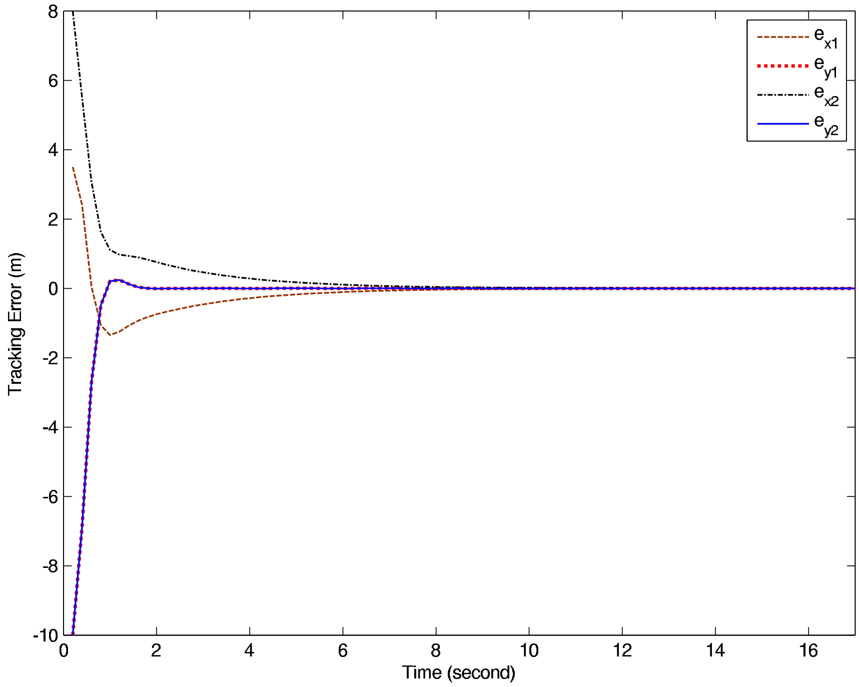


Figure 7.47. Tracking errors of the group with group communication

Chapter 8

Experimental Results

8.1 Sampled-data NCSs with Stochastic Packet Loss and Varying Time Delay

In order to study the network property in the real environment, a real-time network induced delay and packet loss measurement system has been developed based on MATLAB applications.

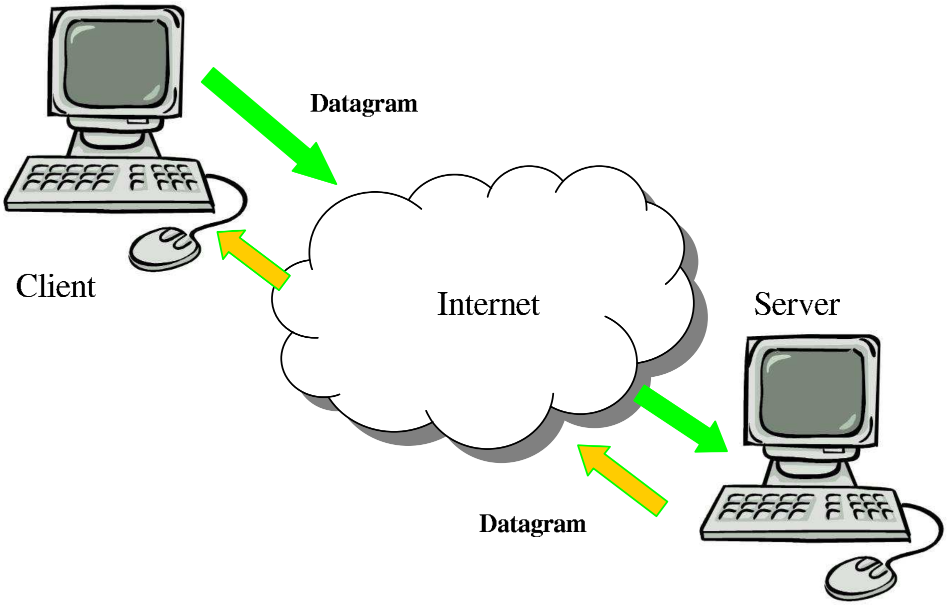


Figure 8.1. Real-time network measurement system

With the Instrument Control Toolbox in MATLAB, two computers located in different places can communicate with each other via networks following special protocols (e.g. TCP/IP and UDP) directly from MATLAB. The data generated in MATLAB on the client side can be sent out to another computer (server), in which a related MATLAB application is running synchronously, through the networks as shown in Fig.8.1. The connection and communication were established using UDP protocol. The delay and packet loss information could be recorded and saved in MATLAB for further analysis.

8.1.1 Network Induced Delays and Packet Losses

During one measurement, data packets have been sent with a fixed frequency from the client to the server following the receiver's IP address, and then they would be returned to the client at the same time when they have been received by the server. In this case, 6,000 data packets were transmitted between the client and the server once per hour on Feb.5th 2008 from 8:00am to 13:00pm and on Feb.6th 2008 from 12:00pm to 18:00pm, the average delay of each measurement is very close, the histogram of average delay versus hour index has been shown in Fig.8.2. The small difference is caused by the network load change over time; however, generally the network tested in the experiment keeps stable.

When the information of delays recorded in one measurement has been drawn in Fig.8.3, it can be found that the minimum time delay value is 10 ms and up to 93% of packets can be received in 50 ms, only 6% of packets have been received in the time which is longer than 50 ms, and those packets appeared randomly. Most of

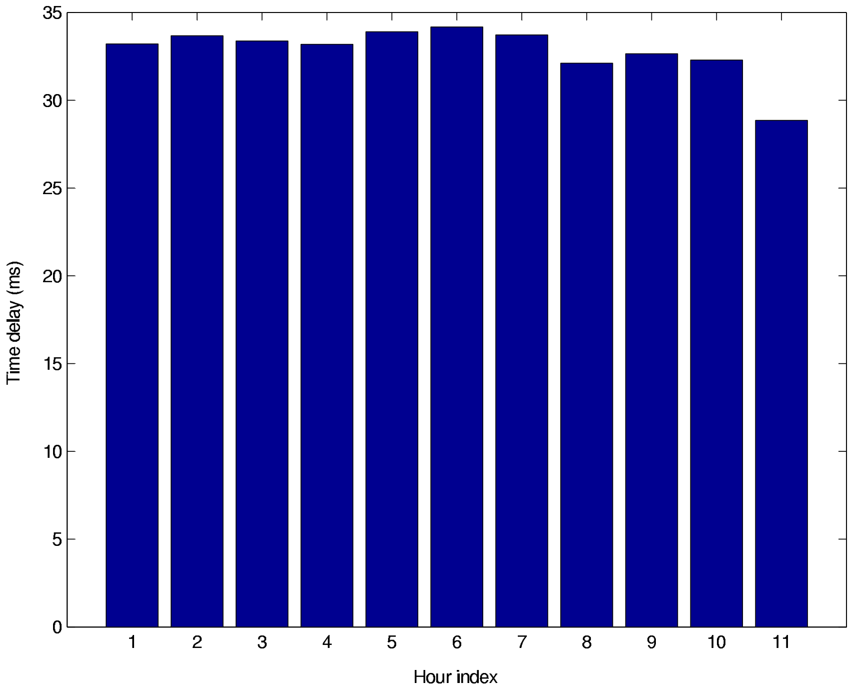


Figure 8.2. Histogram of average time delay vs hour index

the recorded delays are in the time domain from 15 ms to 50 ms, this network time varying delay property has been applied in the simulation.

The packet loss case is quite complicated and hard to predict. People believe that the network overload may cause packet loss, so even for the same network the situation when it is overloaded may be quite different from when it is idle. The packet loss rates in four different cases were recorded and analyzed as shown in the pie chart Fig.8.4. In order to vary the case of the network, FlashGet has been used to download and upload data files from an internet web service. The measurement in Case 1 was processed without running FlashGet. In Cases 2, 3 and 4 the FlashGet was used to download date files with the rate of 60 KB/s, 100 KB/s and 400KB/s respectively.

In the pie chart, the slices which present the packet loss rates in four cases were pulled out. It is clear that the packet loss rate increases as the download rate increases.

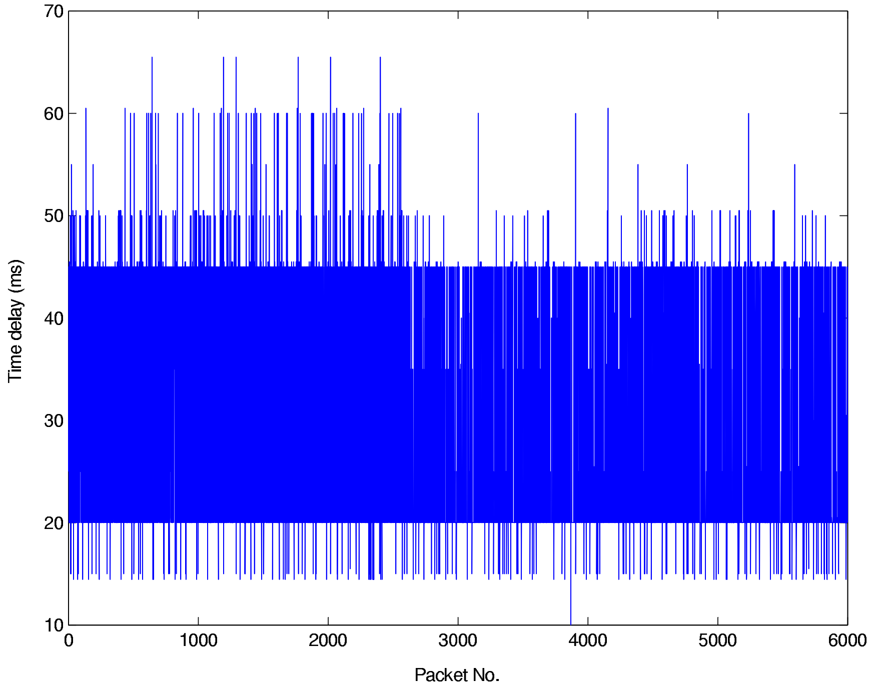


Figure 8.3. Time delay vs numbers of data packet

For the rest two slices, the small one presents the rate of the packets which have been received but exceeded the bounded delay value, and the big one presents the rate of the packets received in a shorter time than the bound delay value. For example, in Case 1 1% of packets was lost during transmissions, 6% was received in a longer time than the upper bound of the delay and 93% were received in the fixed time domain. From the result, the network tested in the experiment was keeping in a stable condition since over 90% packets could be received in a fixed time domain.

8.1.2 Bounded Delays and Bounded Packet Losses

In NCSs, there is usually a common assumption that the delay and the packet loss amount between two successful transmissions are bounded. This condition is imposed on network communications by dropping packets that exceed the delay bounds. The time delay distribution in one measurement can help us to establish the upper bound

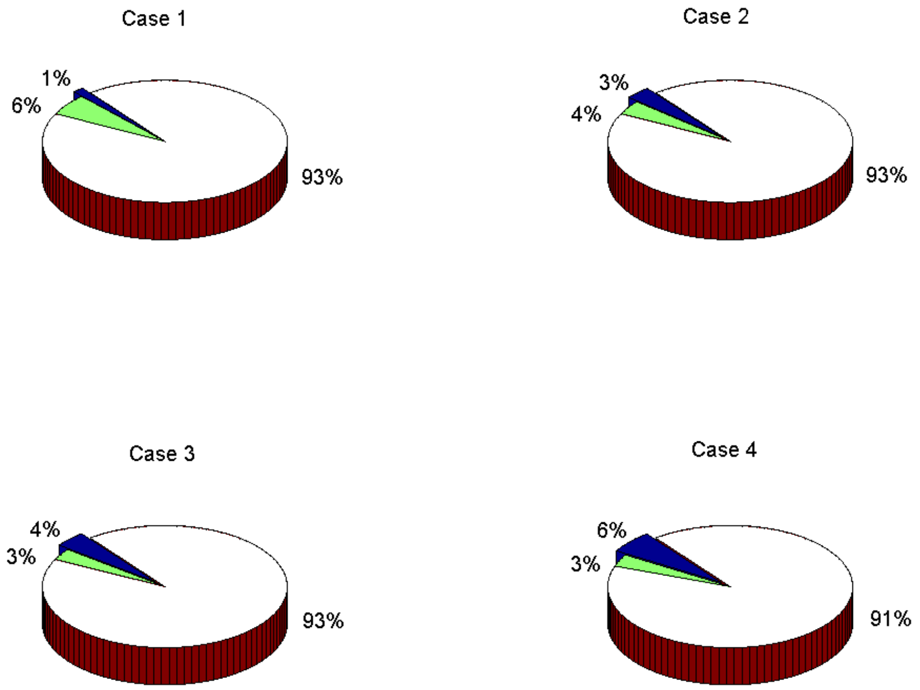


Figure 8.4. Pie chart of packet loss rate

of delays. The results shown in Fig.8.5 are still used here. In Fig.8.5 (a), it shows the delay distribution in one measurement. Since the minimum delay is 10 ms and over 90% packets arrived in 50 ms, the upper and lower bound of delays were chosen as 50 ms and 10 ms respectively. The packets exceed the bounded delay were dropped and the new delay distribution under bounded delay was shown in Fig.8.5 (b). The packets with large delay may become out-of-order. By dropping those packets can increase the perceived loss rate but can better represent the transmitted signal by not introducing artefact to NCSs. In the pie chart Fig.8.5 (c), the slice marked with 1% presents the original packet loss rate. The slice marked with 6% presents the rate of packets which exceed the bounded delay and were dropped. The slice marked with 93% shows us the data rate received in the bounded delay.

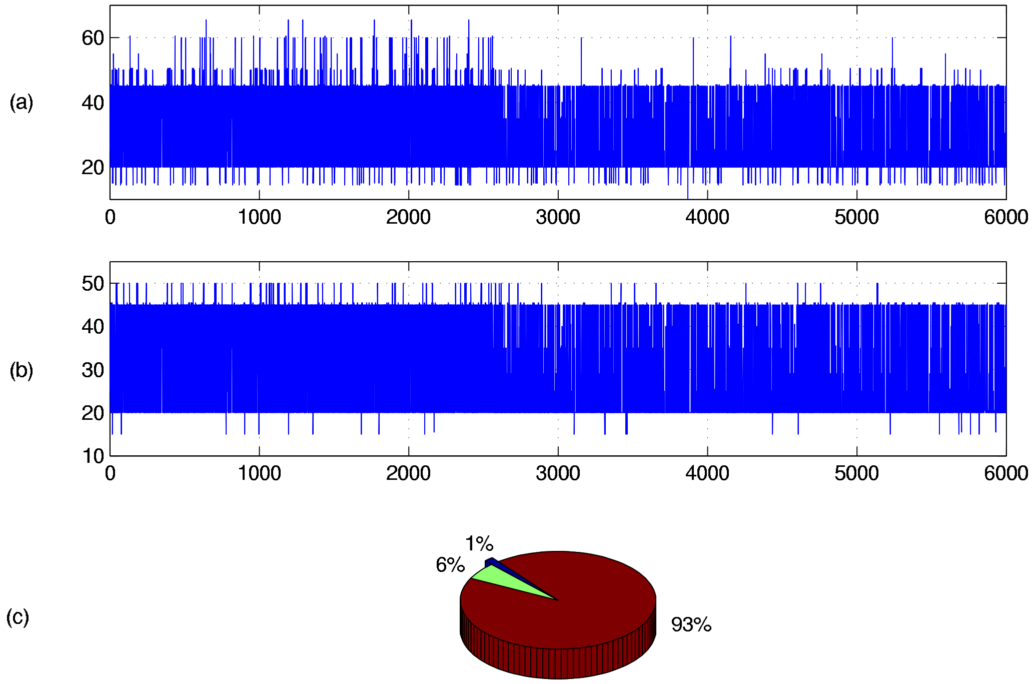


Figure 8.5. Bounded delay and packet lost

8.1.3 Markov Chains

Predicting packet loss is very difficult due to the lack of information of network conditions. For this reason, stochastic methods have been used to describe the networks from a statistical standpoint. During the study of this method, Markov Chain (MC) is considered to model the packet loss phenomena using conditional probabilities where the probability of moving from one state to the next depends only on the previous state.

As discussed in Section 4.3, in this work the packet loss process is governed by a Markov Chain. The order of the MC is determined due to the upper bound of the packet loss amount between two successful transmissions. From the results obtained from experiments (Fig.8.4), the upper bounds of the packet loss in Case 1 and Case 4 are both three packets, which means the maximum continuing packet loss amounts in both cases are three. With the experimental data from real network, a third-order MC probability transition matrix can be calculate. The number of transitions from state 1 to state 1 are divided by the total number of transitions from state 1 to state 1, 2 and 3. For each case, there are two different MC transition matrix. The first one is calculated before dropping the packets which exceed the bounded delay; the second one is calculated after dropping them. Only the transition matrices in Case 1 and 4 (Fig.8.4) are listed here as an example and those transition matrices are applied in the simulation example in subsection 7.1

Case 1, before dropping the packets exceed the bounded delay:

$$\prod = \begin{bmatrix} 0.9861 & 0.013 & 0.0009 \\ 0.9744 & 0.0256 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{8.1}$$

Case 1, after dropping the packets:

$$\prod = \begin{bmatrix} 0.9424 & 0.0541 & 0.0036 \\ 0.9666 & 0.0334 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{8.2}$$

Case 4, before dropping the packets exceed the bounded delay:

$$\prod = \begin{bmatrix} 0.9426 & 0.0553 & 0.0021 \\ 0.9159 & 0.0779 & 0.0062 \\ 0.9231 & 0.0769 & 0 \end{bmatrix}. \tag{8.3}$$

Case 4, after dropping the packets:

$$\prod = \begin{bmatrix} 0.9128 & 0.0788 & 0.0084 \\ 0.8984 & 0.0831 & 0.0185 \\ 0.92 & 0.08 & 0 \end{bmatrix}. \tag{8.4}$$

8.1.4 Real-Time NCSs Experiment

A real-time networked control system based on MATLAB is tested. The experiment setup is as same as shown in Fig.8.1. One computer is treated as the controller. Its task is to receive the response from the system and send control input to it. The system which is shown by (7.1) run on a second computer. The controller and system communicated with each other directly from MATLAB application through the real-time network channel with delays and packet loss occurring randomly. At each sampling time, which is the same as used in subsection. 7.1, the controller sampled the system response once and sent control input back to it. Since the delays and packet loss cases are the same as we discussed in previous subsections, the gain design of the controller is the same with the one shown by (7.2). The real-time system response has been shown in Fig.8.6. The result showed that the system can be stochastically stabilized in 3 second due to the proper design of the controller.

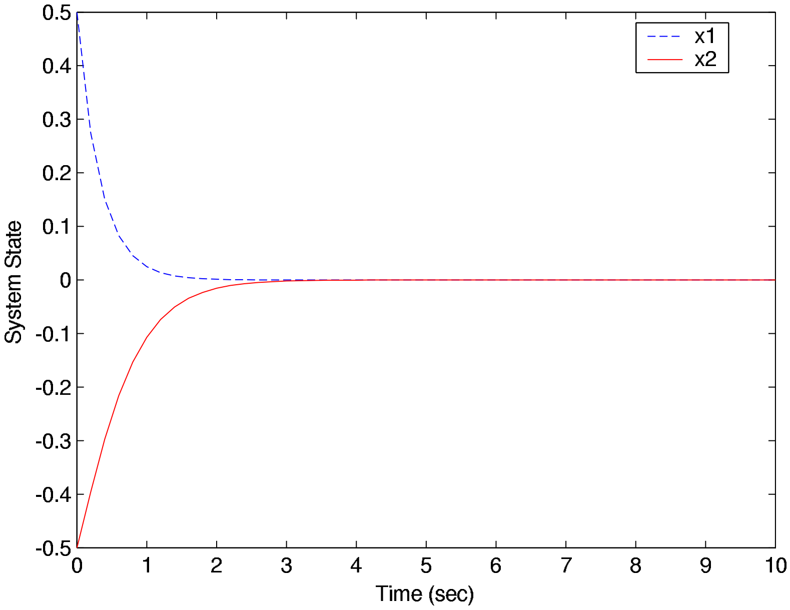


Figure 8.6. The state response of real-time networked control system

8.2 Implementation of Consensus Formation Control to Pioneer-3 AT and DX Mobile Robots

In this section, the proposed distributed formation control strategy is applied to a multiple mobile robot experimental platform. The main work for this section is divided into three part: first the hardware experimental set-up will be introduced; next the software regarding the control operation system of $P3$ mobile robots will be introduced. The experimental results will be discussed and shown.

8.2.1 Experimental Set-up

Experimental tests are implemented in the Advanced Control and Mechatronics Laboratory at Dalhousie University. The multiple mobile robot test platform consists of one Pioneer $3 - DX$ and one Pioneer $3 - AT$ as shown in Fig.8.7. The two robots are connected with two laptops through serial ports and the two laptops can communicate with each other via wireless network with TCP/IP protocols. Each robot has an encoder for their position and orientation information. Laptops are used for calculating control input values and connecting with $P3$ mobile robot.

In the experimental tests, a group of two $P3$ mobile robots is required to maintain the desired group formation according to its desired trajectory. The kinematic model for each robot has been shown in (6.1). This model has been linearized by setting a hand position for each robot and the simplified model has been represented by (6.6). After the group starts moving, laptops attached on the robots will keep performing the following tasks once per second: 1) reading position and orientation data from the robots; 2) exchanging those data with each other through the wireless network with

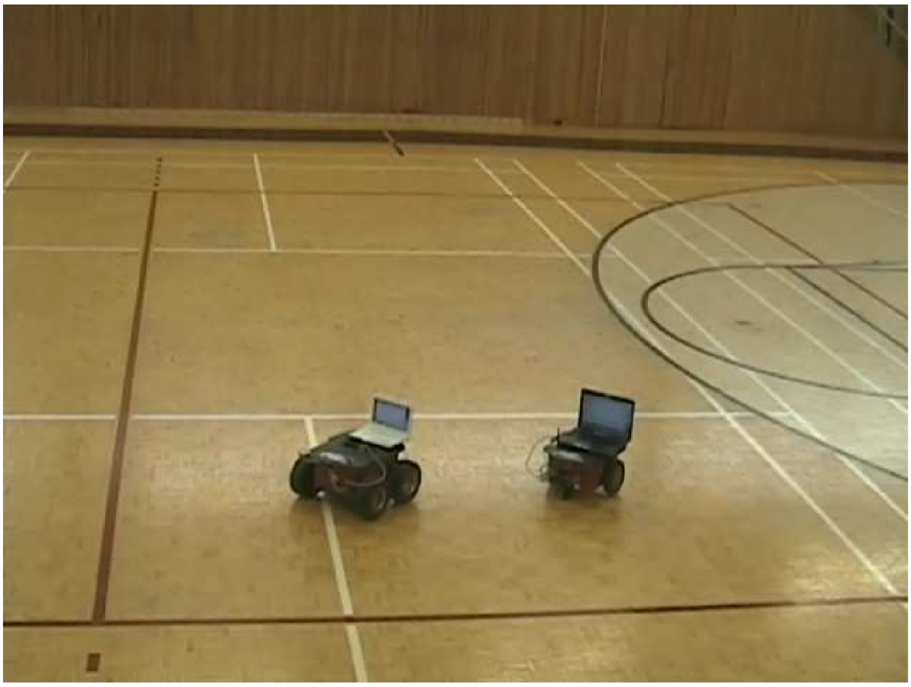


Figure 8.7. Pioneer 3 mobile robot test platform framework

1 second communication delay; 3) calculating the control input values according to the distributed consensus formation control algorithm (6.8) and then calculating the desired linear and orientation speed for the robots according to (6.5); 4) converting the desired linear and orientation speed of the robots into the desired rotating speed of the left and right wheel of the robots respectively based on the following equation:

$$\begin{aligned} \tilde{R} &= \frac{v}{\omega} \\ V_L &= \omega(\tilde{R} - 0.5l) \\ V_R &= \omega(\tilde{R} + 0.5l), \end{aligned} \tag{8.5}$$

where V_L and V_R are the left and right wheel rotating speed of the robot, $l = 33cm$ is the length between left and right wheel of Pioneer $3 - DX$ and $l = 38cm$ is that of Pioneer $3 - AT$; 5) sending signals to robots to set the rotating speeds as the desired values.

8.2.2 Introduction on Software

In the experimental tests, software programming is another important issue. All the commands and calculations processed by the controller (laptop) attached on each robot are programmed in C++ language. For simplifying the programming task, Advanced Robot Interface for Applications (ARIA) has been used in this work.

ARIA is a programming library for C++ programmers who want to access their Mobile Robot platform and accessories at either a high or low level. Written in the C++ language, ARIA is client-side software for easy, high-performance access to and management of the robot, as well as to the many accessory robot sensors and effectors. ARIA includes many useful utilities for general robot programming and cross-platform (Linux and Windows) programming as well.

ARIA can dynamically control the robot's velocity, heading, relative heading, and other motion parameters either through simple low-level commands or through its high-level Actions infrastructure. In this work ARIA also receives odometric position estimates, sonar readings, and all other current operating data sent by the robot platform. In this work, functions defined in ARIA have been invoked to read position data and set robot's velocity and orientations. To do so can save time in programming.

Since the two laptops are required to communicate with each other via the wireless network with TCP/IP topology, a library named ArSocket in ARIA has been used for this issue. With the help of ArSocket, we can directly create a server with one laptop and create a client with another. At the beginning of the group movement, the server will open a server port for the client and wait for the client to connect. The client will connect to the server by pursuing the server's IP address. When the server receive the client's call, it will send a call back to the client and start action. Then the client will start action as soon as receives the call from the server. The server and client will act synchronously in this way.

8.2.3 Experimental Results

This section shows the experimental results. In this test, two Pioneer 3 robots are required to maintain a fixed group formation during the movement with communication delay $\tau = 1s$. So the consensus based controller gain $k_i = 1.5675$ and the adjacency matrix A_{ij} are the same as results described in Section 7.3.1 for long delay case ($\tau = 1$ second).

The two robots in the team start moving at initial position of $(0, 0)$ for robot 1 ($P3 - AT$) and $(1050, 1050)$ for robot 2 ($P3 - DX$) with a unit in mm. As described in Section.6.2, the hand position for each robot has a distance L to the center point of the robot. In the experiment, $L = 30cm$ for both robots, so the initial hand positions of robot 1 and robot 2 are $(300, 300)$ and $(1350, 1350)$ respectively. The unit is mm. In the first test, the "hands" of the robot group are required to track a straight line as

$$\begin{aligned} x_d &= 60t + 500 \\ y_d &= 60t + 500, \end{aligned} \tag{8.6}$$

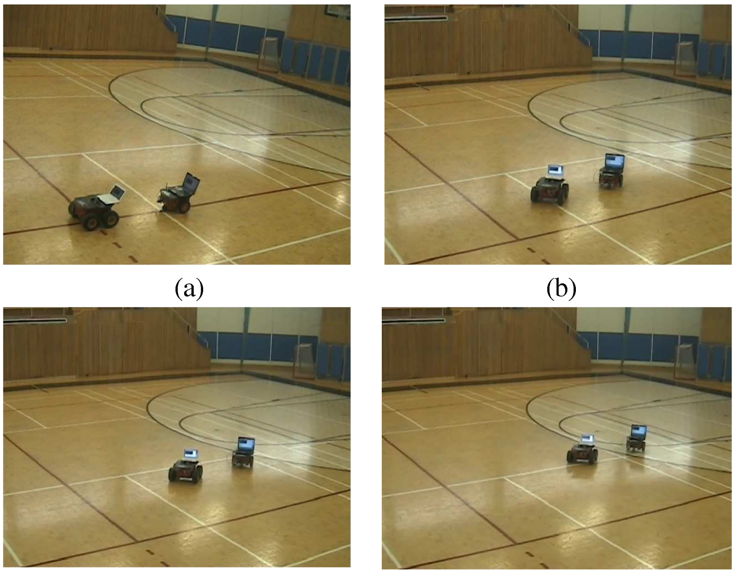
where t is the time starting from 0 second. The trajectory described by Eq.(8.6) is according to the coordinate system C_i of the ith robot itself. The group movement trajectory is shown in Fig.8.8.

In order to evaluate the group performance, the group movement trajectory has been recorded by the laptops and analyzed using MATLAB. The group trajectory is drawn in Fig.8.9. Since the position data has been recorded once per second, the time period between each location point in the picture is 1 second. As shown in the picture, $P3-AT$ and $P3-DX$ start moving from initial positions together at 0 seconds. The group gets close to the virtual leader's trajectory or the desired trajectory in the first 8 second of the movement. After 8 seconds, the group moves on the desired trajectory. During this process the desired group formation has been kept very well. Fig.8.10 shows that the consensus tracking errors for the virtual center position converge to

0, the lowest

perty of the

tracking erro



(a) (b) (c) (d)

Figure 8.8. Experimental result of group movement tracking straight line: (a) 0 second; (b) 20 second; (c) 40 second; (d) 60 second

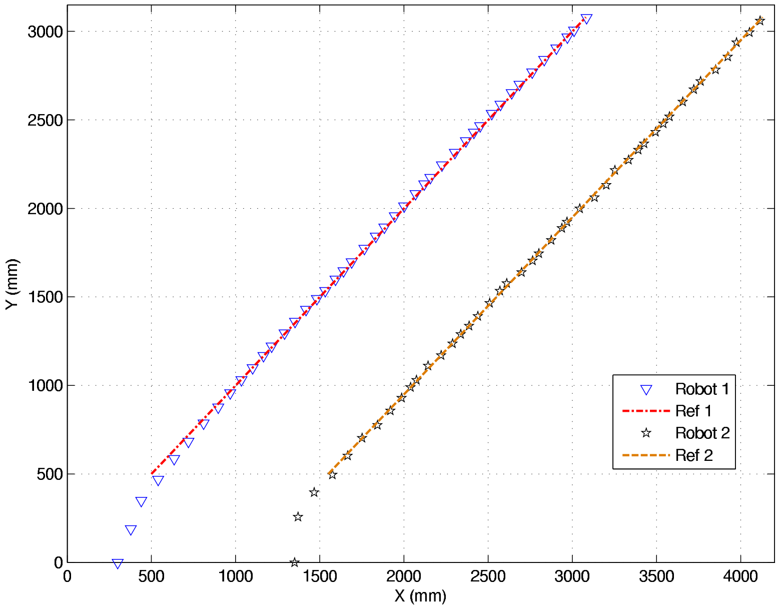


Figure 8.9. Group movement tracking straight line

For the second test, the group is required to track a curve as

$$\begin{aligned} x_d &= 20t + 500 \\ y_d &= 2t^2 + 500, \end{aligned} \tag{8.7}$$

where t is the time starting from 0 second, the unit of x_d and y_d is mm, and (8.7) is according to the coordinate system C_i of the ith robot itself. The team movement with fixed group formation is shown in Fig.8.11.

For this case the data has also been recorded and analyzed. Fig.8.12 shows the group movement trajectory and virtual leaders' trajectory. As shown in the picture, the group trajectory of each robot undulates a little bit in the first 8 second transition time. After 8 seconds, the group moves on the desired trajectory and undulates a little bit more than the previous test (the straight line case). Drawing the consensus tracking errors for virtual center positions in Fig.8.13 shows that the tracking error

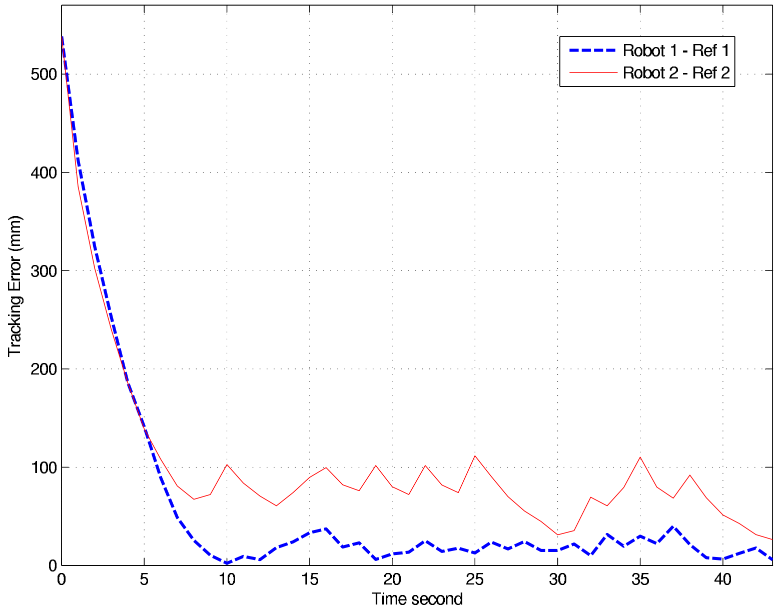


Figure 8.10. Consensus tracking errors for the virtual position: straight line case

of each robot converges in the first 20 second and the minimum value is below 2 cm. After 20 seconds it increases between 2 cm and 3 cm. This is because the motor speed of each robot, which is calculated and set by the controller once per second, is held for each 1 second during the movement. So the system is not ideal time continuous. For the curve tracking case, the virtual leader of each robot moves with an acceleration in Y axis direction which is equal to 4 mm/s^2. After the group starts moving, the virtual leader's velocity increases greatly. Then it is difficult for each robot to react and match the velocity and motion of its virtual leader. This hardware limitation causes the consensus tracking errors' performance as shown in Fig.8.13. If the group moves for over 200 seconds, then the speed of the virtual leader along Y axis becomes more than 0.8 meters per second which is the maximum speed of $P3-AT$. It is the main difference compared with the previous test (straight line case).

Due to the proper design of the control gain k_i, the mobile robot group can track

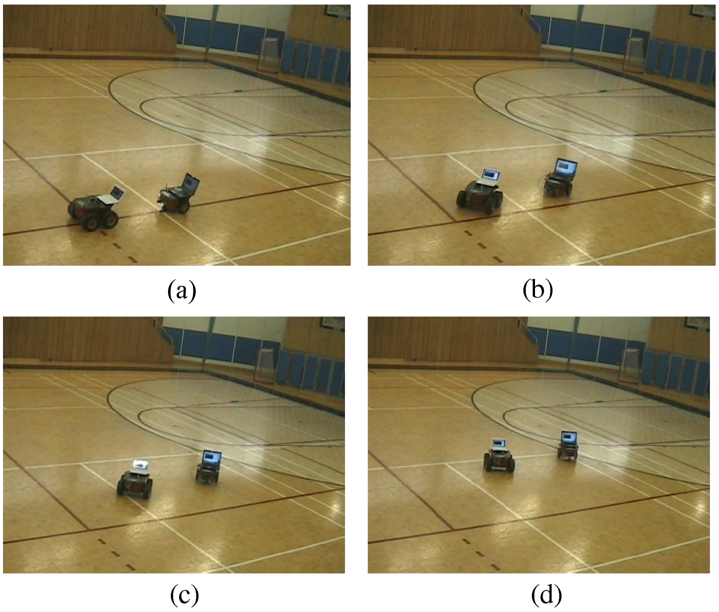


Figure 8.11. Experimental result of group movement tracking curve:(a) 0 second; (b) 7 second; (c) 20 second; (d) 35 second

the virtual leader's trajectory and maintain the desired group formation.

8.3 Summary

In this section, the experimental results of the work described in Chapter 4 and Chapter 6 have been shown and discussed. The network delay and packet loss measurement system has been introduced. The network control system has been established, the results show the feasibility of the proposed control strategy. The multiple $P3$ mobile robot experimental platform has been set up. The consensus formation control strategy described in Chapter 6 has been applied to the platform. The results show the good group performance due to the feasibility of the proposed approach.

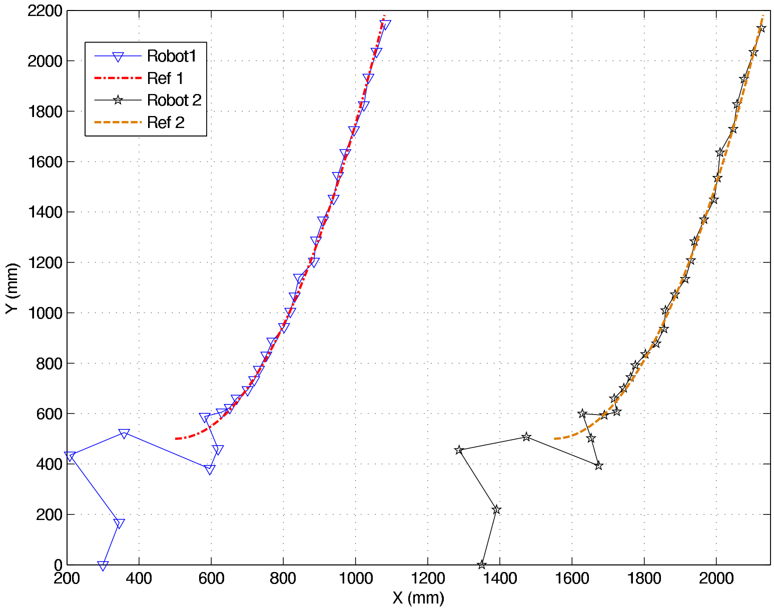


Figure 8.12. Group movement tracking curve

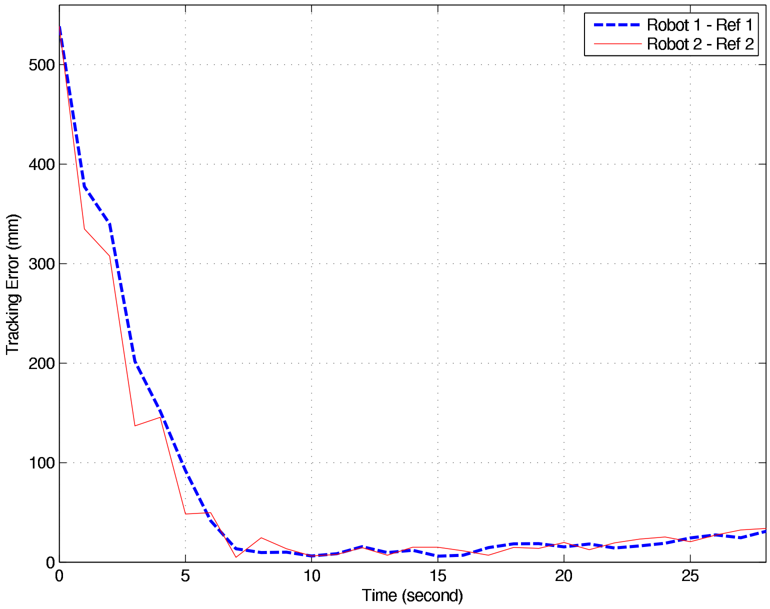


Figure 8.13. Consensus tracking errors for the virtual position: curve case

Chapter 9

Conclusions

9.1 Conclusions

In this thesis, the stabilizing problem of Networked Control Systems (NCSs) with packet losses and bounded time varying delays have been investigated. The packet loss has been modelled as a random process. The control objective is set to design a sampled-data stabilizing controller via communication channels with random packet losses and bounded time varying delays.

A real-time network induced delay and packet loss measurement system was built to study the real network characteristics. With the experimental measurement, the characters of time varying delays and packet losses in the real network have been studied and applied into the packet loss modelling process.

A novel packet loss dependent Lyapunov functional candidate has been constructed to solve the problem on the stochastic stabilization of the NCSs with the effect of Markovian packet loss only. Stability conditions are derived via Lyapunov approach and the corresponding stabilizing sampled-data controller design techniques are also given based on the conditions. Then the effects of both Markovian packet loss and time varying delays occurring in both channels are considered. Another novel packet loss dependent Lyapunov-Krasovskii functional (LFK) has been constructed to solve the stochastic stabilization problem for NCSs with both random packet loss and time varying delays. The delays and packet losses that were measured during the network experiments are firstly replayed for the simulation. A real-time networked control system has been built based on MATLAB application to test the stabilizing ability of the controller. Both simulation and experimental results show the effectiveness and feasibility of the proposed approach,

After dealing with the stabilization of single network control system, the multi-agent system has been investigated. The consensus-based design scheme has been applied to the formation control of multiple wheeled mobile robot group with a virtual leader. The distributed formation control architecture has been defined. The consensus tracking algorithm is applied to develop the proposed consensus based formation control strategy. A novel delay-dependent multiple Lyapunov functional candidate has been constructed to investigate the convergence of the tracking error. The networked induced delays also have been considered in the proposed approach. In simulation works, the performance of the robot group with and without group communication have been compared with each other. The comparing results show the importance of the communication in formation maintenance of the robot group. The theoretic design also has been supported by the simulation results. The Pioneer 3 mobile robots have been used as an experimental platform for the implementation of the proposed distributed formation control architecture. The experimental results show the effectiveness and feasibility of the proposed approach.

9.2 Future Work

Some interesting work extensions are listed as follows:

1) Consider both packet loss and delays in the communication channels of multi-agent system. The packet loss can be modeled as a random process as what have been done in the work of stochastic stabilization of a single network control system. Then the stochastic stabilization of multi-agent systems can be tackled by designing a novel consensus-based distributed control strategy. The results can be compared with that of the proposed approach presented in this thesis.

2) Consider the consensus-based stabilization problem of multi-agent systems with switching communication topology. There are two ways to model the switching communication topology. One is assuming there are n fixed topologies, where n is a scale. The communication topology will switch from one to another at each time. The probability of each switch is known. Then the switching process can be modelled as a random process by using Markov chain. Another is to choose any two members in the group as investigating objective. Then there are only two statuses for the communication between them: “yes” or “no”. The probability of “yes” or “no” is assumed to be known. The communication topology can be modelled as a deterministic process. Compared with the previous way, deterministic switching topology is more closed to real case and more difficult to tackle.

3) Consider the consensus-based stabilization problem of multi-agent systems with limited information exchange. In this case, each member in the group is supposed to communicate only with its local neighbors. It is the real situation of the behavior of

a flock of animals, such as fish, ant and birds. How to define the local neighbor is a key problem for this case. There is one way that the local neighbor can be defined by distance. However, the distance keeps changing during the group movement which means the communication topology is switching. The limited information exchange and the switching communication topology make it very difficult deal with the consensus problem of the systems. This situation also make it become a challenging and interesting research area.

Appendix A: Brief Operational Manual for Experiment in Section 8.2

1) Install Serial-to-USB connector driver to the laptop.

2) Run Microsoft Visual C++ 2008 Express Edition on the laptop. Create a project named “simple connection”. Copy the example code named as “simple connection” from ARIA manual to the project, compile the code.

3) Connect the Pioneer 3 mobile robot with the laptop using the Serial-to-USB connector, run the “simple connection” project. If the robot reacts to the project (clicking) which means the driver has been installed successfully, then the robot is ready for the experiment.

4) For another Pioneer 3 robot, repeat the steps $1 - 3$, make sure each robot is connected well with the laptop.

5) Connect each laptop to the wireless router via the wireless network. Use “ipconfig” command to acquire the ip address of each laptop. Use “ping” command to ping from one to another to make sure both are connected via the router.

6) Run the project on the laptop for each robot. The experiment starts.

Appendix B: Author's Publication List

Publications in International Journals

1. L. Sheng and Y.J. Pan, Stochastic stabilization of sampled-data networked control systems, *International Journal of Innovative Computing, Information and Control*, Vol. 6, No. 4, pp. 1949-1972, 2010.

2. L. Sheng and Y.J. Pan, Distributed control for consensus of networked multi-agent robotic systems with time delays, *International Journal of Information and Systems Sciences*, Vol. 5, No.2, pp. 161-178, 2009.

3. Y.J. Pan, H. J. Marquez, T. Chen and L. Sheng, Effects of network communications on a class of learning controlled nonlinear systems, *International Journal of Systems Science*, Vol. 40, No. 7, pp. 757-767, 2009.

4. Y.J. Pan and L. Sheng, Predictor-based repetitive learning control for a class of remote control nonlinear systems, *International Journal of Robust and Nonlinear Control*, Vol. 17, pp.1455-1473, 2007.

5. L. Sheng and Y.J.Pan, Distributed consensus-based formation control approach for multi-robotic vehicle systems, submitted to *IEEE Transactions on Mechatronics*,2010.

Publications in International Conferences

1. L. Sheng and Y.J. Pan, Stochastic analysis and stabilization of remote control systems, In Proceedings of the 2009 IEEE American Control Conference, June 2009, St. Louis, Missouri, USA, pp. 2984-2989.

2. Y.J. Pan and L. Sheng, A new predictor-based repetitive learning control approach for a class of remote control nonlinear systems, In Proceedings of the 2007

IEEE American Control Conference, July 2007, New York, USA, pp. 1281-1286.

3. L. Sheng and Y.J. Pan, Distributed consensus formation control of multiple mobile robots, submitted to the 2011 IEEE International Conference on Robotics and Automation, May 2011, Shanghai, China.

Bibliography

[1] S. Zampieri. Trends in networked control systems. In *Proceedings of the 17th IFAC World Congress*, COEX, Korea, pages 151–159, July 2008.

[2] K. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *IEEE Transactions on Automatic Control*, 49:237–252, 1998.

[3] S. H. Low and D. E. Lapsley. Optimization flow control i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

[4] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong. Analysis and design of controllers for aqm routers supporting tcp flows. *IEEE Transactions on Automatic Control*, 47:945–959, 2002.

[5] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8:556–567, 2000.

[6] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low. Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Transactions on Networking*, 13:43–56, 2005.

[7] I. Lestas and G Vinnicombe. Scalable decentralized robust stability certificates for networks of interconnected heterogeneous dynamical systems. *IEEE Transactions on Automatic Control*, 51:1613–1625, 2006.

[8] I. Lestas and G Vinnicombe. Scalable robust stability for nonsymmetric heterogeneous networks. *IEEE Transactions on Automatic Control*, 43:714–723, 2007.

[9] L. Ying, G. E. Dullerud, and R. Srikant. Global stability of internet congestion controllers with heterogeneous delays. *IEEE/ACM Transactions on Networking*, 14:579–590, 2006.

[10] Y. J. Pan and L. Sheng. Predictor-based repetitive learning control for a class of remote control nonlinear systems. *International Journal of Robust and Nonlinear Control*, 17:1455–1473, 2007.

[11] H. Yamaguchi, T. Arai, and B. Gerardo. A distributed control scheme for multiple robotic vehicles to make group formations. *Robotics and Autonomous Systems*, 36:126–147, 2001.

[12] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. In *Proceedings of the IEEE*, volume 95, pages 138–162, Jan 2007.

[13] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1):57–64, 1998.

[14] B. Azimi-Sadjadi. Stability of networked control systems in the presence of packet losses. In *Proceedings of the 42th IEEE Conference on Decision and Control*, Maui, HI, pages 676–681, December 2003.

[15] M. Yu, L. Wang, T. Chu, and F. Hao. An lmi approach to networked control systems with data packet dropout and transmission delays. *International Journal of Hybrid Systems*, 3:291–303, 2003.

[16] M. Wu, Y. He, J. She, and G. Liu. Delay-dependent criteria for robust stability of time-varying delay systems. *Automatica*, 40:1435–1439, 2004.

[17] M. Yu, L. Wang, T. Chu, and G. Xie. Stabilization of networked control systems with data packet dropout and network delays via switching system approach. In *Proceedings of the 43th IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, pages 3539–3544, December 2004.

[18] P. Seiler and R. Sengupta. An H_∞ approach to networked control. *IEEE Transactions on Automatic Control*, 50(3):356–364, 2005.

[19] Y. J. Pan, H. J. Marquez, and T. Chen. Stabilization of remote control systems with unknown time varying delays by LMI techniques. *International Journal of Control*, 79(7):752–763, 2006.

[20] E. Fridman. Stability of systems with uncertain delays: A new complete Lyapunov-Krasovskii functional. *IEEE Transactions on Automatic Control*, 51(5):885–890, 2006.

[21] J. Xiong and J. Lam. Stabilization of linear systems over networks with bounded packet loss. *Automatica*, 41:889–992, 2006.

[22] P. Shi, Y. Xia, G. Liu, and D. Rees. On designing of sliding mode control for stochastic jump systems. *IEEE Transactions on Automatic Control*, 51(1):97–103, 2006.

[23] E. K. Boukas and N. F. Al-Muthairi. Delay-dependent stabilization of singular linear systems with delays. *International Journal of Innovative Computing,Information and Control,*, 2(2):283–291, 2006.

[24] B. Chen, J. Lam, and S. Xu. Memory state feedback guaranteed cost control for neutral delay systems. *International Journal of Innovative Computing,Information and Control,*, 2(2):293–303, 2006.

[25] C. Kao and A. Rantzer. Stability analysis of systems with uncertain time-varying delays. *Automatica*, 43:959–970, 2007.

[26] R. Touri and C. N. Hadjicostis. Stabilisation with feedback control utilising packet-dropping network links. *IET Control Theory & Applications*, 1(1):334–342, 2007.

[27] M. S. Mahmoud, Y. Shi, and N. Nounou. Resilient observer-based control of uncertain time-delay systems. *International Journal of Innovative Computing,Information and Control,*, 3(2):407–418, 2007.

[28] J. P. Rechard. Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39(10):1667–1694, 2003.

[29] J. M. Azorin, O. Reinoso, J. M. Sabater, and R. P. Neco. Dynamic analysis for a teleoperation system with time delay. In *Proceedings of IEEE Conference on Control Applications*, Maui, HI, pages 1170–1175,, June 2003.

[30] A. Fattouh and O. Sename. H infinity based impedance control of teleoperation systems with time delay. In *Proceedings of 4th IFAC Workshop on Time Delay Systems*, INRIA, France, July 2003.

[31] C. E. Garcia, B. Morales, R. Carelli, and J. Postigo. Stability analysis for a teleoperation system with time delay and force feedback. In *Proceedings of the 39th Conference on Decision and Control*, December 2000.

[32] Y. J. Pan, C. Canudas-de Wit, and O. Sename. A new predictive approach for bilateral teleoperation with applications to drive-by-wire systems. *IEEE Transactions on Robotics*, 22(6):1146–1161, 2006.

[33] E. Witrant, C. Wit, and D. Georges. Remote output stabilization under two channels time-varying delays. In *Proceedings of the 4th IFAC Workshop on Time Delay Systems*, INRIA, France, July 2003.

[34] W. Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. *IEEE Control Systems Magzines*, 21(1):84–99, 2001.

[35] E. Fridman and U. Shaked. Delay dependent stability and h infinity control: constant and time-varying delays. *International Journal of Control*, 76(1):48–96, 2003.

[36] W. Huang. Generalization of lyapunov's theorem in a linear delay system. *Journal of Mathematical Analysis and Applications,*, 142:83–94, 1989.

[37] V. Kharitonov and J. P. Richard. Stability of some linear systems with delays. *IEEE Transactions on Automatic Control*, 44(5):984–989, 1999.

[38] V. Kharitonov and S. Niculescu. On the stability of linear systems with uncertain delay. *IEEE Transactions on Automatic Control*, 48(1):127–132, 2002.

[39] H. Gao, T. Chen, and J. Lam. A new delay system approach to network-based control. *Automatica*, 44:39–52, 2008.

[40] J. F. Gao, H. P. Pan, and J. X. Fu. A new delay-dependent absolute stability criterion for lurie systems with time-varying delay. *Acta Automatica Sinica,*, 36(6):845–850, 2010.

[41] A. Papachristodoulou, M. Peet, and S. Lall. Constructing lyapunov-krasovskii functionals for linear time delay systems. In *Proceedings of the American Control Conference*, Minneapolis, Minnesota, USA, pages 2845–2850, Jun 2005.

[42] Y. Ariba and F. Gouaisbaut. Construction of lyapunov-krasovskii functional for time-varying delay systems. In *Proceedings of the 47th Conference on Decision and Control*, Cancun, Mexico, pages 3995–4000, Jan 2008.

[43] P. G. Park. A delay-dependent stability criterion for systems with uncertain time-invariant delays. *IEEE Transactions on Automatic Control*, 44(4):876–877, 1999.

[44] Y. S. Moon, P. Park, W. H. Kwon, and Y. S. Lee. Delay-dependent robust stabilization of uncertain state-delayed systems. *International Journal of Control*, 74:1447–1455, 2001.

[45] E. Fridman and U. Shaked. A descriptor system approach to h infinity control of linear time-delay systems. *IEEE Transactions on Automatic Control*, 47(2):253–270, 2002.

[46] Y. S. Lee, Y. S.and Moon, W. H. Kwon, and P. G. Park. Delay dependent robust h infinity control for uncertain systems with a state delay. *Automatica*, 40:65–72, 2004.

[47] Peng C. and Y. C. Tian. Delay-dependent robust h infinity control for uncertain systems with time-varying delay. *Information Sciences*, 178(8):3187–3197, 2009.

[48] Y. He, M. Wu, and G. P. Liu. Parameter-dependent lyapunov functional for stability of time-varying systems with polytopic-type uncertainties. *IEEE Transactions on Automatic Control*, 49:828–832, 2004.

[49] Y. He, Q. G. Wang, L. H. Xie, and C. Lin. Further improvment of free-weighting matrices technique for systems with time-varying delay. *IEEE Transactions on Automatic Control*, 52:293–299, 2007.

[50] M. Wu, Y. He, J. H. She, and G. P. Liu. Delay-dependent criteria for robust stability of time-varying delay systems. *Automatica*, 40:1435–1439, 2004.

[51] D. Yue, Q. L. Han, and J. Lam. Network-based robust H_∞ control of systems with uncertainty. *Automatica*, 41:999–1007, 2005.

[52] W. B. Dunbar and R. M. Murray. Model predictive control of coordinated multi-vehicle formations. In *In Proceedings of the Conference on Decision and Control*, Las Vegas, Nevada, USA, 2002. pp. 4631-4636.

[53] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42:549–558, 2006.

[54] H. Yamaguchi. A cooperative hunting behavior by mobile robot troops. *International Journal of Robotics Research*, 18(9):931–940, 1999.

[55] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb. Autonomous oceanographic sampling networks. *Oceanography,*, 6:86–94, 1993.

[56] T. R. Smith, H. Hmann, and N. E. Leonard. Orientation control of multiple underwater vehicles. In *In Proceedings of 40th Conference on Decision and Control*, Orlando, Florida, USA, pages 4598–4603, 2001.

[57] L. E. Buzogany, M. Pachter, and J. J. Azzo. Automated control of aircraft in formation flight. In *Proceedings of AIAA Conference Guidance, Navigation and Control*, Monterey, California, USA, pages 1349–1370, 2001.

[58] J. D. Wolfe, D. F. Chichka, and J. L. Speyer. Decetntralized controllers for unmanned aerial vehicle formation flight. In *Proceedings of AIAA Conference Guidance, Navigation and Control*, San Diego, California, USA, pages 3833–3851, 1996.

[59] E. Shaw. Fish in schools. *Natural History,*, 84(8):40–45, 1975.

[60] R. Olfati-Sabet, A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE,*, 95(1):215–233, 2007.

[61] B. Mohar. The laplacian spectrum of graphs. In *In Graph Theory, Combinatorics and Applications*, pages 871–898, 1991.

[62] R. Merris. Laplacian matrices of a graph: A survey. *Linear Algebra its Applications,*, 197:143–176, 1994.

[63] A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.

[64] R. Olfati-Sabet and R. M. Murray. Consensus problems in networks of agents with switching topology and time-dealys. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[65] Olfati-Sabet. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[66] P. K. C. Wang. Navigation strategies for multiple autonomous mobile robots moving in formation. *Journal of Robotic Systems*, 8(2):177–195, 1991.

[67] P. K. C. Wang and F. Y. Hadaegh. Coordination and control of multiple microspacecraft moving in formation. *Journal of the Astronautical Sciences*, 44(3):315–355, 1996.

[68] T. Sugar and V. Kumar. Decentralized control of cooperating mobile manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pages 2916–2921, May 1998.

[69] J. P. Desai, J. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pages 2864–2869, May 1998.

[70] H. Yamaguchi and J. W. Burdick. Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations. In *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pages 3573–3580, May 1998.

[71] C. Belta and V. Kumar. Abstraction and control for groups of robots. *IEEE Transactions on Robotics and Automation*, 20(5):865–875, 2004.

[72] R. W. Beard and D. Lee. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, 2001.

[73] M. Egerstedt, X. Hu, and A. Stotsky. Control of mobile platforms using a virtual vehicle approach. *IEEE Transactions on Automatic Control*, 46(11):1777–1782, 2001.

[74] P. Ogren, M. Egerstedt, and X. Hu. A control lyapunov function approach to multiagent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, 2002.

[75] P. Seiler and R. Sengupta. Analysis of communication lossers in vehicle control problems. In *Proceedings of the American Control Conference*, Arlington, VA, USA, pages 1491–1496, Jun 2001.

[76] Y. V. Mikheev, V. A. Sobolev, and E. M. Fridman. Asymptotic analysis of digital control systems. *Automation and Remote Control,*, 49(9):1175–1180, 1988.

[77] E. Fridman, S. Alexandre, and J. Richard. Robust sampled-data stabilization of linear systems: an input delay approach. *Automatica*, 40:1441–1446, 2004.

[78] D. Yanakiev and I. Kanellakopoulos. A simplified framework for string stability analysis in automated highway systems. In *Proceedings of 13th IFAC World Conference*,San Fransisco, USA, pages 177–182, 1996.