CONSENSUS CONTROL OF MULTIPLE-QUADCOPTER SYSTEMS UNDER COMMUNICATION DELAYS

by

Zipeng Huang

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

at

Dalhousie University
Halifax, Nova Scotia
December 2017

Table of Contents

List of Tables

List of Figures

Abstract

Multiple-quadcopter systems have various civilian and military applications, such as forest fire monitoring and load transportation. However, since multiple-quadcopter systems are networked control systems (NCSs), they suffer from network-induced constraints, such as time delay and packet loss. Consensus, which is a basic coordination problem, is often desired for the group in achieving tasks. The objective of this thesis is to develop novel distributed consensus algorithms for multiple-quadcopter systems over two types of communication delays: uniform constant delays and asynchronous time-varying delays. The quadcopter system is simplified into four decoupled subsystems such that it can be studied in a multi-agent system (MAS) scale. The interactions among quadcopters are modeled using algebraic graph theory. The consensus problem is then converted to a stability analysis problem by defining the consensus error dynamics. Sufficient conditions for stabilizing controller design are developed based on Lyapunov's method and linear matrix inequality (LMI) techniques for both cases. Finally, extensive MATLAB simulations are carried out for both cases to verify the proposed algorithms. Discussions are given regarding the feasibility and effectiveness of the proposed controllers under various conditions.

List of Abbreviations and Symbols Used

$A > 0$	positive definite matrix A
A^{-1}	inverse of matrix A
A^T	transpose of matrix A
H_∞	H-infinity control
I_n	$n \times n$ identity matrix
$*$	transpose of the off-diagonal block in a block matrix
$\in$	belongs to
∞	infinity
$\mathbb{R}$	set of real numbers
$\mathbb{R}^{n\times n}$	set of $n \times n$ real matrices
$\mathbb{R}^n$	set of $n \times 1$ real vectors
$\mathcal{A}$	adjacency matrix
$\mathcal{D}$	in-degree matrix
$\mathcal{E}$	edge set of a graph
$\mathcal{G}$	graph
$\mathcal{L}$	Laplacian matrix
$\mathcal{V}$	node set of a graph
$\otimes$	kronecker product
$\rightarrow$	tends to
$\sum$	summation
τ	constant communication time delay
$\tau_{\delta_{ij}}$	time-varying delay in communication channel from agent j to i
τ_{max}	maximum allowable constant delay
d	bound of time-varying portion of the delay
e_c	consensus error
e_i	system error for i^{th} agent
l_2	l_2-norm of a vector

t_c	consensus time
DOF	degrees of freedom
GPS	global positioning system
GS	ground station
LMI	linear matrix inequality
LPV	linear parameter-varying
LQR	linear-quadratic regulator
MAS	multi-agent system
NCS	networked control system
PD	proportional-derivative
SISO	single-input-single-output
UAV	unmaned aerial vehicle
VTOL	vertical takeoff and landing

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Ya-Jun Pan of the Mechanical Engineering Department for her continuous support and guidance. Besides my supervisor, I would like to thank Dr. Robert Bauer and Dr. Mohamed E. El-Hawary for their role on my supervisory committee, and the Advanced Controls and Mechatronics research group for their suggestions and comments over the years. Last but not the least, I would like to thank my parents for supporting me spiritually throughout writing this thesis and my life in general.

Chapter 1

Introduction

The following sections outline the background for the research work, including a research motivation, overviews in applications and communication constraints, and research contributions.

1.1 Research Motivation

Multi-agent systems (MASs) are computerized systems composed of multiple intelligent agents that can interact with each other through exchanging information. The agents are decentralized which means there is no single controlling agent in the system, and autonomous which means they are at least partially independent. In addition, each agent only has local views of the whole system [4]. Using MAS to achieve a common task is usually more efficient and with more operational capability than a single agent system especially for tasks which are difficult or impossible for a single agent to complete such as combat, surveillance, mapping and underwater mine hunting.

Coordinations between the agents are mandatory for the MAS in achieving tasks as a group. Consensus control, which has received considerable attention from researchers in recent years, is a basic coordination problem in MASs, that is to develop a consensus algorithm for each agent only based on its local information such that the group of agents can reach an agreement on certain quantities of interest. Many other coordination algorithms, such as axial alignment and formation control, can be designed based on the consensus algorithm. In addition, information exchange between agents in MASs is mostly and popularly through shared wireless network by the virtue of its flexible architectures and generally reduced installation and maintenance costs. However, wireless networks are not always as reliable as hardwired ones due to connection strength, bandwidth constraints, which can cause packet delays and losses. Motivated by these discussions, this thesis investigates the robust consensus control design against network-induced delays for multiple-quadcopter systems.

1.2 Applications Overview

A quadcopter is a special type of vertical takeoff and landing (VTOL) unmaned aerial vehicle (UAV). It is mechanically simple, highly maneuverable and it can carry large payload compared with conventional helicopters. Due to the above advantages, multiple-quadcopter systems can be applied in accomplishing civilian and military tasks, such as forest fire monitoring, load transportation and surveillance.

1.2.1 Forest Fire Monitoring

Wildfire consumed approximately 27 million acres of land in North America during 2005-2007 [5]. In many cases, lots of residents have been displaced thus inducing great financial losses. Besides, wildfires also pose great threats to human and wildlife health due to the smoke. Therefore, early detection of the forest fire is crucial in protecting forests and limiting the fire spread. A system of multiple quadcopters in leader-follower structure equipped with different types of sensors, which can cooperatively confirm and reduce the false fire alarm using the measurements from different sensors, is proposed for forest monitoring and fire detection in [1].

The forest monitoring and fire detection is achieved through three stages shown in

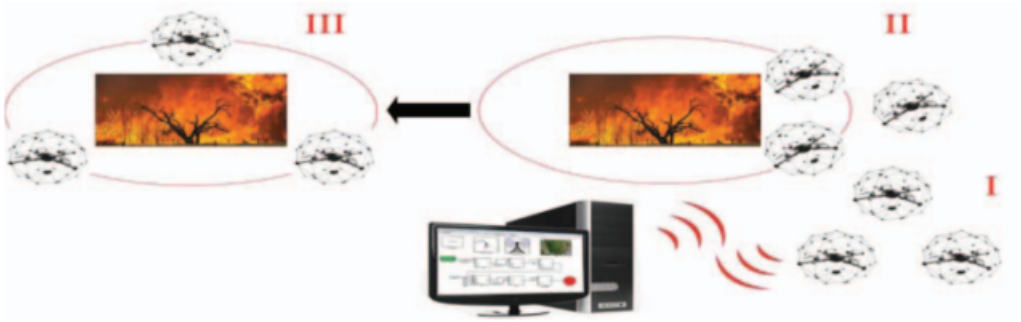

Figure 1.1: Forest monitoring and fire detection scenario [1]

Fig. 1.1. In the first stage, the system takes off and flying in formation while following a planned searching and coverage path. In the second stage, once a agent detects fire, it will send fire alarm to the ground station (GS) and the remaining agents. Sensory data will then be sent to the GS from all agents. Based on the information received, the GS will re-plan the reference trajectory and send this information to

the leader. The followers will then start tracking this new reference trajectory using their own local on-board controller. In the final stage, according to the new situation surrounding the fire spot, the system will reconfigure its formation to track the fire perimeter to confirm alarm and provide updated information about the fire.

1.2.2 Load Transportation

Load transportation using a quadcopter has various potential applications ranging from package delivery, automated construction, and emergency rescue missions due to its ability of vertical taking off and landing in rough or inaccessible areas, hovering capability and omni-directional maneuverability. Load transportation with multiple quadcopters is advantageous when the load is heavier than the maximum payload of a single quadcopter, or when additional redundancy is required for safety.

In [2], multiple quadcopters are controlled to cooperatively grasp and transport a

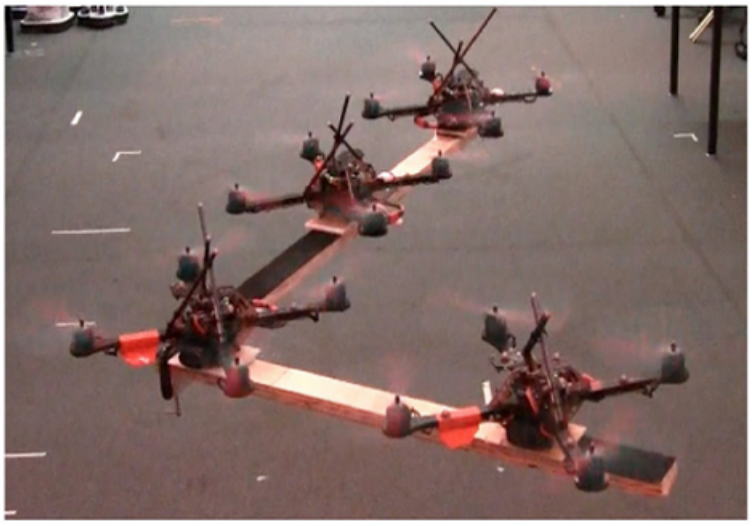

Figure 1.2: Payload transportation [2]

payload in three-dimensional space as shown in Fig. 1.2. A special griping mechanism which allows the quadcopter to attach to and release from the payload is developed. Similar research but with suspended cables for transporting loads are conducted in [6, 7].

1.3 Communication Constraints Review

The attribute of NCSs that distinguishes with other control systems is that information (reference input, plant output, control input, etc.) is exchanged using a shared digital communication network as shown in Fig. 1.3, among control system components (sensor, controller, actuator, etc.). In contrast to the traditional point-to-point centralized control systems, NCSs are more flexible, cheaper and easier to install and maintain. However, they also suffer from network-induced constraints, such as time delay, packet loss, time-varying transmission intervals, competition of multiple nodes accessing networks, and data quantization [8, 9]. The first two constraints will be emphasized in this section.

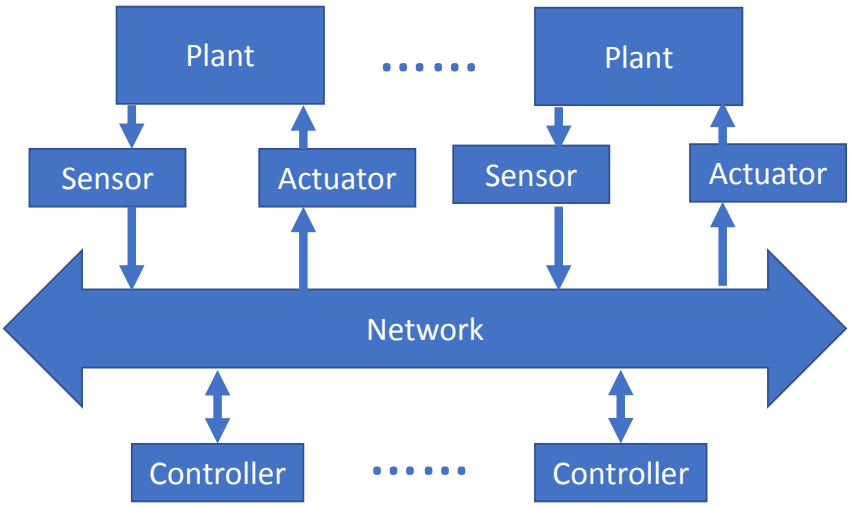

Figure 1.3: Typical NCS structure

1.3.1 Time Delay

It is well known from studies on traditional time-delayed systems that delays can deteriorate the system performance or even induce instability of the closed-loop system. The delays in NCSs are composed of three components: computational delay, network access delay, and transmission delay. The computational delay is a result of the finite processing speed of digital devices which are often negligible compared with the other two classes of delays that are commonly called network-induced delays. The network access delay is the time that a queued packet has to wait before being sent

out. The transmission delay is the amount of time required for the packet through the network medium.

The time delays in NCSs are usually dealt in robustness or adaptation frameworks. A typical approach in robustness framework is to treat an NCS as a traditional input-delay system as follows:

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ u(t) &= Kx(t - \tau_l(t)), \end{cases} \tag{1.1}$$

where K is state-feedback controller gain, $\tau_l(t)$ is a lumped time-varying delay in feedback and forward channels. In addition, $0 \leq \tau_l(t) \leq d$, where d is a known constant. Then the Lyapunov-based method can be implemented in deriving the conditions of system stability performance [10, 11]. In an adaptation framework, the NCSs are modeled as stochastic switched systems. In contrast to the static control gain design in robustness framework, the controller gains are actually switching depending on the size of delays [12].

1.3.2 Packet Loss

Data are transmitted as information packets in NCSs and they may be lost due to network traffic congestion and transmission error in physical network links especially in a wireless network. Besides, long propagation delay of a packet can also be treated as a packet loss since the outdated packets will generally be discarded by the receiver. Therefore, dealing with the packet loss in NCSs is critical.

Generally, the off-line and on-line frameworks are proposed for dealing the packet loss in NCSs. In the off-line framework [13], the stabilizing controller is designed based on the feasible solution of sufficient LMI formed for a given maximum consecutive packet losses m, despite any knowledge of real packet losses in the system. The controller is efficient as long as the number of consecutive packet losses in the system is less than m. In another off-line framework [14], the packet losses in both feedback and forward channels are modeled by the Bernoulli random process. The control design is then carried out by using the expectation of the stochastic variables. In an on-line framework, the control gains are calculated off-line, yet the control input to the plant is implemented on-line depending on the situation whether a packet is lost or received.

1.4 Consensus Control Review

Consensus behavior of multi-agent systems commonly exists in the real world. It has wide engineering applications such as clock synchronization, and multi-motor synchronization [15]. This section reviews existing works related to the consensus control design for MASs with or without communication constraints.

1.4.1 Consensus Control without Communication Constraints

The groundwork of the consensus problem for clusters of linear dynamic agents with fixed or switching topologies was presented in [16]. A high-order consensus algorithm for integrator-type systems was developed in [17], which generalizes the single-integrator and double-integrator consensus algorithms existing in the literature. Motivated by this work, a formation control algorithm for a group of quadcopters with fourth order linearized flight dynamics was designed in [18]. The full-order observer [19] and reduced-order observer [20] based output feedback consensus protocols for MASs with general linear dynamics and directed communication topology were also developed. Inspired by these output feedback algorithms, a new observer based consensus algorithm was proposed in [21]. The consensus problem for networks of fully-actuated Euler-Lagrange systems where all of the states are measurable was studied in [22]. The synchronization problem for a group of nonlinear oscillators was studied in [23] and the synchronization of linear parameter-varying (LPV) systems for both homogeneous and heterogeneous agents was investigated in [24].

1.4.2 Consensus Control with Communication Constraints

The communication constraints considered in consensus control design mainly are time-delays and packet losses. The packet loss has been modeled as Bernoulli random processes in [25] and Markovian switching processes in [26] . Most approaches used Lyapunov-based methods to ensure mean square stability due to the stochastic nature that packet loss imposed on the system. The research of delay effects in MASs is initiated in [16], where a delay bound that ensures the asymptotic convergence of the system was developed based on the frequency domain analysis for single-integrator systems under uniform communication delays. The result was extended to

single-integrator [27] and double-integrators systems [28] under non-uniform delays. In [29], the leader-follower consensus problem for double-integrators systems under synchronous time-varying delays is investigated using Lyapunov-based method. In general, the delay effect is analyzed using frequency domain method for the system with linear agent dynamics and constant uniform or nonuniform delays, and using Lyapunov's method for the system with time-varying delays. A review of some of the recent progresses and challenges in the consensus problem of MASs was given in [30].

1.5 Thesis Contribution

The main contribution of this thesis is the developments of novel consensus algorithms for multiple-quadcopter system under synchronous constant delays and asynchronous time-varying communication delays.

1. Specifically, the quadcopter dynamics is simplified such that it can be studied in a MAS scale.

2. The consensus problem is converted to a stability analysis problem by defining the consensus error dynamics.

3. Lyapunov-based methods along with LMI techniques are applied in forming the sufficient conditions for the static stabilizing control gain design for the error dynamics under constant and time-varying delays.

4. The stability conditions are decomposed into small conditions with the size of a single agent in order to reduce the computational complexity for a large group of quadcopters.

5. Extensive simulations are conducted, and discussions are given regarding the feasibility and effectiveness of the proposed controllers under various conditions.

1.6 Thesis Outline

The rest of this work is organized as follows. Chapter 2 describes the background theories used in this work. Chapter 3 presents the mathematical model used in this

work. The control objective is also described in this chapter. Chapter 4 gives a detailed description for the consensus algorithm design for the MAS under synchronous constant and asynchronous time-varying communication delays. Chapter 5 discusses the simulation results in the constant delay case regarding the effects of the communication topology, the number of agents, the constant delay and the arbitrary design parameters. Chapter 6 discusses the simulation results in the time-varying delay case. Chapter 7 summarizes the conclusion of this work and suggests areas for future research.

Chapter 2

Background Theory

This chapter presents the notations, preliminaries and theoretical concepts used in this thesis: 1)graph theory 2)Lyapunov-based controller design 3)matrix theory.

2.1 Multi-Agent Systems with Graph Theory

This section discusses the fundamental concepts of control of MASs modeled using the graph theory. This section is mainly based on [31, 32].

2.1.1 Graph Theory

In MASs, agents interacting with each other through exchanging information is naturally modeled by directed or undirected graphs. A directed graph, also called digraph, is represented by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the node which symbolizes the agent and $\mathcal{E}$ denotes the edge which symbolizes the communication channel. The edges in digraphs are unidirectional, an edge $(i, j) \in \mathcal{E}$ indicates that agent j (child node) receives information from agent i (parent node) but not necessarily vice versa. In addition, node i is a neighbor of node j and the set of neighbors of node j is denoted as $\mathcal{N}_j$. In contrast to digraphs, the edges in undirected graphs are bidirectional, i.e. $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$. Thus an undirected graph can be viewed as a special case of directed graphs.

A directed path in a digraph from nodes i_1 to i_k is a sequence of ordered edges of the form (i_l, i_{l+1}), $l = 1, \cdots, k-1$. A digraph has a directed spanning tree if it contains a node called root which has no parent node and has directed paths from such node to every other nodes. The adjacency matrix $\mathcal{A} = [a_{ij}]$ of a digraph is defined such that a_{ij} is a positive weight if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise ($a_{ii} = 0$, self-edge is not allowed). Throughout this thesis, the weight is assumed to be irrelevant, thus $a_{ij} = 1$ if $(j, i) \in \mathcal{E}$. The associated in-degree matrix $\mathcal{D}$, is a diagonal matrix with its diagonal entry given as $d_{ii} = \sum_{j=1}^{N} a_{ij}, i = 1, \cdots, N$. Then the Laplacian matrix $\mathcal{L}$ is

obtained as $\mathcal{L} = \mathcal{D} - \mathcal{A}$, which is always symmetric for an undirected graph.

Consider the directed graph of five agents shown in Fig. 2.1 as an illustrative

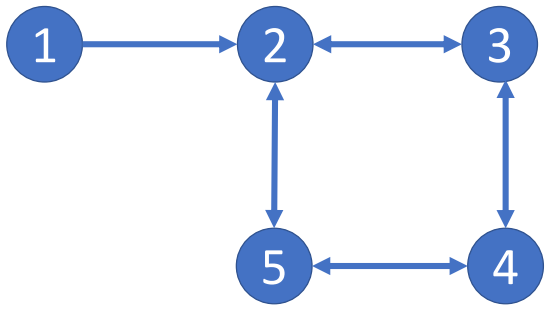

Figure 2.1: Directed graph with five agents

example. For the edge (1,2), node 1 is the parent node and node 2 is the child node. The set of neighbors of node 2 is given as $\mathcal{N}_2 = \{\text{node } 1, \text{ node } 3, \text{ node } 5\}$. This digraph contains a directed spanning tree since there is a directed path from node 1 to every other node (e.g. directed path from node 1 to node 5, (1,2),(2,5) or (1,2),(2,3),(3,4),(4,5)). The adjacency and in-degree matrices of this digraph are given as

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \mathcal{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}. \tag{2.1}$$

The Laplacian matrix is then given as

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}. \tag{2.2}$$

In addition, this digraph can be categorized as a leader-follower structure, where node 1 is the leader which does not receive information from any other nodes. Nodes 2, 3, 4, 5 are the followers. Moreover, the followers form an undirected graph among

themselves and the Laplacian matrix can be partitioned as

$$\mathcal{L} = \left[\begin{array}{c|c} L_{11} & L_{12} \\ \hline L_{21} & L_{22} \end{array}\right],$$

where L_{22} is symmetric and

$$L_{22} = \begin{bmatrix} 3 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}.$$

2.1.2 Consensus Control of Multi-Agent Systems

The consensus problem, which is the development of a consensus algorithm for each agent only based on its local information such that the group of agents can reach an agreement on certain quantities of interest, is a basic MASs coordination problem. A simple consensus framework is shown in Fig. 2.2. Consider a group of N agents with

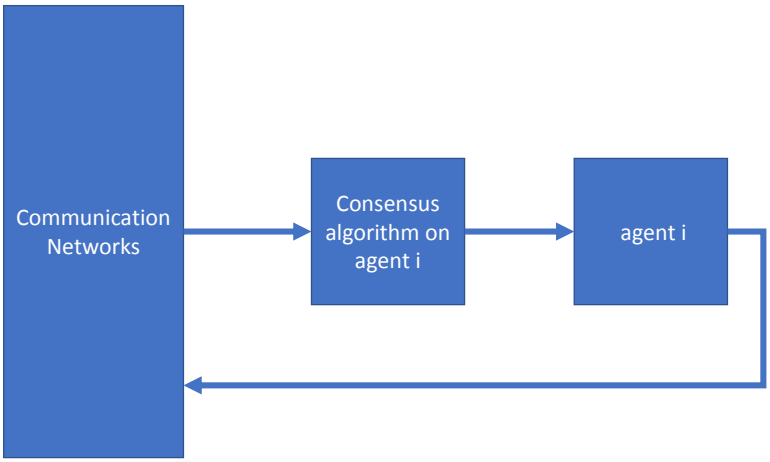

Figure 2.2: A simple consensus framework

dynamics of the i^{th} agent $(i \in 1, \cdots, N)$ given as a 1^{st} order integrator:

$$\dot{\xi}_i(t) = u_i(t), \tag{2.3}$$

where ξ_i is the information state and u_i is the control input. A fundamental consensus algorithm is given by

$$u_i(t) = -\sum_{j=1}^{N} a_{ij}(\xi_i(t) - \xi_j(t)), \tag{2.4}$$

where a_{ij} is the $(i,j)^{th}$ entry of the adjacency matrix associated with the communication graph of N agents. Consensus is said to be achieved if $|\xi_i(t) - \xi_j(t)| \to 0$, as $t \to \infty$, $\forall\, i,j \in 1,\cdots,N$. The basic idea underneath algorithm (2.4) is that each agent is driven towards its neighbors. In addition, this algorithm can be modified accordingly to achieve different control objectives, such as rendezvous(also called consensus), axial alignment and formation maneuvering.

2.1.3 Simulations

This subsection presents simulations for consensus-based cooperative control applications of rendezvous, axial alignment and formation maneuvering. In the rendezvous, agents are required to simultaneously reach an unknown target position obtained through team negotiation. In the axial alignment, agents are required to be evenly distributed on a line with a given separation distance. In the formation maneuvering, agents are required to form a specific geometry and move at a given group velocity.

Rendezvous

Consider a group of four agents with two dimensional single integrator dynamics ($\xi_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$) under communication topology as in Fig. 2.3. Simulations of the group behavior are conducted under following assumptions:

1) it is a homogeneous system ($\xi_i = u_i,\ i = 1,\cdots,N$);

2) the collision and obstacle avoidance are not considered;

3) the low-level control is not considered;

4) the communication delay are not considered;

5) the physical constraints of the agent are neglected.

The rendezvous behavior as shown in Fig. 2.4, is accomplished with the control algorithm (2.4).

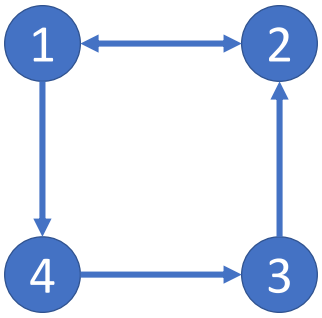

Figure 2.3: Communication topology for rendezvous

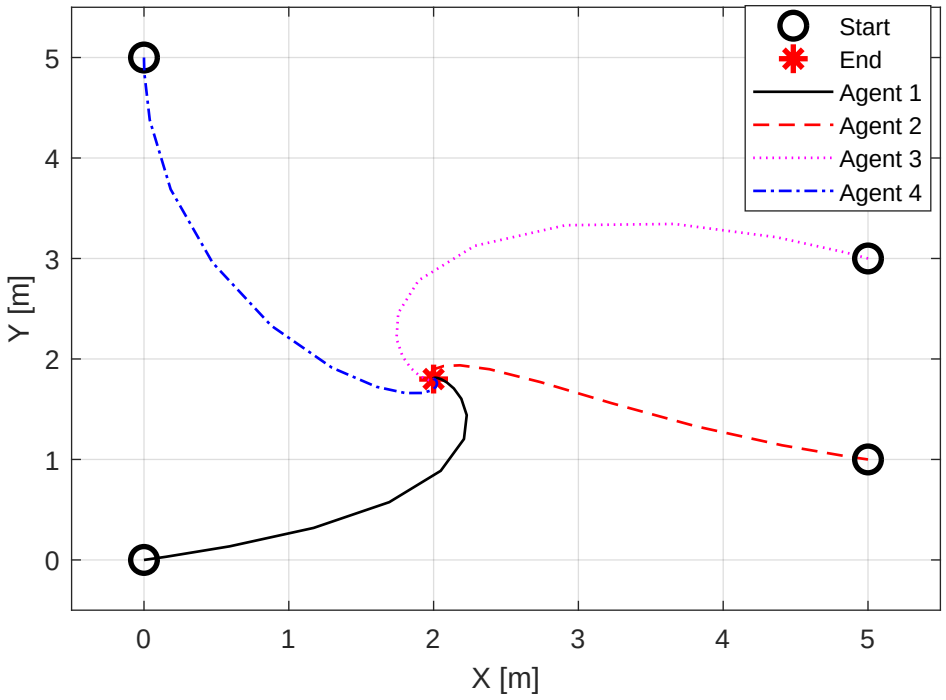

Figure 2.4: Rendezvous

Axial Alignment

In the axial alignment, the fundamental consensus algorithm is modified as

$$u_i = -\sum_{j=1}^{N} a_{ij}[(\xi_i - \xi_j) - (\delta_i - \delta_j)], \tag{2.5}$$

where δ_i is a predefined a constant vector and $\Delta_{ij} = \delta_i - \delta_j$ which is the desired separation distance. With the control algorithm (2.5) and the communication topology

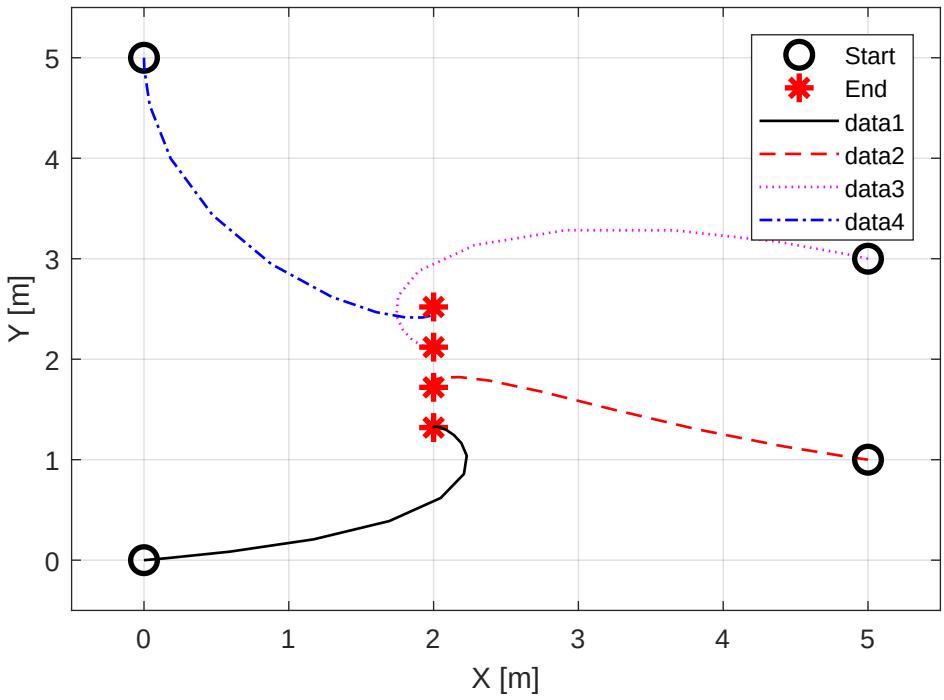

Figure 2.5: Axial alignment

in Fig. 2.3, the group of agents, as shown in Fig. 2.5, are relatively 0.4 m apart in the Y direction at the end, given that $\delta_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\delta_2 = \begin{bmatrix} 0 \\ 0.4 \end{bmatrix}$, $\delta_3 = \begin{bmatrix} 0 \\ 0.8 \end{bmatrix}$ and $\delta_4 = \begin{bmatrix} 0 \\ 1.2 \end{bmatrix}$.

Formation Maneuvering

Consider the leader-follower communication structure given in Fig. 2.1, Agent 1 is the leader, and the rest of the agents are followers. In this simulation, every agent is required to maintain a separation distance of 0.4 m in both X and Y directions with its neighbors (forming a V shape) and simultaneously move at 0.1 m/s in X direction as a group. A proposed control algorithm based on the fundamental consensus law is given as

$$\begin{cases} \dot{\xi}_i &= \dot{\xi}_d - k_i(\xi_i - \xi_d - \delta_i) - \sum_{j=1}^{N} a_{ij}[(\xi_i - \xi_j) - (\delta_i - \delta_j)],\ i = l, \\ \dot{\xi}_i &= \frac{1}{\sum_{j=1}^{N} a_{ij}} \sum_{j=1}^{N} a_{ij}\{\dot{\xi}_j - [(\xi_i - \xi_j) - (\delta_i - \delta_j)]\}, i \neq l, \end{cases} \tag{2.6}$$

where k_i is a positive constant and it is set as 1 in this simulation. l denotes coefficient of the leader. $\xi_d = \begin{bmatrix} 0.1t \\ 2.5 \end{bmatrix}$ is the desired trajectory. The constant separation vectors are given as $\delta_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\delta_2 = \begin{bmatrix} -0.4 \\ 0.4 \end{bmatrix}$, $\delta_3 = \begin{bmatrix} -0.4 \\ -0.4 \end{bmatrix}$, $\delta_4 = \begin{bmatrix} -0.8 \\ 0.8 \end{bmatrix}$, and $\delta_5 = \begin{bmatrix} -0.8 \\ -0.8 \end{bmatrix}$. The system response under the controller (2.6) is shown in Fig. 2.6.

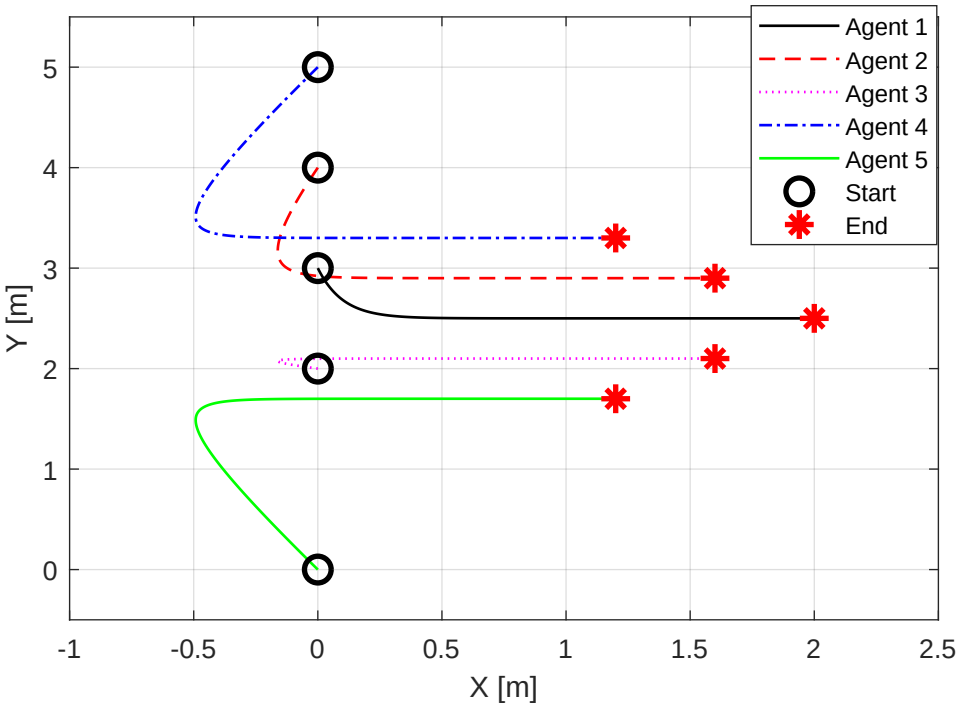

Figure 2.6: Formation maneuvering

2.2 Lyapunov-Based Methods

The Lyapunov-based controller design method is introduced in this section. This section is mainly based on [33, 34].

2.2.1 Lyapunov Stability

Consider $x = 0$, without loss of generality, to be an equilibrium point ($f(0) = 0$) of a single-input-single-output (SISO) autonomous system $\dot{x} = f(x)$. An equilibrium point is stable if the system trajectory remains nearby when it starts nearby.

An equilibrium point is asymptotically stable if the system trajectory asymptotically converges to this point when it starts nearby. Lyapunov stability theorem states that, $x = 0$ is a stable (asymptotically stable) equilibrium point if there exists a continuously-differentiable positive-definite function $V(x)$ (known as Lyapunov function) and its time derivative $\dot{V}(x)$ is negative semi-definite (negative definite). $V(x)$ is positive definite (semidefinite) if $V(x) > 0$ ($V(x) \geq 0$), $\forall x \neq 0$ and $V(0) = 0$, and the negative definiteness (semidefiniteness) is defined analogously. In addition, if the Lyapunov function is radially unbounded ($V(x) \to \infty$ as $\|x\| \to \infty$), then global asymptotic stability can be achieved. At last, the above statements are also valid in multidimensional systems.

2.2.2 Controller Design Based on Lyapunov's Method

There are basically two ways of applying the Laypunov's method in the controller design, and both are trial-and-error methods. The first method requires assuming a certain form of the control law and then finding a Lyapunov function that justifies the choice. In contrast, the second method involves hypothesizing a Lyapunov function candidate and then searching for a control law that can realize such a candidate. The second method is employed in control design in this thesis.

First Method

An n-link robot manipulator has the following dynamics

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \tag{2.7}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the robot joint position vector, $\boldsymbol{\tau}$ is the joint torque input vector, $H(\cdot) \in \mathbb{R}^{n \times n}$ is the inertia matrix (positive definite), $C(\cdot)$ represents the Coriolis and centripetal forces, and $\mathbf{g}(\cdot)$ is the gravitational torque vector. Consider a proportional-derivative (PD) controller with the gravity compensation term

$$\boldsymbol{\tau} = -K_D\dot{\mathbf{q}} - K_P\mathbf{q} + \mathbf{g}(\mathbf{q}), \tag{2.8}$$

where K_D and K_P are positive definite constant matrices. To analyze the stability of the closed loop dynamics, a Lyapunov function candidate mimicking the physical

energy of a manipulator system is proposed as

$$V = \frac{1}{2}[\dot{\mathbf{q}}^T H \dot{\mathbf{q}} + \mathbf{q}^T K_p \mathbf{q}]. \tag{2.9}$$

Then the time derivative of the candidate, with the aid of the skew symmetric property of $\dot{M} - 2C$, is obtained as

$$\dot{V} = -\dot{\mathbf{q}}^T K_D \dot{\mathbf{q}}. \tag{2.10}$$

Thus the proposed Lyapunov function candidate is a real Lyapunov function and the system under the controller (2.8) is asymptotically stable.

Second Method

Consider the problem of stabilizing the linear system

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \tag{2.11}$$

where $\mathbf{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $\mathbf{u} \in \mathbb{R}^m$. It is sufficient to choose the control law with the following form

$$\mathbf{u} = K\mathbf{x}, \tag{2.12}$$

where $K \in \mathbb{R}^{m\times n}$. Consider a Lyapunov function candidate for the closed-loop dynamics as

$$V(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T P \mathbf{x}, \tag{2.13}$$

where P is a symmetric positive definite matrix. Its time derivative is then given as

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T[(A + BK)^T P + P(A + BK)]\mathbf{x}. \tag{2.14}$$

The Lyapunov function candidate (2.13) is a real Lyapunov function, thus the closed-loop dynamics is asymptotically stable, if K is the solution of

$$(A + BK)^T P + P(A + BK) = -Q, \tag{2.15}$$

where Q is an arbitrary positive definite matrix.

2.3 Matrix Theory

This section introduces the essential matrix concepts used in this thesis.

2.3.1 Properties in Linear Matrix Inequality

Many control problems can be formulated as LMI feasibility problems. The following lemma is very useful in converting a class of convex nonlinear inequalities to an LMI.

Lemma 1. [35] *The Schur Complement states that for any symmetric negative definite matrix $C \in \mathbb{R}^{q\times q}$, symmetric matrix $A \in \mathbb{R}^{p\times p}$, and matrix $B \in \mathbb{R}^{p\times q}$, the LMIs*

$$C < 0, \quad A - BC^{-1}B^T < 0, \tag{2.16}$$

are equivalent to

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} < 0. \tag{2.17}$$

2.3.2 Quadratic Integral Inequality

Lemma 2. [36] *Given any constant symmetric positive definite matrix $E \in \mathbb{R}^{n\times n}$, and vector function $\boldsymbol{\omega} : [t-\tau, t] \to \mathcal{R}^n$ where the integrations concerned are well defined, then*

$$-\tau \int_{t-\tau}^{t} \boldsymbol{\omega}^T(s) E \boldsymbol{\omega}(s) ds \leq -\left[\int_{t-\tau}^{t} \boldsymbol{\omega}(s) ds\right]^T E \left[\int_{t-\tau}^{t} \boldsymbol{\omega}(s) ds\right]. \tag{2.18}$$

2.3.3 Kronecker Product

For $A \in \mathbf{R}^{m\times n}$, $B \in \mathbf{R}^{p\times q}$, the kronecker product of A and B, $A \otimes B$, is a $mp \times nq$ matrix defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}.$$

The Kronecker product satisfies the following rules of calculation [37]:

$$
\begin{aligned}
(kA) \otimes B &= A \otimes (kB) = k(A \otimes B) \\
(A + B) \otimes C &= A \otimes C + B \otimes C \\
A \otimes (B \otimes C) &= (A \otimes B) \otimes C \\
(A \otimes B)(C \otimes D) &= (AC) \otimes (BD) \\
(A \otimes B)^T &= A^T \otimes B^T \\
(A \otimes B)^{-1} &= A^{-1} \otimes B^{-1},
\end{aligned}
$$

where k is a constant and the last property holds if and only if both A and B are invertible.

Chapter 3

Problem Formulation

This chapter presents the system modeling for a quadcopter. Simplified models for multiple-quadcopter systems are then discussed. The control objective is also defined. This section presents the dynamic model of a quadcopter obtained via Euler-Lagrange equations.

3.1 Quadcopter Characteristics

A quadcopter, also known as a quadrotor, is basically a helicopter with four rotors. The inertial frame $\mathrm{E_I}(x, y, z)$ and body frame $\mathrm{E_B}(x_\mathrm{B}, y_\mathrm{B}, z_\mathrm{B})$ coordinates are utilized to represent the quadcopter configuration given in Fig. 3.1. It is assumed that, the

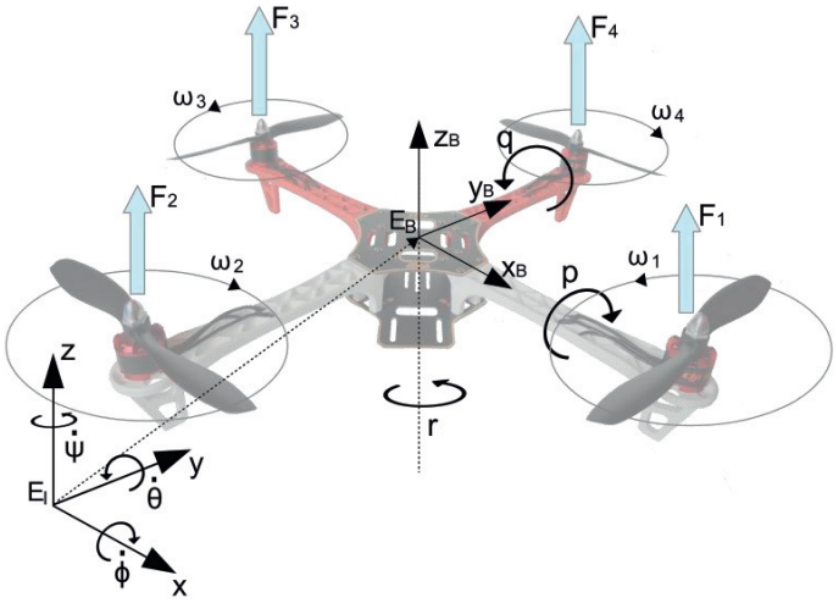

Figure 3.1: Quadcopter configuration [3]

origin of $\mathrm{E_B}$ is attached to the center of mass of the quadcopter and moving along with it. The absolute linear position (x, y, z) is defined in $\mathrm{E_I}$ as $\boldsymbol{\xi}$. The angular

position (ϕ, θ, ψ), where ϕ, θ, ψ denote the roll, pitch and yaw Euler angles (rotation around x,y,z-axes), is defined as $\boldsymbol{\eta}$ in $\mathrm{E_I}$. In the body frame $\mathrm{E_B}$, the linear and angular velocities are defined as $\mathbf{v}_B$ and $\boldsymbol{\nu}$.

$$\boldsymbol{\xi} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \boldsymbol{\eta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \mathbf{v}_B = \begin{bmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}.$$

The rotation matrix (zyx) [38] from $\mathrm{E_B}$ to $\mathrm{E_I}$ is given as

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix},$$

where S_x and C_x denote $\sin x$ and $\cos x$. In addition, the rotation matrix R is orthogonal thus $R^{-1} = R^T$. The angular velocity $\boldsymbol{\nu}$ in $\mathrm{E_B}$ and the rate of change of Euler angles $\dot{\boldsymbol{\eta}}$ are related by the transformation matrix W_η as

$$\boldsymbol{\nu} = W_\eta \dot{\boldsymbol{\eta}}, \tag{3.1}$$

and

$$\dot{\boldsymbol{\eta}} = W_\eta^{-1} \boldsymbol{\nu}, \tag{3.2}$$

where W_η^{-1} and W_η are given in [39] as,

$$W_\eta^{-1} = \begin{bmatrix} 1 & S_\phi T_\theta & C_\phi T_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix}, W_\eta = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix},$$

where T_x denotes $\tan x$. The quadcopter is assumed to have symmetric structure about x_B and y_B axes, thus $I_{xx} = I_{yy}$ and the inertia matrix of the quadcopter is diagonal and given as

$$I = \begin{bmatrix} I_{xx} & & \\ & I_{yy} & \\ & & I_{zz} \end{bmatrix}.$$

It is also assumed that the thrust force f_i generated by the i^{th} rotor is proportional to the square of its rotational speed w_i [40]. The proportionality k is constant for all rotors and the thrust force is as

$$f_i = k w_i^2, \ i = 1, 2, 3, 4. \tag{3.3}$$

Therefore, the force input vector can be expressed as

$$\mathbf{f}_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} f_i \end{bmatrix}, \tag{3.4}$$

where T is the total thrust. And the torque input vector is

$$\boldsymbol{\tau}_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l(f_4 - f_2) \\ l(f_3 - f_1) \\ d(-f_1 + f_2 - f_3 + f_4) \end{bmatrix}, \tag{3.5}$$

where l is the link length (center to rotor), d is the torque factor (related to the torque generated around rotor axis from its rotation). In addition, some researchers also included the aerodynamic drag terms [41], while assuming the translational drag force is proportional to the relative linear velocity with respect to the air, and rotational drag is proportional to relative angular velocity. However, for the sake of simplicity, the aerodynamic terms will be neglected in the sequel. The quadcopter is an underactuated system since it has six degrees of freedom (6-DOF: $x, y, z, \phi, \theta, \psi$) but only has four inputs ($T, \tau_\phi, \tau_\theta, \tau_\psi$). This means, only up to 4-DOF (pitch and roll motions are coupled with motions in x and y directions) can be independently controlled. For instance, the rotational pitch motion will always result in a translational motion in x-direction.

Newton-Euler[42, 43] and Euler-Lagrange [40, 44] equations have been widely adopted in modeling the quadcopter dynamics. In the following subsection, the quadcopter dynamic model is presented via Euler-Lagrange formalism. The nomenclatures of the quadcopter system are summarized in Table 3.1.

Table 3.1: Nomenclature of quadcopter system

Parameter	Definition
$\mathrm{E_I}$	Inertial frame coordinate system
$\mathrm{E_B}$	Body frame coordinate system
$\boldsymbol{\xi}$	Linear position in the inertial frame
$\boldsymbol{\eta}$	Angular position in the inertial frame
v_B	Linear velocity in the body frame
$\boldsymbol{\nu}$	Angular velocity in the body frame
I	Inertial matrix
w_i	Rotation speed of the i^{th} rotor
f_i	Thrust force generated by the i^{th} rotor
l	Link length (from center to rotor)
$\mathbf{f}_B$	Force input vector in the body frame
$\boldsymbol{\tau}_B$	Torque input vector in the body frame

3.2 Quadcopter Dynamics

The Lagrangian $\mathcal{L}$ is defined as the kinetic energy minus the potential energy, where the kinetic energy is the sum of the translational energy E_{trans} and the rotational energy E_{rot} of the system.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = E_{trans} + E_{rot} - E_{pot} = \frac{m}{2}\dot{\boldsymbol{\xi}}^T\dot{\boldsymbol{\xi}} + \frac{1}{2}\boldsymbol{\nu}^T I\boldsymbol{\nu} - mgz, \tag{3.6}$$

where $\mathbf{q} = \begin{bmatrix}\boldsymbol{\xi}^T & \boldsymbol{\eta}^T\end{bmatrix}^T$. Then the Euler-Lagrange equations with external forces and torques are given as

$$\begin{bmatrix}\mathbf{f} \\ \boldsymbol{\tau}_B\end{bmatrix} = \frac{d}{dt}(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}}. \tag{3.7}$$

The linear and angular components are independent thus can be studied separately and the linear part is given below

$$\mathbf{f} = R\mathbf{f}_B = m\ddot{\boldsymbol{\xi}} + mg\begin{bmatrix}0\\0\\1\end{bmatrix}, \tag{3.8}$$

which can be rearranged as

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = -g\begin{bmatrix}0\\0\\1\end{bmatrix} + \frac{T}{m}\begin{bmatrix}C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi\end{bmatrix}. \tag{3.9}$$

The rotational energy can also be expressed in the inertial frame as

$$E_{rot} = \frac{1}{2}\dot{\boldsymbol{\eta}}^T J \dot{\boldsymbol{\eta}}, \tag{3.10}$$

where the Jacobian matrix $J(\boldsymbol{\eta})$ can be obtained from (3.1) as

$$\begin{aligned} J(\boldsymbol{\eta}) =& W_\eta^T I W_\eta \\ =& \begin{bmatrix} I_{xx} & 0 & -I_{xx}S_\theta \\ 0 & I_{yy}C_\phi^2 + I_{zz}S_\phi^2 & (I_{yy} - I_{zz})C_\phi S_\phi C_\theta \\ -I_{xx}S_\theta & (I_{yy} - I_{zz})C_\phi S_\phi C_\theta & I_{xx}S_\theta^2 + I_{yy}S_\phi^2 C_\theta^2 + I_{zz}C_\phi^2 C_\theta^2 \end{bmatrix}. \end{aligned}$$

The angular part of Euler-Lagrange equation is then given as

$$\boldsymbol{\tau}_B = J(\eta)\ddot{\boldsymbol{\eta}} + \frac{d}{dt}(J(\eta))\dot{\boldsymbol{\eta}} - \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\eta}}(\dot{\boldsymbol{\eta}}^T J(\eta)\dot{\boldsymbol{\eta}}) = J(\eta)\ddot{\boldsymbol{\eta}} + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}}, \tag{3.11}$$

where $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ is Coriolis term [44] given as.

$$\mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix},$$

where

$$\begin{aligned} C_{11} &= 0 \\ C_{12} &= (I_{yy} - I_{zz})(\dot{\theta}C_\phi S_\phi + \dot{\psi}S_\phi^2 C_\theta) + (I_{zz} - I_{yy})\dot{\psi}C_\phi^2 C_\theta - I_{xx}\dot{\psi}C_\theta \\ C_{13} &= (I_{zz} - I_{yy})\dot{\psi}C_\phi S_\phi C_\theta^2 \\ C_{21} &= (I_{zz} - I_{yy})(\dot{\theta}C_\phi S_\phi + \dot{\psi}S_\phi C_\theta) + (I_{yy} - I_{zz})\dot{\psi}C_\phi^2 C_\theta + I_{xx}\dot{\psi}C_\theta \\ C_{22} &= (I_{zz} - I_{yy})\dot{\phi}C_\phi S_\phi \\ C_{23} &= -I_{xx}\dot{\psi}S_\theta C_\theta + I_{yy}\dot{\psi}S_\phi^2 S_\theta C_\theta + I_{zz}\dot{\psi}C_\phi^2 S_\theta C_\theta \\ C_{31} &= (I_{yy} - I_{zz})\dot{\psi}C_\theta^2 S_\phi C_\phi - I_{xx}\dot{\theta}C_\theta \\ C_{32} &= (I_{zz} - I_{yy})(\dot{\theta}C_\phi S_\phi S_\theta + \dot{\phi}S_\phi^2 C_\theta) + (I_{yy} - I_{zz})\dot{\phi}C_\phi^2 C_\theta \\ &\quad I_{xx}\dot{\psi}S_\theta C_\theta - I_{yy}\dot{\psi}S_\phi^2 S_\theta C_\theta - I_{zz}\dot{\psi}C_\phi^2 S_\theta C_\theta \\ C_{33} &= (I_{yy} - I_{zz})\dot{\phi}C_\phi S_\phi C_\theta^2 - I_{yy}\dot{\theta}S_\phi^2 C_\theta S_\theta - I_{zz}\dot{\theta}C_\phi^2 C_\theta S_\theta + I_{xx}\dot{\theta}C_\theta S_\theta. \end{aligned}$$

Assume the singular position does not occur ($\theta \neq \frac{\pi}{2}$) in all control situations, thus the inverse of Jacobian matrix J always exists, and then the angular dynamics (3.12) can be rearranged as

$$\ddot{\boldsymbol{\eta}} = J^{-1}(\boldsymbol{\eta})(\boldsymbol{\tau}_B - C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}}). \tag{3.12}$$

3.3 Simplified Quadcopter Models

Various simplifications on quadcopter dynamics have been employed in the controller design in the literature for dealing with the highly coupled nonlinearities. Most simplifications are based on certain linearization techniques such as the planar motion assumption, full linearization, feedback linearization and the small angle variation assumption.

3.3.1 Partially-Linearized Quadcopter Model

A partially-linearized model, where the angular dynamics (3.12) is feedback linearized while the linear dynamic remains unchanged, has been used extensively in the literature [45]. A feedback linearization law (assume ϕ, θ, ψ, $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ are measurable) is applied to change the torque input variables as,

$$\boldsymbol{\tau}_B = C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + J\tilde{\boldsymbol{\tau}}, \tag{3.13}$$

where $\tilde{\boldsymbol{\tau}}$ are the new torque inputs and

$$\tilde{\boldsymbol{\tau}} = \begin{bmatrix} \tilde{\tau}_\phi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\psi \end{bmatrix}. \tag{3.14}$$

Then the angular dynamics can be linearized as

$$\ddot{\boldsymbol{\eta}} = \tilde{\boldsymbol{\tau}}. \tag{3.15}$$

Thus the system dynamics becomes

$$\begin{cases} \ddot{x} &= \frac{T}{m}(C_\psi S_\theta C_\phi + S_\psi S_\phi) \\ \ddot{y} &= \frac{T}{m}(S_\psi S_\theta C_\phi - C_\psi S_\phi) \\ \ddot{z} &= \frac{T}{m} C_\theta C_\phi - g \\ \ddot{\phi} &= \tilde{\tau}_\phi \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\psi} &= \tilde{\tau}_\psi, \end{cases} \tag{3.16}$$

This model was initially used in [46, 45] where nonlinear controllers based on the nested saturation method were proposed for real-time stabilization and tracking. A

new sliding mode control algorithm was then introduced for this underactuated system [47]. Motivated by this work, the distributed sliding mode control algorithms were developed for the leaderless consensus as well as a virtual leader based consensus tracking for a group of quadcopters [48]. The same authors also considered the multi-quadcopter consensus problem using zero dynamics method under directed and switching communication topologies [49]. The containment control for multiple quadcopter system has also been discussed in [50]. A feedback linearization controller and an adaptive sliding mode controller were proposed and discussed in [51]. In addition, a similar model, as given in Appendix B.1 which is derived based on the left-handed coordinate system as in Appendix C, was also widely adopted in the literature.

3.3.2 Fully-Linearized Quadcopter Model

The fully linearized quadcopter model has been mainly used in the H_∞ controller [52, 53, 54] and linear-quadratic regulator (LQR) controller design [55]. The partially linearized dynamics (3.16) can be rewritten in a state-space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ (\frac{u_1}{m} + g)(C_{x_7} S_{x_9} C_{x_{11}} + S_{x_7} S_{x_{11}}) \\ x_4 \\ (\frac{u_1}{m} + g)(S_{x_7} S_{x_9} C_{x_{11}} - C_{x_7} S_{x_{11}}) \\ x_6 \\ (\frac{u_1}{m} + g)(C_{x_9} C_{x_{11}}) - g \\ x_8 \\ u_2 \\ x_{10} \\ u_3 \\ x_{12} \\ u_4 \end{bmatrix}, \tag{3.17}$$

where the states are defined as

$$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}]^T = [x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z} \quad \psi \quad \dot{\psi} \quad \theta \quad \dot{\theta} \quad \phi \quad \dot{\phi}]^T,$$

and the control inputs are given as

$$\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T = \begin{bmatrix} T - mg & \tilde{\tau}_\psi & \tilde{\tau}_\theta & \tilde{\tau}_\phi \end{bmatrix}^T$$

The state-space representation (3.17) was then linearized around the origin ($f(0,0) = 0$, equilibrium point) by Taylor series expansion, which produces the following linear system

$$\dot{x} = Ax + Bu, \tag{3.18}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T.$$

To implement this model, the system has to be operating around equilibrium states. With this in mind, H_∞ robust and LQR optimal controllers are favored. Moreover, a fully linearized model based on the left-handed coordinate system, perhaps more popular in literature, is included in Appendix B.2.

3.3.3 Planar Quadcopter Model

Planar quadcopter model can be obtained if it flies slowly enough, thus the external aerodynamic forces can be neglected, and no yawing moment will ever be generated.

Under these assumptions, linearized longitudinal and lateral models under a constant altitude equilibrium point are given as follows

$$\frac{d}{dt}\begin{bmatrix} x \\ u \\ \theta \\ q \end{bmatrix} = \begin{bmatrix} u \\ -g\theta \\ q \\ \frac{M_\theta}{I_{yy}} \end{bmatrix}, \frac{d}{dt}\begin{bmatrix} y \\ v \\ \phi \\ p \end{bmatrix} = \begin{bmatrix} v \\ g\phi \\ p \\ \frac{M_\phi}{I_{xx}} \end{bmatrix}, \tag{3.19}$$

where M_θ and M_ϕ are associated torque inputs. If we define, $\bar{\theta} = -g\theta$, $\bar{q} = -gq$, $\overline{M}_\theta = -\frac{g}{I_{yy}}M_\theta$, $\bar{\phi} = g\phi$, $\bar{p} = gp$ and $\overline{M}_\phi = \frac{g}{I_{xx}}M_\phi$. Then the model (3.19) can be rewritten as

$$\frac{d}{dt}\begin{bmatrix} x \\ u \\ \bar{\theta} \\ \bar{q} \end{bmatrix} = \begin{bmatrix} u \\ \bar{\theta} \\ \bar{q} \\ \overline{M}_\theta \end{bmatrix}, \frac{d}{dt}\begin{bmatrix} y \\ v \\ \bar{\phi} \\ \bar{p} \end{bmatrix} = \begin{bmatrix} v \\ \bar{\phi} \\ \bar{p} \\ \overline{M}_\phi \end{bmatrix}, \tag{3.20}$$

Then system (3.20) can be grouped into a lumped linear fourth order system by defining $r^0 = \begin{bmatrix} x \\ y \end{bmatrix}$, $r^1 = \begin{bmatrix} u \\ v \end{bmatrix}$, $r^2 = \begin{bmatrix} \bar{\theta} \\ \bar{\phi} \end{bmatrix}$, $r^3 = \begin{bmatrix} \bar{q} \\ \bar{p} \end{bmatrix}$, and $M = \begin{bmatrix} \overline{M}_\theta \\ \overline{M}_\phi \end{bmatrix}$, and is given as follows

$$\frac{d}{dt}\begin{bmatrix} r^0 \\ r^1 \\ r^2 \\ r^3 \end{bmatrix} = \begin{bmatrix} r^1 \\ r^2 \\ r^3 \\ M \end{bmatrix}. \tag{3.21}$$

This model was primarily used in the formation control of multiple-quadcopter systems with [56] and without [57] collision avoidance. Their results have been experimentally validated using Parrot AR.Drone, which is a slow-flying quadcopter platform.

3.3.4 Quadcopter Model under Small Angle Variation

Perhaps, the most common assumption that researchers make in simplifying the quadcopter model is that the quadcopter only performs many angular motions of low amplitude, thus $W_\eta \approx I_3$, $\boldsymbol{\nu} \approx \dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\nu}} \approx \ddot{\boldsymbol{\eta}}$. Under this assumption, the translational

dynamics remains same as

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{T}{m} \begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix} - g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.22}$$

whereas the rotational dynamics (3.12) becomes

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})\dot{\theta}\dot{\psi}/I_{xx} \\ (I_{zz} - I_{xx})\dot{\phi}\dot{\psi}/I_{yy} \\ (I_{xx} - I_{yy})\dot{\phi}\dot{\theta}/I_{zz} \end{bmatrix} + \begin{bmatrix} \tau_\phi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{bmatrix}. \tag{3.23}$$

In [58], an adaptive sliding mode flight controller was proposed for quadcopter attitude stabilization and altitude trajectory tracking in the presence of parameter uncertainties. The slightly different second order sliding mode controller in [59] also addressed the parameter uncertainties. A sliding mode controller based on the backstepping approach was proposed in [60]. It was then extended to the output feedback in conjunction with a nonlinear full order observer design in [61]. A multi-mode flight sliding mode control system encompassing manual, altitude, global positioning system (GPS) fixed and autonomous mode was designed and analyzed using the switching nonlinear theory in [62].

3.4 Simplified Quadcopter Dynamics for Multi-Agent Systems

Fully linearized and planar models are only valid for systems operating around the equilibrium states which can impose many constraints in the controller design. The model under small angle variations is still too complicated to study in a MAS scale. Therefore, the partially linearized model (3.16) is employed for studying multiple-quadcopter systems. However, further simplifications are required in the actual distributed controller design. Specifically, the controllers are designed in a such way that the system dynamics in ψ and z are collapsed ($\ddot{z} = 0$, $\ddot{\psi} = 0$). To illustrate the

designing procedures, system (3.16) can be rewritten as

$$\begin{cases} \ddot{x} &= u_1(C_\psi S_\theta C_\phi + S_\psi S_\phi) \\ \ddot{y} &= u_1(S_\psi S_\theta C_\phi - C_\psi S_\phi) \\ \ddot{z} &= u_1 C_\theta C_\phi - g \\ \ddot{\phi} &= u_2 \\ \ddot{\theta} &= u_3 \\ \ddot{\psi} &= u_4, \end{cases} \tag{3.24}$$

where $u_1 = \frac{T}{m}$, $u_2 = \tilde{\tau}_\phi$, $u_3 = \tilde{\tau}_\theta$ and $u_4 = \tilde{\tau}_\psi$. In order to eliminate the dynamics in the z-direction, the control input $u_1 = \frac{g}{C_\theta C_\phi}$ has to be satisfied. With this input, the system dynamics in the x and y directions can be written as

$$\ddot{x} = g(\frac{\cos\phi \sin\theta \cos\psi}{\cos\phi\cos\theta} + \frac{\sin\phi\sin\psi}{\cos\phi\cos\theta}), \tag{3.25}$$

$$\ddot{y} = g(\frac{\cos\phi \sin\theta \sin\psi}{\cos\phi\cos\theta} - \frac{\sin\phi\cos\psi}{\cos\phi\cos\theta}). \tag{3.26}$$

If we can design a controller so that $\psi \to 0$ asymptotically (dynamics collapses in ψ), which can be easily achieved by a PD controller, then the system dynamics in x and y directions can be simplified as

$$\ddot{x} \approx g\tan\theta, \tag{3.27}$$

$$\ddot{y} \approx -g\frac{\tan\phi}{\cos\theta}, \tag{3.28}$$

It can be observed from (3.27) that x is only coupled with θ, so if we can design a distributed controller such that $\theta \to 0$ asymptotically. The system dynamics in the y-direction can be further simplified as

$$\ddot{y} \approx -g\tan\phi, \tag{3.29}$$

which is decoupled from θ. Then under such proposed controllers, (3.24) can be rewritten as

$$\begin{cases} \ddot{x} &= g\tan\theta \\ \ddot{y} &= -g\tan\phi \\ \ddot{z} &= u_1 C_\theta C_\phi - g \\ \ddot{\phi} &= u_2 \\ \ddot{\theta} &= u_3 \\ \ddot{\psi} &= u_4. \end{cases} \tag{3.30}$$

It can be observed from the simplified system dynamics above that x and y are only related to θ and ϕ while they are decoupled with the input u_1 and the other two Euler angles. If we define $v = g\tan\theta$, then the system dynamics in the x-direction can be transformed into a fourth order integrator

$$\begin{cases} \ddot{x} = v \\ \ddot{v} = u_3', \end{cases} \tag{3.31}$$

where $u_3' = 2\frac{v\dot{v}^2}{g^2+v^2} + (g + \frac{v^2}{g})u_3$.
In the same manner, define $w = -g\tan\phi$, thus the system dynamics in the y-direction becomes

$$\begin{cases} \ddot{y} = w \\ \ddot{w} = u_2', \end{cases} \tag{3.32}$$

where $u_2' = 2\frac{\dot{w}^2 w}{g^2+w^2} - (g + \frac{w^2}{g})u_2$. The system dynamics in the z-direction is given as

$$\ddot{z} = u_1', \tag{3.33}$$

where $u_1' = u_1(\cos\phi\cos\theta) - g$. And, the system dynamics in the yaw motion remains the same as

$$\ddot{\psi} = u_4. \tag{3.34}$$

Thus the quadcopter system dynamics has been transformed to four decoupled linear subsystems (two fourth-order and two double integrator systems). Consequently, the consensus algorithm can be designed based on the decoupled model shown above. A fundamental control law for high-order systems which may serve as a starting point

for the delayed controller design, is given in [63]

$$u_i = -\sum_{j=1}^{N} a_{ij} \left[\sum_{k=0}^{l-1} r_k (\xi_i^{(k)} - \xi_j^{(k)}) \right], \quad i \in \{1, 2, \cdots, N\}, \tag{3.35}$$

where l denotes the system order, ξ_i denotes the system state, and r_k denotes control gain. However, it is not only required that all the states reach consensus, but also that the states $\dot{x}, v, \dot{v}$ should all converge to 0. Thus the controller for the fourth-order subsystems in our simplified quadcopter model may be modified as

$$\begin{aligned} u_i = k_1\dot{x} + k_2 v + k_3\dot{v} - r_1 \sum_{j=1}^{n} a_{ij}(x_i - x_j) - r_2 \sum_{j=1}^{n} a_{ij}(\dot{x}_i - \dot{x}_j) \\ - r_3 \sum_{j=1}^{n} a_{ij}(v_i - v_j) - r_4 \sum_{j=1}^{n} a_{ij}(\dot{v}_i - \dot{v}_j). \end{aligned} \tag{3.36}$$

In the presence of delays among communication channels, the delayed state information may be used in the controller above.

3.5 Control Objectives

The control objective of this work it to design a distributed consensus control law for each quadcopter agent with system dynamics (3.24), in the presence of synchronous constant or asynchronous time-varying delays among each communication channel. In particular, the consensus algorithm will be designed based on four decoupled linear systems and the resulting system response should guarantee that the system can be decoupled. In other words, the control objective is achieved for a group of N quadcopters if $\lim_{t\to\infty} \|\boldsymbol{\xi}_j - \boldsymbol{\xi}_i\| = 0$ and $\lim_{t\to\infty} \|\boldsymbol{\eta}_i\| = 0$, $\forall i, j \in 1, \cdots, N$, which means the translational positions are asymptotically converging to each other and the angular positions are asymptotically converging to zero.

Chapter 4

Controller Design

This chapter discusses the static consensus controller design for MASs with general linear dynamic agents under synchronous constant and asynchronous time-varying delays respectively. The system error dynamics are formed at first for both cases and then the controller gains can be obtained from solving the LMI conditions derived based on Lyapunov's method.

4.1 System Error Dynamics

To apply the Lyapunov-based controller design methodology, the system error dynamics of the MASs are derived. The system error dynamics for both constant and time-varying delay cases will be described in this section.

4.1.1 Case I: Constant Delay

Consider a MAS consisting of N agents with i^{th} agent described by the linear system dynamics

$$\dot{\mathbf{x}}_i = A\mathbf{x}_i + Bu'_i, \;\; i \in 1, \cdots, N, \tag{4.1}$$

where $\mathbf{x}_i \in \mathbb{R}^n$, $u'_i \in \mathbb{R}$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$. Matrices and vectors are assumed to have compatible dimensions if not explicitly stated.

In the presence of synchronous constant delay among communication channels, the following delayed feedback control law is proposed for the i^{th} agent

$$\begin{cases} u'_i = u'_{ia} + u'_{ib} \\ u'_{ia} = \mathbf{l}\mathbf{x}_i \\ u'_{ib} = -\mathbf{k}\sum\limits_{j\in\mathcal{N}_i} a_{ij}(\mathbf{x}_i(t-\tau) - \mathbf{x}_j(t-\tau)), \end{cases} \tag{4.2}$$

where u'_{ia} is the local feedback controller and u'_{ib} is the network interaction feedback controller, τ is the known constant communication delay, $\mathbf{k}, \mathbf{l}$ are static row vector

gains that will be designed separately. The local feedback gain $\mathbf{l}$ can be designed using pole placement method while the network interaction gain $\mathbf{k}$ will be designed using Lyapunov based method. Then with the controller (4.2), the system (4.1) can be written as

$$\dot{\mathbf{x}}_i = (A + B\mathbf{l})\mathbf{x}_i - B\mathbf{k}\sum_{j\in\mathcal{N}_i} a_{ij}(\mathbf{x}_i(t-\tau) - \mathbf{x}_j(t-\tau)),\ \ i \in 1, \cdots, N. \tag{4.3}$$

Define the system error for the i^{th} agent as $\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}_1$, $i \in 2, \cdots, N$. Therefore, the error dynamics can be obtained as

$$\dot{\mathbf{e}}_i = (A + B\mathbf{l})\mathbf{e}_i - B\mathbf{k}\sum_{j\in\mathcal{N}_i} a_{ij}(\mathbf{e}_i(t-\tau) - \mathbf{e}_j(t-\tau)) - B\mathbf{k}\sum_{j\in\mathcal{N}_1} a_{1j}\mathbf{e}_j(t-\tau). \tag{4.4}$$

The overall error dynamics is then given as

$$\dot{\mathbf{e}} = \bar{A}\mathbf{e} + \bar{L}\mathbf{e}(t-\tau), \tag{4.5}$$

where $\mathbf{e} = \begin{bmatrix} \mathbf{e}_2 \\ \mathbf{e}_3 \\ \vdots \\ \mathbf{e}_N \end{bmatrix}$, $\bar{A} = \mathbf{I}_{N-1} \otimes (A + B\mathbf{l})$, $\bar{L} = L \otimes (B\mathbf{k})$ and

$$L = \begin{bmatrix} -\sum_{j=1}^{N} a_{2j} - a_{12} & a_{23} - a_{13} & \cdots & a_{2N} - a_{1N} \\ a_{32} - a_{12} & -\sum_{j=1}^{N} a_{3j} - a_{13} & \cdots & a_{3N} - a_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N2} - a_{12} & \cdots & \cdots & -\sum_{j=1}^{N} a_{Nj} - a_{1N} \end{bmatrix}. \tag{4.6}$$

Remark. *The L matrix characterizes the communication topology among agents. In a leader-follower type consensus structure with the leader defined as agent 1, $a_{1j} = 0$ if the leader does not receive any information from its followers. In addition L is symmetric if the communication between followers is bidirectional. Throughout this thesis, L is assumed to be symmetric.*

The system (4.1) asymptotically achieves consensus under the controller (4.2) if the overall system error dynamics (4.5) is asymptotically stable.

4.1.2 Case II: Time-Varying Delay

In this subsection, the asynchronous yet bounded time-varying delay is considered in each communication channel. Thus, the communication delay can be expressed as $\tau_{ij}(t) = \tau + \delta_{\tau_{ij}}(t)$, where τ is the nominal delay which is a known constant, and $\delta_{\tau_{ij}}(t)$ is the time-varying portion which is bounded by the known constant d as $|\delta_{\tau_{ij}}(t)| \leq d$.

In the presence of time-varying delay, the controller (4.2) can be modified as

$$\begin{cases} u'_i = u'_{ia} + u'_{ib} \\ u'_{ia} = \mathbf{l}\mathbf{x}_i \\ u'_{ib} = -\mathbf{k} \sum\limits_{j \in \mathcal{N}_i} a_{ij}(\mathbf{x}_i(t-\tau) - \mathbf{x}_j(t-(\tau + \delta_{\tau_{ij}}))). \end{cases} \tag{4.7}$$

The closed loop dynamics with the controller (4.7) for system (4.1) can be written as

$$\dot{\mathbf{x}}_i = (A + B\mathbf{l})\mathbf{x}_i - B\mathbf{k} \sum_{j \in \mathcal{N}_i} a_{ij}(\mathbf{x}_i(t-\tau) - \mathbf{x}_j(t-(\tau + \delta_{\tau_{ij}}))). \tag{4.8}$$

Consider again the system error defined in (4.4), the following error dynamics can be obtained for the system under time-varying communication delays.

$$\begin{aligned} \dot{\mathbf{e}}_i = (A + B\mathbf{l})\mathbf{e}_i - B\mathbf{k} \sum_{j \in \mathcal{N}_i} a_{ij}(\mathbf{e}_i(t-\tau) - \mathbf{e}_j(t-\tau)) - B\mathbf{k} \sum_{j \in \mathcal{N}_i} a_{ij}\Delta_{ij} \\ - B\mathbf{k} \sum_{j \in \mathcal{N}_1} a_{1j}\mathbf{e}_j(t-\tau) + B\mathbf{k} \sum_{j \in \mathcal{N}_1} a_{1j}\Delta_{1j}. \end{aligned} \tag{4.9}$$

The overall system error dynamics can be derived as

$$\dot{\mathbf{e}} = \bar{A}\mathbf{e} + \bar{L}\mathbf{e}(t-\tau) + \tilde{L}\Delta, \tag{4.10}$$

where

$$\Delta_{ij} = \mathbf{x}_j(t-\tau) - \mathbf{x}_j(t-(\tau+\delta_{\tau_{ij}}))$$

$$\Delta = \begin{bmatrix} \Delta_{11}^T & \Delta_{12}^T & \cdots & \Delta_{1N}^T & \cdots & \cdots & \cdots & \cdots & \Delta_{N1}^T & \Delta_{N2}^T & \cdots & \Delta_{NN}^T \end{bmatrix}^T$$

$$\tilde{L} = L_0 \otimes (-B\mathbf{k})$$

$$L_0 = \begin{bmatrix} a_{11} & \cdots & a_{1N} & -a_{21} & \cdots & -a_{2N} & 0 & \cdots & 0 & \cdots & & \\ a_{11} & \cdots & a_{1N} & 0 & \cdots & 0 & -a_{31} & \cdots & -a_{3N} & \cdots & & \\ & \vdots & & & \vdots & & & \vdots & & \cdots & & \\ a_{11} & \cdots & a_{1N} & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & & \\ & & & & & & & \cdots & 0 & \cdots & 0 & \\ & & & & & & & \cdots & 0 & \cdots & 0 & \\ & & & & & & & \cdots & & \vdots & & \\ & & & & & & & \cdots & -a_{N1} & \cdots & -a_{NN} & \end{bmatrix}.$$

The system (4.1) achieves consensus with bounded error under the controller (4.7) if the system (4.10) is stable.

4.2 Main Results

The stability analysis is conducted for both overall system error dynamics (4.5) and (4.10) in this section. The Lyapunov stability concept along with the LMI are utilized to provide the sufficient conditions for asymptotic consensus in the constant delay case and consensus with bounded error in the time-varying delay case.

4.2.1 Case I: Controller Design Under Constant Delays

Theorem 1. *In the presence of constant delay τ in each communication channel, the system (4.1) achieves consensus asymptotically with the control law (4.2) if for given positive scalars θ_i, $i \in 1, \cdots, 5$, there exist symmetric positive definite matrices $\hat{P}_{11}$, $\hat{P}_{22}$, $\hat{Q}_{11}$, $\hat{Q}_{22}$, $\hat{R}_{11}$, $\hat{R}_{22}$, as well as matrices $\hat{N}_i$, $i \in 1, \cdots, 5$ and Y, and a*

non-singular matrix $\hat{X}$ *such that*

$$\hat{H}_i = \begin{bmatrix} \hat{H}_{i_{11}} & * & * & * & * \\ \hat{H}_{i_{21}} & \hat{H}_{i_{22}} & * & * & * \\ \hat{H}_{i_{31}} & \hat{H}_{i_{32}} & \hat{H}_{i_{33}} & * & * \\ \hat{H}_{i_{41}} & \hat{H}_{i_{42}} & \hat{H}_{i_{43}} & \hat{H}_{i_{44}} & * \\ \hat{H}_{i_{51}} & \hat{H}_{i_{52}} & \hat{H}_{i_{53}} & \hat{H}_{i_{54}} & \hat{H}_{i_{55}} \end{bmatrix} < 0, \;\; i \in 1, \cdots, N-1, \tag{4.11}$$

where

$$\begin{aligned}
\hat{H}_{i_{11}} &= (N-2)\hat{Q}_{11} + \tau(N-2)\hat{R}_{11} + \hat{N}_1 + \hat{N}_1^T + \theta_1(A+B\mathbf{l})\hat{X}^T + \theta_1\hat{X}(A+B\mathbf{l})^T \\
\hat{H}_{i_{21}} &= -\hat{N}_1^T + \hat{N}_2 + \theta_2(A+B\mathbf{l})\hat{X}^T + \theta_1\lambda_i(Y^TB^T) \\
\hat{H}_{i_{22}} &= -(N-2)\hat{Q}_{11} - \hat{N}_2 - \hat{N}_2^T + \theta_2\lambda_i(BY) + \theta_2\lambda_i(Y^TB^T) \\
\hat{H}_{i_{31}} &= (N-2)\hat{P}_{11} + \hat{N}_3 - \theta_1\hat{X} + \theta_3(A+B\mathbf{l})\hat{X}^T \\
\hat{H}_{i_{32}} &= -\hat{N}_3 - \theta_2\hat{X} + \theta_3\lambda_i(BY) \\
\hat{H}_{i_{33}} &= (N-2)\hat{Q}_{22} + \tau(N-2)\hat{R}_{22} - \theta_3\hat{X}^T - \theta_3\hat{X} \\
\hat{H}_{i_{41}} &= (N-2)\hat{P}_{22} + \hat{N}_4 + \theta_4(A+B\mathbf{l})\hat{X}^T \\
\hat{H}_{i_{42}} &= -(N-2)\hat{P}_{22} - \hat{N}_4 + \theta_4\lambda_i(BY) \\
\hat{H}_{i_{43}} &= -\theta_4\hat{X}^T \\
\hat{H}_{i_{44}} &= -\frac{N-2}{\tau}\hat{R}_{11} \\
\hat{H}_{i_{51}} &= \hat{N}_5 - \hat{N}_1^T + \theta_5(A+B\mathbf{l})\hat{X}^T \\
\hat{H}_{i_{52}} &= -\hat{N}_5 - \hat{N}_2^T + \theta_5\lambda_i(BY) \\
\hat{H}_{i_{53}} &= -\hat{N}_3^T - \theta_5\hat{X}^T \\
\hat{H}_{i_{54}} &= -\hat{N}_4^T \\
\hat{H}_{i_{55}} &= -\hat{N}_5 - \frac{N-2}{\tau}\hat{R}_{22} - \hat{N}_5^T,
\end{aligned}$$

and λ_i *is the* i^{th} *eigenvalue of the matrix* L *defined in (4.6) in ascending order. Then, the static controller gain* $\mathbf{k}$ *can be computed as* $Y(\hat{X}^T)^{-1}$.

Proof. Consider a Lyapunov function candidate for the i^{th} component of error dynamics (4.5) of the following form, similarly as in [64] (refer to the Lyapunov-based

method in Section 2.2.2)

$$
\begin{aligned}
V_i = &\sum_{j=2,j\neq i}^{N} \begin{bmatrix} \mathbf{e}_i^T & \int_{t-\tau}^{t} \mathbf{e}_j^T(s)ds \end{bmatrix} \begin{bmatrix} P_{11} & \\ & P_{22} \end{bmatrix} \begin{bmatrix} \mathbf{e}_i \\ \int_{t-\tau}^{t} \mathbf{e}_j(s)ds \end{bmatrix} \\
&+ \sum_{j=2,j\neq i}^{N} \int_{t-\tau}^{t} \begin{bmatrix} \mathbf{e}_j^T(s) & \dot{\mathbf{e}}_j^T(s) \end{bmatrix} \begin{bmatrix} Q_{11} & \\ & Q_{22} \end{bmatrix} \begin{bmatrix} \mathbf{e}_i(s) \\ \dot{\mathbf{e}}_j(s) \end{bmatrix} ds \\
&+ \sum_{j=2,j\neq i}^{N} \int_{-\tau}^{0} \int_{t+r}^{t} \begin{bmatrix} \mathbf{e}_j(s)^T & \dot{\mathbf{e}}_j^T(s) \end{bmatrix} \begin{bmatrix} R_{11} & \\ & R_{22} \end{bmatrix} \begin{bmatrix} \mathbf{e}_j(s) \\ \dot{\mathbf{e}}_j(s) \end{bmatrix} dsdr
\end{aligned}, \tag{4.12}
$$

where $P_{11}, P_{22}, Q_{11}, Q_{22}, R_{11}, R_{22}$ are all symmetric positive definite matrices. The Lyapunov function candidate for the overall error dynamics is then given as

$$
\begin{aligned}
V &= \sum_{i=2}^{N} V_i \\
&= (N-2)\mathbf{e}^T \bar{P}_{11} \mathbf{e} + (N-2) \int_{t-\tau}^{t} \mathbf{e}^T(s)ds \bar{P}_{22} \int_{t-\tau}^{t} \mathbf{e}(s)ds \\
&\quad + (N-2) \int_{t-\tau}^{t} \begin{bmatrix} \mathbf{e}^T(s) & \dot{\mathbf{e}}^T(s) \end{bmatrix} \begin{bmatrix} \bar{Q}_{11} & \\ & \bar{Q}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{e}(s) \\ \dot{\mathbf{e}}(s) \end{bmatrix} ds \\
&\quad + (N-2) \int_{-\tau}^{0} \int_{t+r}^{t} \begin{bmatrix} \mathbf{e}^T(s) & \dot{\mathbf{e}}^T(s) \end{bmatrix} \begin{bmatrix} \bar{R}_{11} & \\ & \bar{R}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{e}(s) \\ \dot{\mathbf{e}}(s) \end{bmatrix} dsdr,
\end{aligned} \tag{4.13}
$$

where

$$
\begin{aligned}
&\bar{P}_{11} = \mathbf{I}_{N-1} \otimes P_{11},\ \bar{Q}_{11} = \mathbf{I}_{N-1} \otimes Q_{11},\ \bar{R}_{11} = \mathbf{I}_{N-1} \otimes R_{11}, \\
&\bar{P}_{22} = \mathbf{I}_{N-1} \otimes P_{22},\ \bar{Q}_{22} = \mathbf{I}_{N-1} \otimes Q_{22},\ \bar{R}_{22} = \mathbf{I}_{N-1} \otimes R_{22}.
\end{aligned}
$$

Taking the time derivative of (4.13) results in (4.14).

$$
\dot{V} = \Xi - (N-2) \int_{t-\tau}^{t} \begin{bmatrix} \mathbf{e}^T(s) & \dot{\mathbf{e}}^T(s) \end{bmatrix} \begin{bmatrix} \bar{R}_{11} & \\ & \bar{R}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{e}(s) \\ \dot{\mathbf{e}}(s) \end{bmatrix} ds, \tag{4.14}
$$

where

$$\begin{aligned}\Xi =&(N-2)\dot{\mathbf{e}}^T(t)\bar{P}_{11}\mathbf{e}(t)+(N-2)\mathbf{e}^T(t)\bar{P}_{11}\dot{\mathbf{e}}(t)\\&+(N-2)\mathbf{e}^T(t)\bar{P}_{22}\int_{t-\tau}^{t}\mathbf{e}(s)ds-(N-2)\mathbf{e}^T(t-\tau)\bar{P}_{22}\int_{t-\tau}^{t}\mathbf{e}(s)ds\\&+(N-2)\int_{t-\tau}^{t}\mathbf{e}^T(s)ds\bar{P}_{22}\mathbf{e}(t)-(N-2)\int_{t-\tau}^{t}\mathbf{e}^T(s)ds\bar{P}_{22}\mathbf{e}(t-\tau)\\&+(N-2)\mathbf{e}^T(t)\bar{Q}_{11}\mathbf{e}(t)-(N-2)\mathbf{e}^T(t-\tau)\bar{Q}_{11}\mathbf{e}(t-\tau)\\&+(N-2)\dot{\mathbf{e}}^T(t)\bar{Q}_{22}\dot{\mathbf{e}}(t)-(N-2)\dot{\mathbf{e}}^T(t-\tau)\bar{Q}_{22}\dot{\mathbf{e}}(t-\tau)\\&+\tau(N-2)\mathbf{e}^T(t)\bar{R}_{11}\mathbf{e}(t)+\tau(N-2)\dot{\mathbf{e}}^T(t)\bar{R}_{22}\dot{\mathbf{e}}(t).\end{aligned}$$

From Lemma 2 and (4.14), the following inequality can be obtained.

$$\begin{aligned}\dot{V}\leq\Xi&-\frac{N-2}{\tau}\int_{t-\tau}^{t}\mathbf{e}^T(s)ds\bar{R}_{11}\int_{t-\tau}^{t}\mathbf{e}(s)ds\\&-\frac{N-2}{\tau}\int_{t-\tau}^{t}\dot{\mathbf{e}}^T(s)ds\bar{R}_{22}\int_{t-\tau}^{t}\dot{\mathbf{e}}(s)ds+\phi_1+\phi_2,\end{aligned}\tag{4.15}$$

where ϕ_1 is a zero equation that increases the system's design flexibility, and ϕ_2 is another zero equation that introduces the controller gain $\mathbf{k}$ into the inequality. ϕ_1 and ϕ_2 are given as shown below.

$$\begin{aligned}\phi_1&=2Z^TN\left[\mathbf{e}(t)-\int_{t-\tau}^{t}\dot{\mathbf{e}}(s)ds-\mathbf{e}(t-\tau)\right],\\\phi_2&=2Z^TM\left[\bar{L}\mathbf{e}(t-\tau)+\bar{A}\mathbf{e}(t)-\dot{\mathbf{e}}(t)\right],\end{aligned}$$

where

$$\begin{aligned}Z&=\left[\mathbf{e}^T(t)\quad\mathbf{e}^T(t-\tau)\quad\dot{\mathbf{e}}^T(t)\quad\textstyle\int_{t-\tau}^{t}\mathbf{e}^T(s)ds\quad\int_{t-\tau}^{t}\dot{\mathbf{e}}^T(s)ds\right]^T,\\N&=\left[\bar{N}_1^T\quad\bar{N}_2^T\quad\bar{N}_3^T\quad\bar{N}_4^T\quad\bar{N}_5^T\right]^T,\\M&=\left[\bar{M}_1^T\quad\bar{M}_2^T\quad\bar{M}_3^T\quad\bar{M}_4^T\quad\bar{M}_5^T\right]^T,\\\bar{N}_i&=\mathbf{I}_{N-1}\otimes N_i,\\\bar{M}_i&=\mathbf{I}_{N-1}\otimes M_i,\end{aligned}$$

and $N_i, M_i, i\in 1,\cdots,5$, are arbitrary matrices.

The inequality (4.15) can be rewritten in a more compact way as

$$\dot{V}\leq Z^THZ,\tag{4.16}$$

where H is given as

$$H = \begin{bmatrix} H_{11} & * & * & * & * \\ H_{21} & H_{22} & * & * & * \\ H_{31} & H_{32} & H_{33} & * & * \\ H_{41} & H_{42} & H_{43} & H_{44} & * \\ H_{51} & H_{52} & H_{53} & H_{54} & H_{55} \end{bmatrix}, \tag{4.17}$$

with

$$\begin{aligned}
H_{11} &= (N-2)\bar{Q}_{11} + \tau(N-2)\bar{R}_{11} + \bar{N}_1 + \bar{N}_1^T + \bar{M}_1\bar{A} + \bar{A}^T\bar{M}_1^T \\
H_{21} &= -\bar{N}_1^T + \bar{N}_2 + \bar{M}_2\bar{A} + \bar{L}^T\bar{M}_1^T \\
H_{22} &= -(N-2)\bar{Q}_{11} - \bar{N}_2 - \bar{N}_2^T + \bar{M}_2\bar{L} + \bar{L}^T\bar{M}_2^T \\
H_{31} &= (N-2)\bar{P}_{11} + \bar{N}_3 - \bar{M}_1^T + \bar{M}_3\bar{A} \\
H_{32} &= -\bar{N}_3 + \bar{M}_3\bar{L} - \bar{M}_2^T \\
H_{33} &= (N-2)\bar{Q}_{22} + \tau(N-2)\bar{R}_{22} - \bar{M}_3 - \bar{M}_3^T \\
H_{41} &= (N-2)\bar{P}_{22} + \bar{N}_4 + \bar{M}_4\bar{A} \\
H_{42} &= -(N-2)\bar{P}_{22} - \bar{N}_4 + \bar{M}_4\bar{L} \\
H_{43} &= -\bar{M}_4 \\
H_{44} &= -\frac{N-2}{\tau}\bar{R}_{11} \\
H_{51} &= \bar{N}_5 - \bar{N}_1^T + \bar{M}_5\bar{A} \\
H_{52} &= -\bar{N}_5 - \bar{N}_2^T + \bar{M}_5\bar{L} \\
H_{53} &= -\bar{N}_3^T - \bar{M}_5 \\
H_{54} &= -\bar{N}_4^T \\
H_{55} &= -\bar{N}_5 - \frac{N-2}{\tau}\bar{R}_{22} - \bar{N}_5^T.
\end{aligned}$$

The control objective will be achieved if there exists a feedback gain $\mathbf{k}$ such that $H < 0$, thus $\dot{V} < 0$. However, the matrix H has to be linearized, thus more conservative, in order to solve the controller gain $\mathbf{k}$ using Matlab LMI toolbox. The linearization is done by imposing the scaling factor θ_i on $\bar{M}_i$ as $\bar{M}_i = \theta_i\bar{M}_3$, $i \in 1, \cdots 5$ and θ_3 is defined as 1. We also impose the positive definiteness on $\bar{M}_3$, thus $\bar{M}_3 > 0$, and

let $X = \bar{M}_3^{-1}$ and $\tilde{W} = \mathbf{I}_5 \otimes X$. The congruence transformation of H is given as $\tilde{H} = \tilde{W} H \tilde{W}^T$. Since $\tilde{W}$ is invertible, $H < 0 \Leftrightarrow \tilde{H} < 0$, where $\tilde{H}$ is given as

$$\tilde{H} = \begin{bmatrix} \tilde{H}_{11} & * & * & * & * \\ \tilde{H}_{21} & \tilde{H}_{22} & * & * & * \\ \tilde{H}_{31} & \tilde{H}_{32} & \tilde{H}_{33} & * & * \\ \tilde{H}_{41} & \tilde{H}_{42} & \tilde{H}_{43} & \tilde{H}_{44} & * \\ \tilde{H}_{51} & \tilde{H}_{52} & \tilde{H}_{53} & \tilde{H}_{54} & \tilde{H}_{55} \end{bmatrix}, \tag{4.18}$$

with

$$\begin{aligned}
\tilde{H}_{11} &= (N-2)\tilde{Q}_{11} + \tau(N-2)\tilde{R}_{11} + \tilde{N}_1 + \tilde{N}_1^T + \theta_1 \bar{A} X^T + \theta_1 X \bar{A}^T \\
\tilde{H}_{21} &= -\tilde{N}_1^T + \tilde{N}_2 + \theta_2 \bar{A} X^T + \theta_1 X \bar{L}^T \\
\tilde{H}_{22} &= -(N-2)\tilde{Q}_{11} - \tilde{N}_2 - \tilde{N}_2^T + \theta_2 \bar{L} X^T + \theta_2 X \bar{L}^T \\
\tilde{H}_{31} &= (N-2)\tilde{P}_{11} + \tilde{N}_3 - \theta_1 X + \theta_3 \bar{A} X^T \\
\tilde{H}_{32} &= -\tilde{N}_3 + \theta_3 \bar{L} X^T - \theta_2 X \\
\tilde{H}_{33} &= (N-2)\tilde{Q}_{22} + \tau(N-2)\tilde{R}_{22} - \theta_3 X^T - \theta_3 X \\
\tilde{H}_{41} &= (N-2)\tilde{P}_{22} + \tilde{N}_4 + \theta_4 \bar{A} X^T \\
\tilde{H}_{42} &= -(N-2)\tilde{P}_{22} - \tilde{N}_4 + \theta_4 \bar{L} X^T \\
\tilde{H}_{43} &= -\theta_4 X^T \\
\tilde{H}_{44} &= -\frac{N-2}{\tau}\tilde{R}_{11} \\
\tilde{H}_{51} &= \tilde{N}_5 - \tilde{N}_1^T + \theta_5 \bar{A} X^T \\
\tilde{H}_{52} &= -\tilde{N}_5 - \tilde{N}_2^T + \theta_5 \bar{L} X^T \\
\tilde{H}_{53} &= -\tilde{N}_3^T - \theta_5 X^T \\
\tilde{H}_{54} &= -\tilde{N}_4^T \\
\tilde{H}_{55} &= -\tilde{N}_5 - \frac{N-2}{\tau}\tilde{R}_{22} - \tilde{N}_5^T,
\end{aligned}$$

Note that $\tilde{P}_{11} = X \bar{P}_{11} X^T$. Since X, $\bar{P}_{11}$ are block diagonal, $\tilde{P}_{11}$ is also block diagonal and can be expressed as $\mathbf{I}_{N-1} \otimes \hat{P}_{11}$, where $\hat{P}_{11} = \hat{X} P_{11} \hat{X}^T$. The following expressions can be obtained

$$\begin{aligned}
&\tilde{P}_{22} = \mathbf{I}_{N-1} \otimes \hat{P}_{22},\ \tilde{Q}_{11} = \mathbf{I}_{N-1} \otimes \hat{Q}_{11},\ \tilde{Q}_{22} = \mathbf{I}_{N-1} \otimes \hat{Q}_{22}, \\
&\tilde{R}_{11} = \mathbf{I}_{N-1} \otimes \hat{R}_{11},\ \tilde{R}_{22} = \mathbf{I}_{N-1} \otimes \hat{R}_{22},\ \tilde{N}_i = \mathbf{I}_{N-1} \otimes \hat{N}_i, i \in 1, \cdots, 5.
\end{aligned}$$

In addition, $X\bar{L}^T = L^T \otimes (Y^T B^T)$, where Y is defined as $\mathbf{k}\hat{X}^T$. Therefore, the block entries in (4.18) can be rewritten as

$$
\begin{aligned}
\tilde{H}_{11} &= \mathbf{I}_{N-1} \otimes \left[(N-2)\hat{Q}_{11} + \tau(N-2)\hat{R}_{11} + \hat{N}_1 + \hat{N}_1^T\right] \\
&\qquad + \mathbf{I}_N \otimes \left[\theta_1(A+B\mathbf{l})\hat{X}^T + \theta_1\hat{X}(A+B\mathbf{l})^T\right] \\
\tilde{H}_{21} &= \mathbf{I}_{N-1} \otimes \left[-\hat{N}_1^T + \hat{N}_2 + \theta_2(A+B\mathbf{l})\hat{X}^T\right] + \theta_1 L^T \otimes (Y^T B^T) \\
\tilde{H}_{22} &= \mathbf{I}_{N-1} \otimes \left[-(N-2)\hat{Q}_{11} - \hat{N}_2 - \hat{N}_2^T\right] + \theta_2 L \otimes (BY) + \theta_2 L^T \otimes (Y^T B^T) \\
\tilde{H}_{31} &= \mathbf{I}_{N-1} \otimes \left[(N-2)\hat{P}_{11} + \hat{N}_3 - \theta_1\hat{X} + \theta_3(A+B\mathbf{l})\hat{X}^T\right] \\
\tilde{H}_{32} &= \mathbf{I}_{N-1} \otimes \left[-\hat{N}_3 - \theta_2\hat{X}\right] + \theta_3 L \otimes (BY) \\
\tilde{H}_{33} &= \mathbf{I}_{N-1} \otimes \left[(N-2)\hat{Q}_{22} + \tau(N-2)\hat{R}_{22} - \theta_3\hat{X}^T - \theta_3\hat{X}\right] \\
\tilde{H}_{41} &= \mathbf{I}_{N-1} \otimes \left[(N-2)\hat{P}_{22} + \hat{N}_4 + \theta_4(A+B\mathbf{l})\hat{X}^T\right] \\
\tilde{H}_{42} &= \mathbf{I}_{N-1} \otimes \left[-(N-2)\hat{P}_{22} - \hat{N}_4\right] + \theta_4 L \otimes (BY) \\
\tilde{H}_{43} &= \mathbf{I}_{N-1} \otimes \left[-\theta_4\hat{X}^T\right] \\
\tilde{H}_{44} &= \mathbf{I}_{N-1} \otimes \left[-\frac{N-2}{\tau}\hat{R}_{11}\right] \\
\tilde{H}_{51} &= \mathbf{I}_{N-1} \otimes \left[\hat{N}_5 - \hat{N}_1^T + \theta_5(A+B\mathbf{l})\hat{X}^T\right] \\
\tilde{H}_{52} &= \mathbf{I}_{N-1} \otimes \left[-\hat{N}_5 - \hat{N}_2^T\right] + \theta_5 L \otimes (BY) \\
\tilde{H}_{53} &= \mathbf{I}_{N-1} \otimes \left[-\hat{N}_3^T - \theta_5\hat{X}^T\right] \\
\tilde{H}_{54} &= \mathbf{I}_{N-1} \otimes \left[-\hat{N}_4^T\right] \\
\tilde{H}_{55} &= \mathbf{I}_{N-1} \otimes \left[-\hat{N}_5 - \frac{N-2}{\tau}\hat{R}_{22} - \hat{N}_5^T\right].
\end{aligned}
$$

The complexity of linear matrix inequality $\tilde{H} < 0$ increases significantly, thus it is difficult to solve numerically, as the number of agents increases. It is necessary to derive equivalent low dimension conditions that can be practically used. Under the assumption that the group of agents communicate to each other in a way such that the matrix L is symmetric, the equivalent low dimension LMIs can be derived from the congruence transformation $\hat{H} = \hat{W}\tilde{H}\hat{W}^T$, where $\hat{W} = \mathbf{I}_5 \otimes (U \otimes \mathbf{I}_n)$ and $U^T = U^{-1}$ as the orthogonal matrix from the diagonalization of $L = U^T \Lambda U$, and Λ is the eigenvalue matrix with eigenvalues in ascending order. It also can be shown that $H < 0 \Leftrightarrow \hat{H} < 0 \Leftrightarrow \hat{H}_i < 0,\ i \in 1, \cdots, N-1$ and $\hat{H}_i$ is given in (4.11). $\square$

Remark. *It can be shown, according to the convexity property of LMI, $\hat{H}_i < 0$, $i \in 1, \cdots, N-1$ is equivalent to $\hat{H}_1 < 0$ and $\hat{H}_{N-1} < 0$. In addition, it can be observed from its diagonal entry that the inequality (4.11) is less feasible given larger number of agents and delay. Besides, θ and ϕ with subscripts in this chapter represent the arbitrary design parameters and zero equations correspondingly, whereas θ and ϕ without subscripts denote the pitch and roll angles respectively.*

4.2.2 Case II:Controller Design Under Time-Varying Delays

Theorem 2. *In the presence of asynchronous time-varying delays $\tau_{\delta_{ij}}$ in each communication channel, the system (4.1) achieves consensus with bounded error under control law (4.7) if for given positive scalars γ, τ, θ_i, $i \in 1, \cdots, 5$, there exist symmetric positive definite matrices $\hat{P}_{11}$, $\hat{P}_{22}$, $\hat{Q}_{11}$, $\hat{Q}_{22}$, $\hat{R}_{11}$, $\hat{R}_{22}$, $\hat{S}$ as well as matrices $\hat{N}_i$, $i \in 1, \cdots, 5$ and Y, and a non-singular matrix $\hat{X}$ such that*

$$\hat{\Phi}_i = \begin{bmatrix} \hat{H}_{i_{11}} + \hat{S} & * & * & * & * & * \\ \hat{H}_{i_{21}} & \hat{H}_{i_{22}} & * & * & * & * \\ \hat{H}_{i_{31}} & \hat{H}_{i_{32}} & \hat{H}_{i_{33}} & * & * & * \\ \hat{H}_{i_{41}} & \hat{H}_{i_{42}} & \hat{H}_{i_{43}} & \hat{H}_{i_{44}} & * & * \\ \hat{H}_{i_{51}} & \hat{H}_{i_{52}} & \hat{H}_{i_{53}} & \hat{H}_{i_{54}} & \hat{H}_{i_{55}} & * \\ \theta_1 \hat{X} & \theta_2 \hat{X} & \theta_3 \hat{X} & \theta_4 \hat{X} & \theta_5 \hat{X} & -\frac{1}{\gamma}\mathbf{I}_n \end{bmatrix} < 0, \;\; i \in 1, \cdots, N-1, \tag{4.19}$$

where $\hat{H}_{i_{ij}}$ is the same as given in (4.11), and λ_i is the i^{th} eigenvalue of matrix L in ascending order. Then, the static controller gain $\mathbf{k}$ can be computed as $Y(\hat{X}^T)^{-1}$. In addition, the consensus error will be bounded by

$$\|\mathbf{e}(t)\| \leq \frac{\gamma^{-1}\lambda_{max}(\tilde{L}^T\tilde{L})\|\Delta\|}{\lambda_{min}(\hat{S})}.$$

Proof. Consider the same Lyapunov function candidate (4.13) as in the constant delay case. Following the same procedures in the previous subsection, the Lyapunov function candidate's time derivative satisfies the following inequality

$$\begin{aligned} \dot{V} \leq \Xi &- \frac{N-2}{\tau}\int_{t-\tau}^{t} \mathbf{e}^T(s)ds\bar{R}_{11}\int_{t-\tau}^{t}\mathbf{e}(s)ds \\ &- \frac{N-2}{\tau}\int_{t-\tau}^{t} \dot{\mathbf{e}}^T(s)ds\bar{R}_{22}\int_{t-\tau}^{t}\dot{\mathbf{e}}(s)ds + \phi_1 + \phi_2 + \phi_3, \end{aligned} \tag{4.20}$$

where, ϕ_1 is same as given in (4.15), ϕ_2 incorporates the error dynamics (4.10), and the newly introduced zero equation ϕ_3 serves for compressing the consensus error.

$$\phi_2 = 2Z^T M \left[\bar{L}\mathbf{e}(t-\tau) + \bar{A}\mathbf{e}(t) - \dot{\mathbf{e}}(t) + \tilde{L}\Delta\right] \tag{4.21}$$

$$\phi_3 = \mathbf{e}^T\bar{S}\mathbf{e} - \mathbf{e}^T\bar{S}\mathbf{e}, \tag{4.22}$$

where $\bar{S} = \mathbf{I}_N \otimes S$, and $S = S^T > 0$.
Using the following quadratic inequality

$$2Z^T M\tilde{L}\Delta \leq \gamma Z^T MM^T Z + \gamma^{-1}\Delta^T\tilde{L}^T\tilde{L}\Delta, \tag{4.23}$$

the inequality (4.20) is changed to

$$\dot{V} \leq Z^T H' Z \quad + \gamma Z^T MM^T Z + \gamma^{-1}\Delta^T\tilde{L}^T\tilde{L}\Delta - \mathbf{e}^T\bar{S}\mathbf{e}, \tag{4.24}$$

where

$$H' = \begin{bmatrix} H_{11}+\bar{S} & * & * & * & * \\ H_{21} & H_{22} & * & * & * \\ H_{31} & H_{32} & H_{33} & * & * \\ H_{41} & H_{42} & H_{43} & H_{44} & * \\ H_{51} & H_{52} & H_{53} & H_{54} & H_{55} \end{bmatrix},$$

and H_{ij} is the same as given in (4.17). Using Lemma 1, $H' + \lambda MM^T < 0$ is equivalent to

$$\Phi = \begin{bmatrix} H_{11}+\bar{S} & * & * & * & * & * \\ H_{21} & H_{22} & * & * & * & * \\ H_{31} & H_{32} & H_{33} & * & * & * \\ H_{41} & H_{42} & H_{43} & H_{44} & * & * \\ H_{51} & H_{52} & H_{53} & H_{54} & H_{55} & * \\ \bar{M}_1^T & \bar{M}_2^T & \bar{M}_3^T & \bar{M}_4^T & \bar{M}_5^T & -\frac{1}{\lambda}\mathbf{I} \end{bmatrix} < 0, \tag{4.25}$$

If the inequality (4.25) is satisfied, then the following inequality holds.

$$\dot{V} \leq \gamma^{-1}\Delta^T\tilde{L}^T\tilde{L}\Delta - \mathbf{e}^T\bar{S}\mathbf{e}. \tag{4.26}$$

Besides, if $\mathbf{e}^T\bar{S}\mathbf{e} > \gamma^{-1}\Delta^T\tilde{L}^T\tilde{L}\Delta$, then

$$\dot{V}(t) \leq 0.$$

As a result, $\|\mathbf{e}(t)\|$ will be compressed, thus $\|\mathbf{e}(t)\|$ is expected to be bounded by

$$\|\mathbf{e}(t)\| \leq \sqrt{\frac{\gamma^{-1}\lambda_{max}(\tilde{L}^T\tilde{L})\|\Delta\|^2}{\lambda_{min}(\bar{S})}} \tag{4.27}$$

It can be shown, using the same linearization and diagonalization techniques described in the previous section, that $\Phi < 0 \Leftrightarrow \hat{\Phi}_i < 0$, $i \in 1, \cdots, N-1$ in (4.19). □

Remark. *Although larger $\lambda_{min}(\bar{S})$ and γ values can reduce the error bound, they tend to induce the infeasibility of (4.19). Thus, there is a trade-off between the feasibility and error bound. In addition, given the fixed $\lambda_{max}(\tilde{L}^T\tilde{L})$ and $\|\Delta\|$, the error bound is minimized when the product of γ and $\lambda_{min}(\bar{S})$ is maximized within certain feasible region.*

4.3 Summary

This chapter contains the stability analysis for consensus design of linear MASs with synchronous constant and asynchronous time-varying communication delays. The consensus problem is converted to a tracking problem by introducing the systems error dynamics. Sufficient LMI conditions are then derived based on Lyapunov's method for both cases. Similar procedures are used in deriving the LMI conditions except with the addition of a disturbance term, which is suppressed by introducing an extra zero equation, in the error dynamics of time-varying case. At last, the LMI conditions are all linearized and decomposed into a set of smaller conditions such that they can be efficiently solved.

Chapter 5

Simulation Results of the Constant Delay Case

This chapter describes simulation results of the system under synchronous constant communication delays among all communication channels. Discussions will be on the effects of the communication topologies, the number of agents, the constant delay and the four arbitrary design parameters θ_i, $i = 1, 2, 4, 5$ on the system performance. The leader-follower consensus structure is adopted for all simulation cases.

5.1 Controller Implementation

The quadcopter system has been transformed to one networked and one local double-integrator subsystems in (3.33) and (3.34), and two equivalent networked fourth-order-integrator subsystems (3.31) and (3.32). Thus, the controllers designed based on these transformed subsystems have to be inversely transformed so that they can be actually implemented to the original system.

The following PD controller is proposed for the local double-integrator system,

$$u_{4i} = -4\psi_i - 4\dot{\psi}_i. \tag{5.1}$$

The local feedback gain is designed as $\mathbf{l}_2 = \begin{bmatrix} 0 & -1 \end{bmatrix}$ for the networked double-integrator subsystem and $\mathbf{l}_4 = \begin{bmatrix} 0 & -6 & -11 & -6 \end{bmatrix}$ for the fourth-order-integrator subsystems given the corresponding desired poles of $\begin{bmatrix} 0 & -1 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 & -2 & -3 \end{bmatrix}$. The network interaction gains are then designed based on the feasibility test of the LMI conditions (4.11) for asymptotic consensus and (4.19) for the consensus with bounded error (see Theorems 1 and 2).

The transformed feedback controller $u'_{1i}, u'_{2i}, u'_{3i}$ can be identified from (4.2) once $\mathbf{l}$ and $\mathbf{k}$ are designed. The original control input u_{1i}, u_{2i}, u_{3i} from (3.31) to (3.33) can

then be obtained as follows

$$\begin{cases} u'_{1i} & = \mathbf{l}_2 \begin{bmatrix} z_i \\ \dot{z}_i \end{bmatrix} - \mathbf{k}_2 \sum_{j\in\mathcal{N}_i} a_{ij} \left(\begin{bmatrix} z_i(t-\tau) \\ \dot{z}_i(t-\tau) \end{bmatrix} - \begin{bmatrix} z_j(t-\tau) \\ \dot{z}_j(t-\tau) \end{bmatrix} \right), \\ u_{1i} & = \frac{u'_{1i}+g}{\cos(\phi_i)\cos(\theta_i)}, \end{cases} \tag{5.2}$$

$$\begin{cases} u'_{2i} & = \mathbf{l}_4 \begin{bmatrix} y_i \\ \dot{y}_i \\ w_i \\ \dot{w}_i \end{bmatrix} - \mathbf{k}_4 \sum_{j\in\mathcal{N}_i} a_{ij} \left(\begin{bmatrix} y_i(t-\tau) \\ \dot{y}_i(t-\tau) \\ w_i(t-\tau) \\ \dot{w}_i(t-\tau) \end{bmatrix} - \begin{bmatrix} y_j(t-\tau) \\ \dot{y}_j(t-\tau) \\ w_j(t-\tau) \\ \dot{w}_j(t-\tau) \end{bmatrix} \right), \\ u_{2i} & = \frac{u'_{2i}-2\frac{\dot{w}^2 w}{g^2+w^2}}{-(g+\frac{w^2}{g})}, \end{cases} \tag{5.3}$$

$$\begin{cases} u'_{3i} & = \mathbf{l}_4 \begin{bmatrix} x_i \\ \dot{x}_i \\ v_i \\ \dot{v}_i \end{bmatrix} - \mathbf{k}_4 \sum_{j\in\mathcal{N}_i} a_{ij} \left(\begin{bmatrix} x_i(t-\tau) \\ \dot{x}_i(t-\tau) \\ v_i(t-\tau) \\ \dot{v}_i(t-\tau) \end{bmatrix} - \begin{bmatrix} x_j(t-\tau) \\ \dot{x}_j(t-\tau) \\ v_j(t-\tau) \\ \dot{v}_j(t-\tau) \end{bmatrix} \right), \\ u_{3i} & = \frac{u'_{3i}-2\frac{\dot{v}^2 v}{g^2+v^2}}{g+\frac{v^2}{g}}, \end{cases} \tag{5.4}$$

where $\mathbf{l}_2$ and $\mathbf{k}_2$ denote the local feedback and network interaction gains for the double-integrator subsystem and $\mathbf{l}_4$ and $\mathbf{k}_4$ for the fourth-order-integrator subsystems. In the time-varying delay case, the neighbor's states under time-varying delays are used (e.g. $z_j(t-\tau)$ is replaced by $z_j(t-\tau_{ij}(t))$ for j^{th} agent sending z_j to the i^{th} agent).

5.2 Effect of Communication Topologies

This section discusses the effect of communication topologies in terms of the maximum allowable communication delay (τ_{max}) which indicates the robustness of the system subject to constant communication delays, and consensus time (t_c) that characterizes the speed that followers can converge to the leader. τ_{max} is defined as the upper threshold of the constant delay that the consensusability of the system can be guaranteed from the feasibility test of (4.11). t_c is the required time for the consensus error $e_c(t)$ defined in (5.5), which is the average distance between the leader and

followers, to reach an established cutoff value.

$$e_c(t) = \frac{\sum_{i=2}^{N} \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2 + (z_i - z_1)^2}}{N - 1} \tag{5.5}$$

The complete set of communication topology cases for a four-agents system that satisfy the assumption of symmetric L matrix defined in (4.6) is given in Figs. 5.1 to 5.3. The arbitrary design parameters θ_i, $i = 1, 2, 3, 4, 5$ are all set as 1. τ_{max}

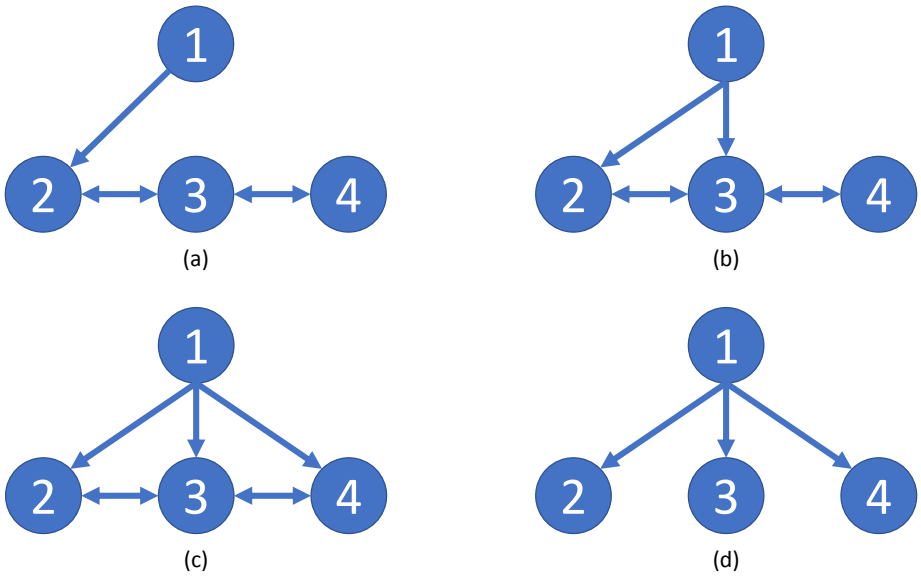

Figure 5.1: Communication topology cases (a), (b), (c) and (d)

associated with each communication topology is summarized in Fig. 5.4.

In evaluating t_c, all the followers are assumed to be initially hovering at different locations while the leader is stationary at the origin and initial conditions are given in Table 5.1. Followers will then converge to the leader under designed control laws (5.1) to (5.4). For instance, the position profile for the group of agents with the proposed control algorithm under communication topology case(f) and constant delay $\tau = 0.5$ s is shown in Fig. 5.5, and the translational response is given as Fig. 5.6, angular response as Fig. 5.7 and the consensus error as Fig. 5.8. The network feedback gains for this case are obtained as $\mathbf{k}_2 = \begin{bmatrix} 0.2284 & 0.3318 \end{bmatrix}$ and $\mathbf{k}_4 = \begin{bmatrix} 0.7886 & 1.4242 & 0.6958 & 0.1174 \end{bmatrix}$. t_c is calculated for each case under constant delay $\tau = 0.5$ s with a cutoff consensus error of 0.1 m and then summarized in Fig. 5.9.

There are three conclusions which can be drawn by observing and comparing

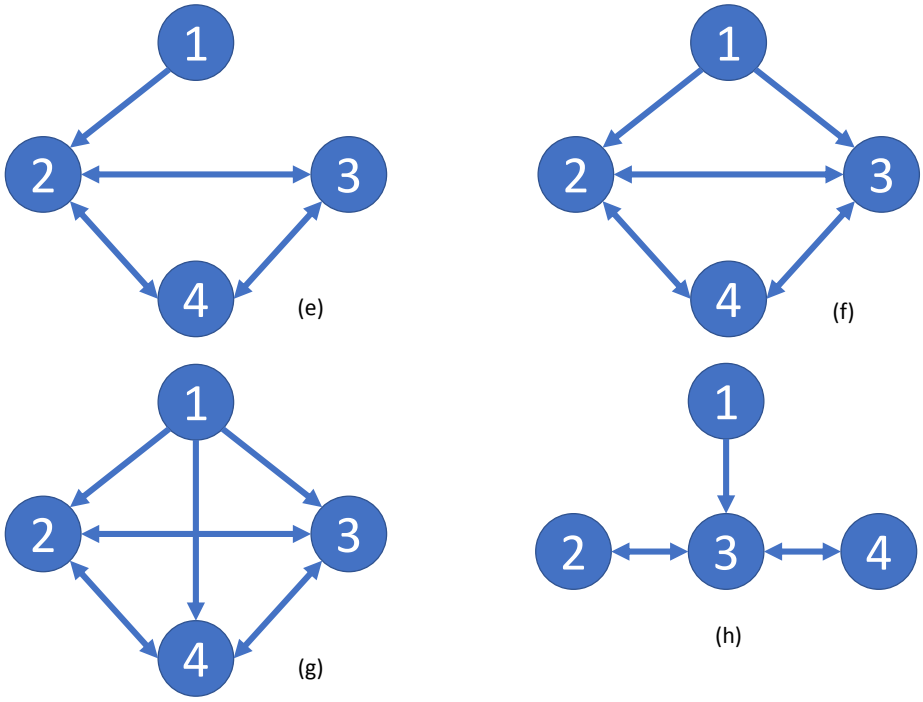

Figure 5.2: Communication topology cases (e), (f), (g) and (h)

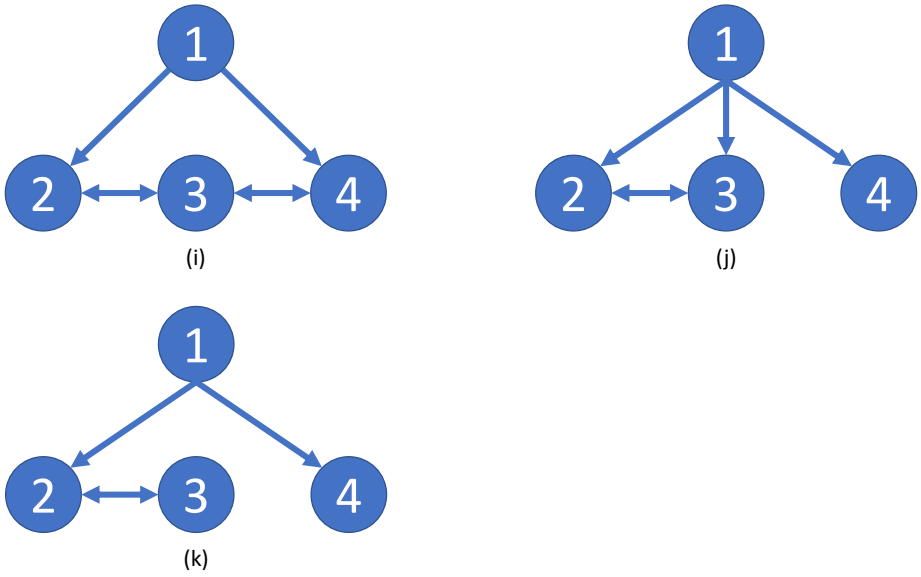

Figure 5.3: Communication topology cases (i), (j) and (k)

τ_{max} and t_c resulting from all communication topology cases. At first, the system can tolerate larger delay and achieve faster consensus with more followers directly receiving information from the leader given that the same communication topology between the followers is maintained. This conclusion can be verified from cases (a),

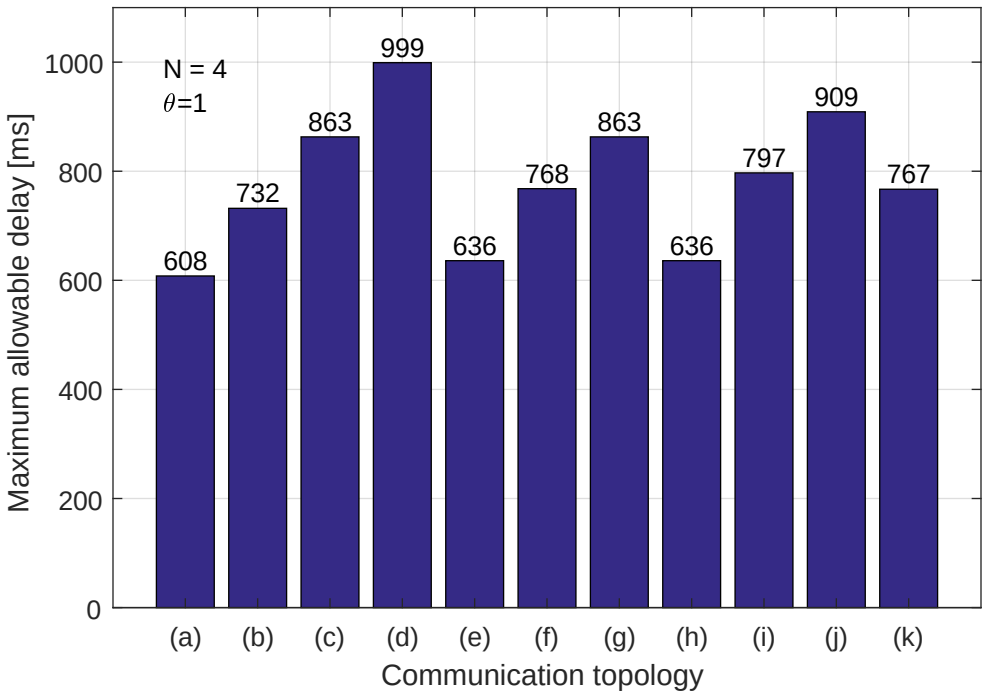

Figure 5.4: τ_{max} for cases (a) to (k)

Table 5.1: Initial conditions of 4-quadcopter system

Quadcopter	X(m)	Y(m)	Z(m)	Yaw(deg)	Pitch(deg)	Roll(deg)
1	0	0	0	0	0	0
2	10	0	10	60	0	0
3	0	10	10	90	0	0
4	0	-10	10	120	0	0

(b) and (c) in Figs. 5.4 and 5.9, where the same communication topology between the followers is maintained. In case (c), the leader sends information directly to all three followers, whereas only to two followers in (b) and one follower in (a). Consequently, the system has the maximum τ_{max} (863 ms) and minimum t_c (43.2 s). Similar results can be found for cases (e), (f) and (g). The τ_{max} and t_c results for cases (a), (b), (c), (e), (f) and (g) are summarized in Table 5.2.

Secondly, in the centralized form, where the leader directly sends information to every follower, adding communication links between the followers will not enhance the system performance. This conclusion can be verified from case (d), where there is no communication between followers, the system has the maximum τ_{max} (999 ms)

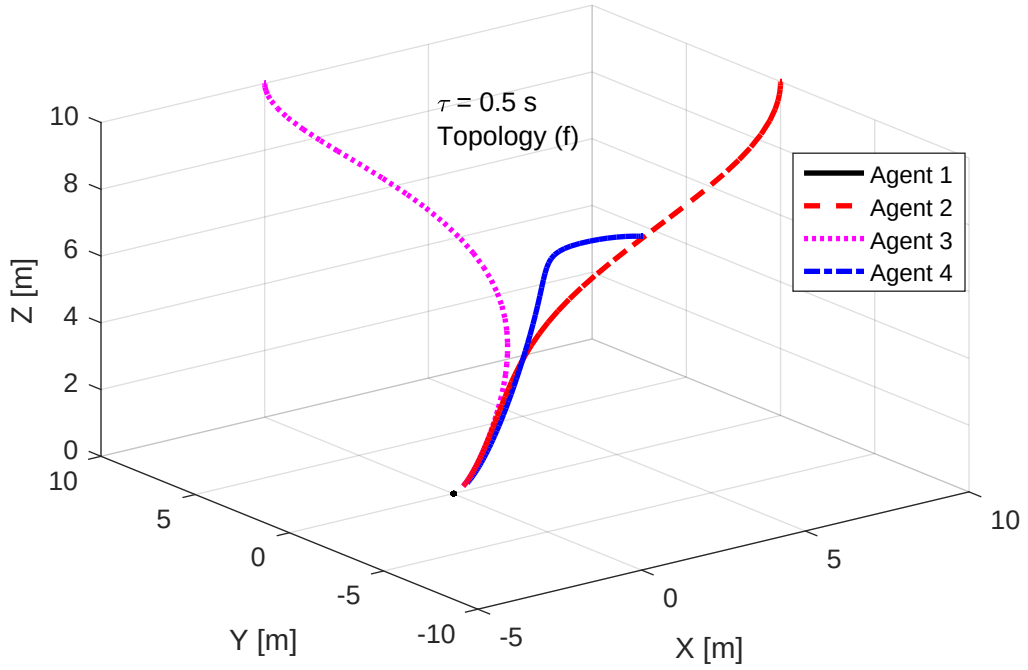

Figure 5.5: Position profile of the system under communication topology (f) and $\tau = 0.5$ s

Table 5.2: τ_{max} and t_c for cases (a), (b), (c), (e), (f) and (g)

Communication topology	(a)	(b)	(c)	(e)	(f)	(g)
τ_{max} (ms)	608	732	863	636	768	863
t_c (s)	61.0	46.0	43.2	60.8	44.7	42.7

and minimum t_c (27.7 s) among all centralized cases (c), (d), (g) and (j). Table 5.3 summarizes the results for the mentioned cases.

Thirdly, in the distributed form, in general, systems perform similarly regard-

Table 5.3: τ_{max} and t_c for cases (c), (d), (g) and (j)

Communication topology	(c)	(d)	(g)	(j)
τ_{max} (ms)	863	999	863	909
t_c (s)	43.2	27.7	42.7	50.9

less of connections between the followers. For example, τ_{max} and t_c values only vary slightly in cases (a), (e) and (h). Similar results can also be found in cases (b), (f), (i) and (k). Table 5.4 summarizes the results for the mentioned cases.

All in all, it seems that the centralized case (d) has the best performance. How-

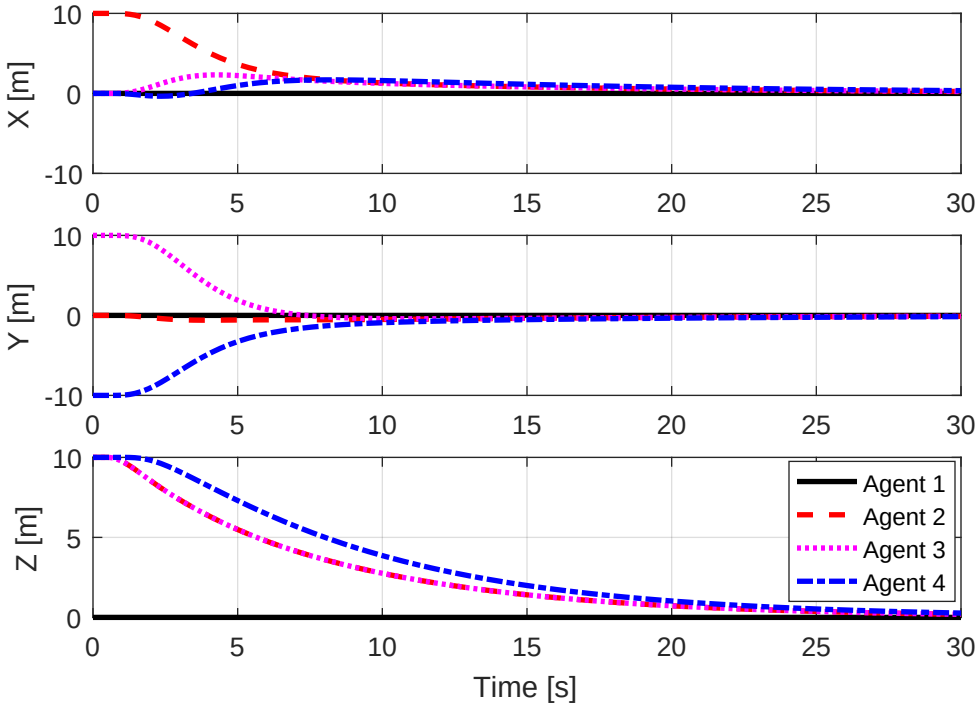

Figure 5.6: Translational response of the system under communication topology (f) and $\tau = 0.5$ s

Table 5.4: τ_{max} and t_c for cases (a), (e), (h), (b), (f), (i) and (k)

Communication topology	(a)	(e)	(h)	(b)	(f)	(i)	(k)
τ_{max} (ms)	608	636	636	732	768	797	767
t_c (s)	61.0	60.8	61.7	46.0	44.7	44.2	51.1

ever, it is the most susceptible one to communication link failure. For example, the consensus will never be reached if there is link failure in the system. Case (g), on the other hand, is the most robust case in terms of communication link failure.

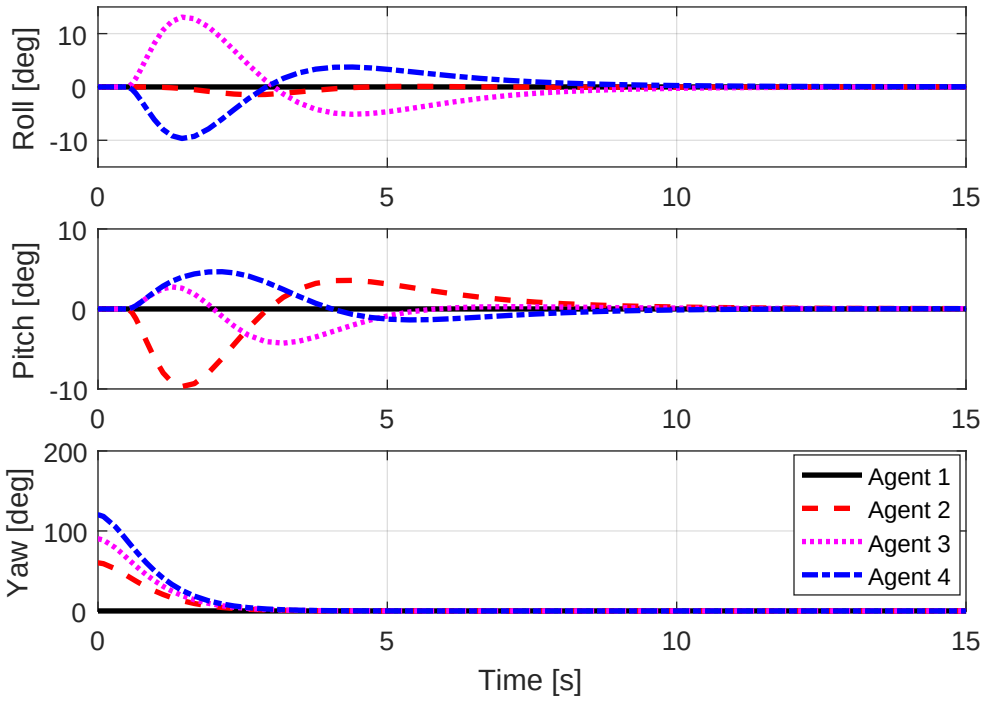

Figure 5.7: Angular response of the system under communication topology (f) and $\tau = 0.5$ s

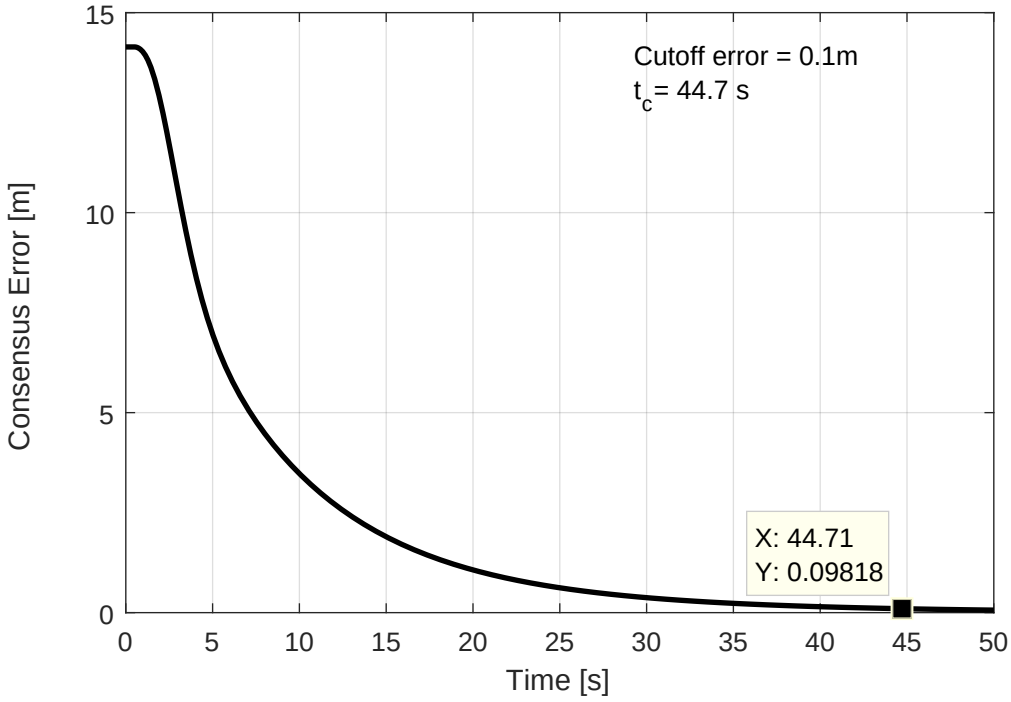

Figure 5.8: Consensus error of the system under communication topology (f) and $\tau = 0.5$ s

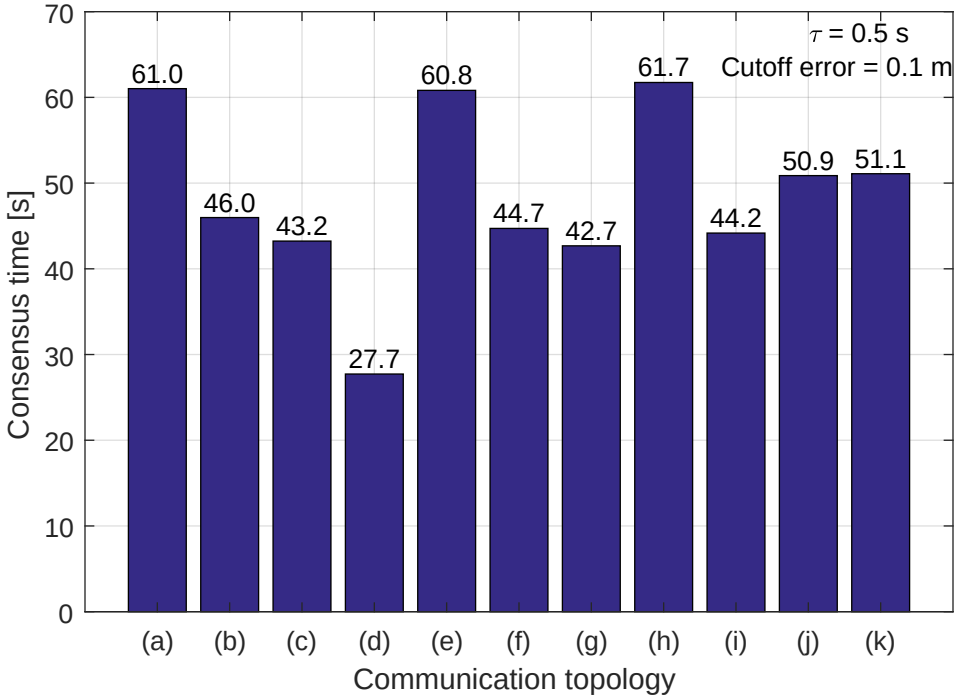

Figure 5.9: Consensus time associated with the communication topologies (a) to (k)

5.3 Effect of Number of Agents

This section discusses the effect of number of agents on τ_{max} and t_c. The ring type communication topology among the followers shown in Fig. 5.10 will be used in discussions because of its superior scalability. The parameters θ_i, $i = 1, \cdots, 5$ are all set as 1. τ_{max} for the systems with four to ten agents are summarized in Figure 5.11.

Again, for evaluating t_c, the leader is set as stationary at the origin and the followers are initially hovering at locations evenly distributed on a sphere with radius ρ of $10\sqrt{2}$. The initial conditions for followers are given in (5.6) with the angles defined in Fig. 5.12. The position profile for a group of six agents under constant communication delay of 0.3 s is shown in Fig. 5.13, the translational response is given as Fig. 5.14, and the angular response is shown in Fig. 5.15. Additionally, the consensus error profile is given in Fig. 5.16 and t_c is computed as 86.3 s with a cutoff error of 0.5 m. t_c for other cases are summarized in Fig. 5.17.

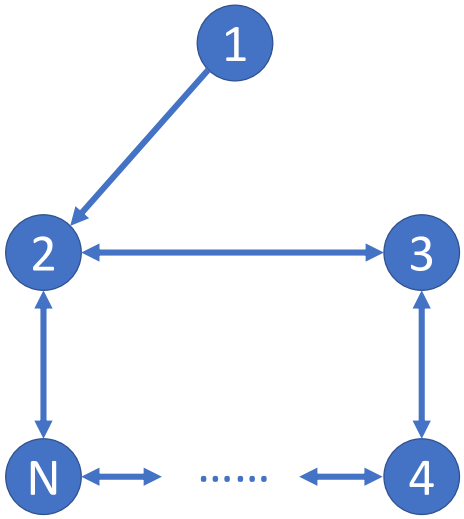

Figure 5.10: Ring type communication topology with N agents

$$\begin{cases} x_i &= \rho \sin(\phi_i)\cos(\theta_i) \\ y_i &= \rho \sin(\phi_i)\cos(\theta_i) \\ z_i &= \rho \cos(\phi_i) \\ \theta_i &= \frac{360^\circ}{N-1} i, \;\; \phi_i = 45^\circ, \;\; i = 1, \cdots, N-1. \end{cases} \tag{5.6}$$

The simulation results summarized in Figs. 5.11 and 5.17 indicate that delays

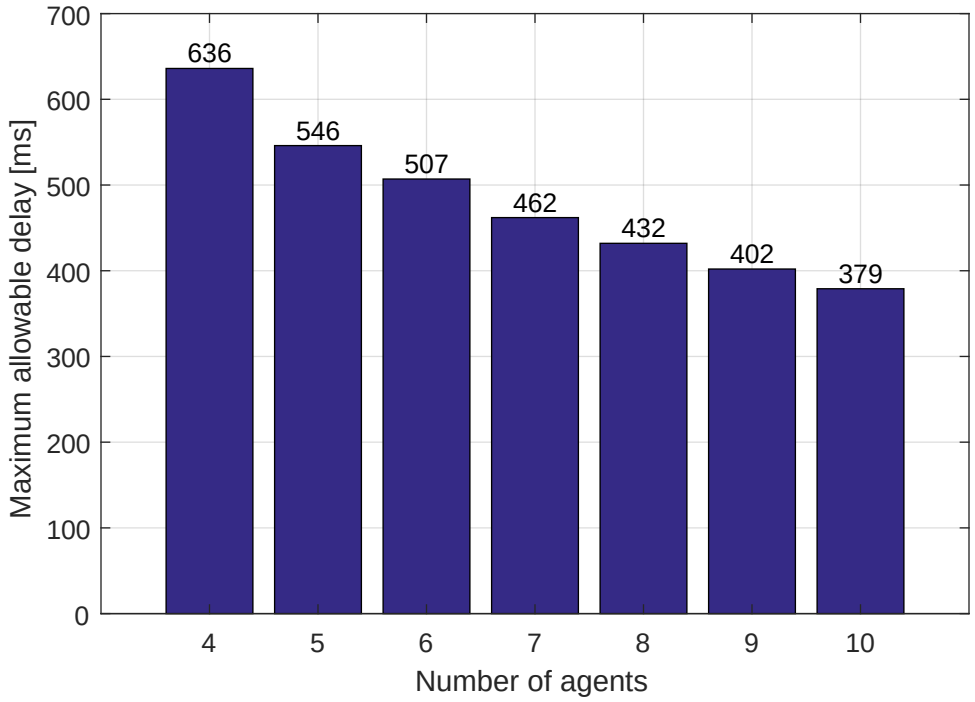

Figure 5.11: Maximum allowable delays of the ring type topology associated with four to ten agents

have more impact on the consensus speed when number of agents increases. Besides, it can be observed from Fig. 5.14 that t_c is mainly governed by motions in Z-direction since the system converges quickly in the X- and Y- directions within the first 15 seconds. This observation can be analyzed by examining the network feedback gains. For instance, consider the six-agent system, the network feedback gains are computed as $\mathbf{k}_2 = \begin{bmatrix} 0.2593 & 0.4279 \end{bmatrix}$ and $\mathbf{k}_4 = \begin{bmatrix} 1.1019 & 2.000 & 0.9778 & 0.1440 \end{bmatrix}$. The gain (0.2593) associated with the position error in the Z-direction is smaller than the gain (1.1019) in the X- and Y- directions. As a result, the system tends to converge faster in X- and Y- directions.

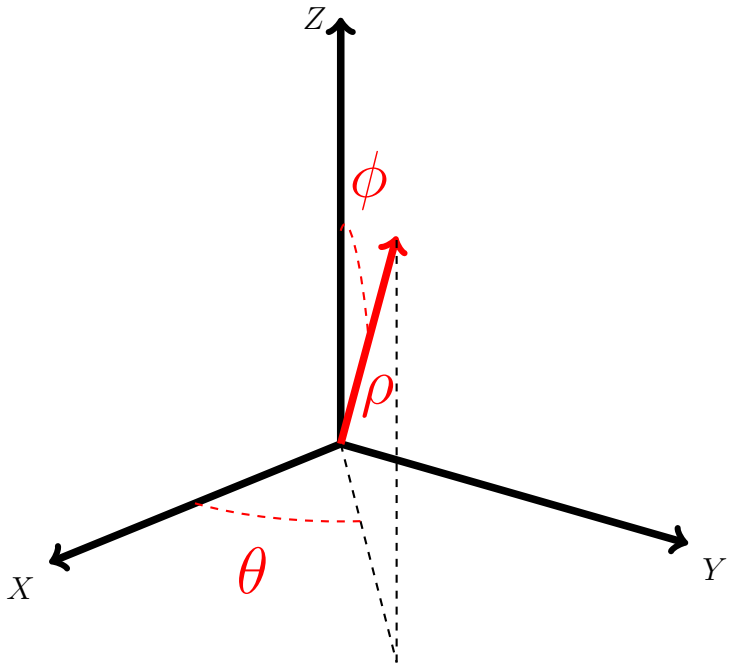

Figure 5.12: Spherical coordinates.

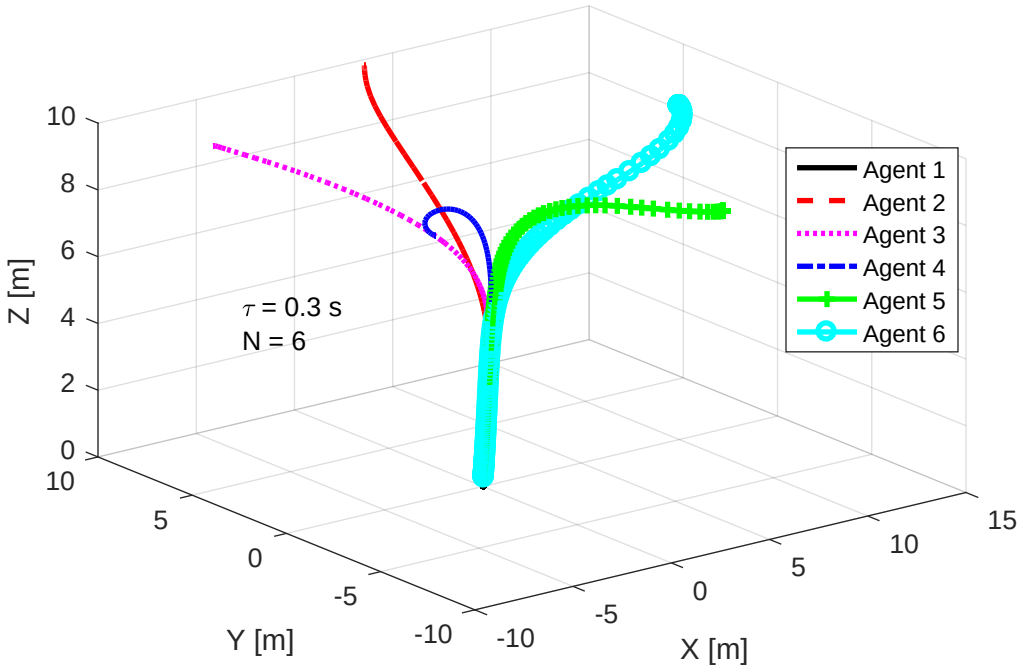

Figure 5.13: Position profile of the six-agent system under $\tau = 0.3$ s

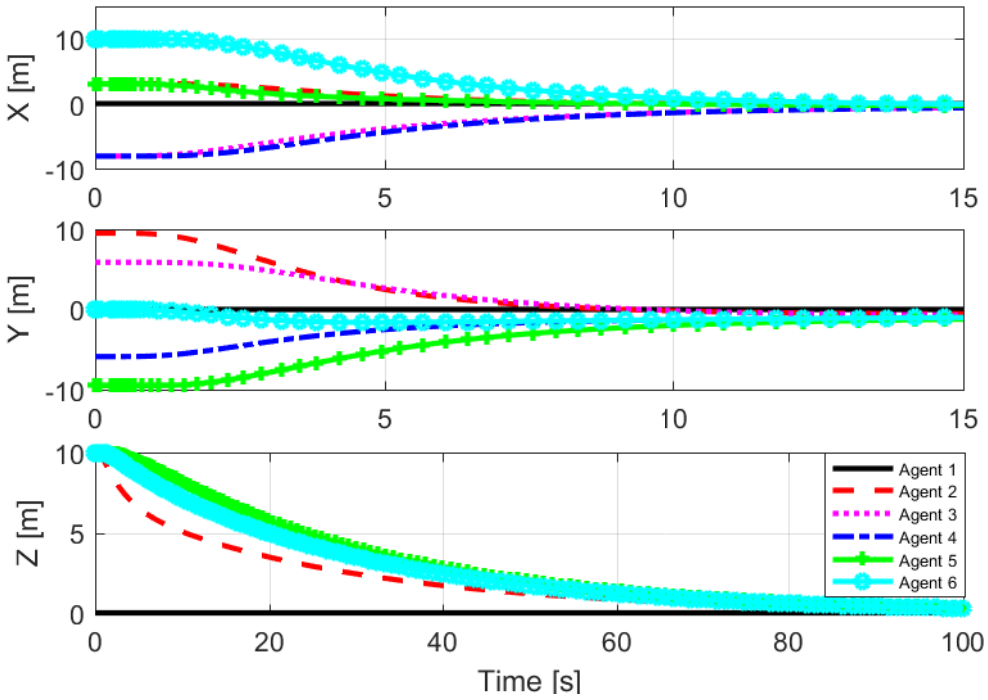

Figure 5.14: Translational responses of the system under $\tau = 0.3$ s

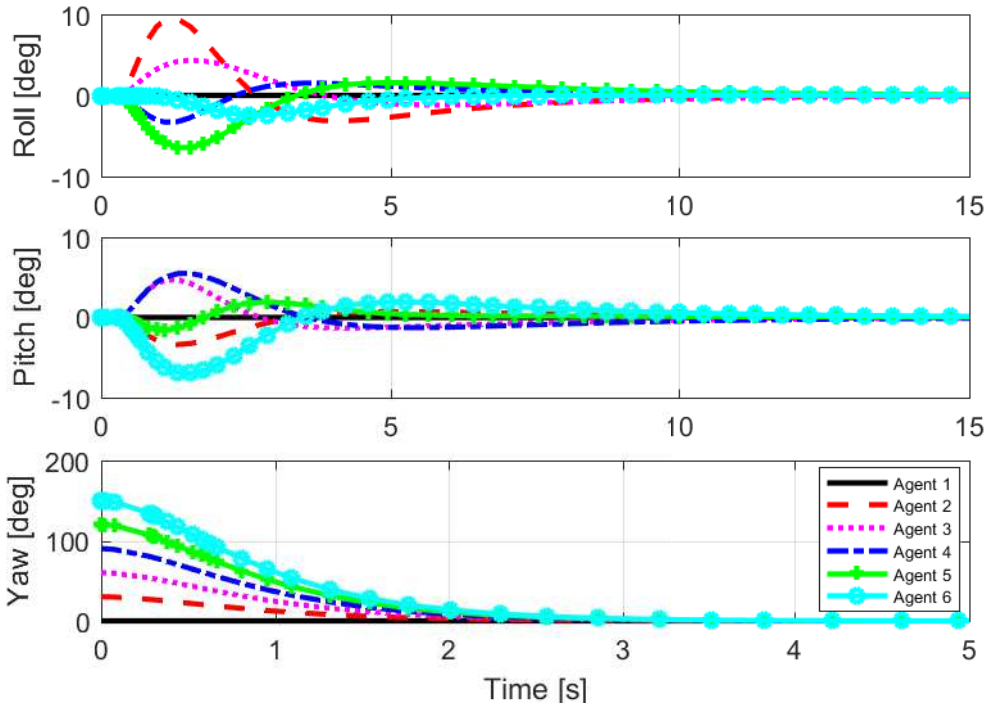

Figure 5.15: Angular response of the six-agent system under $\tau = 0.3$ s

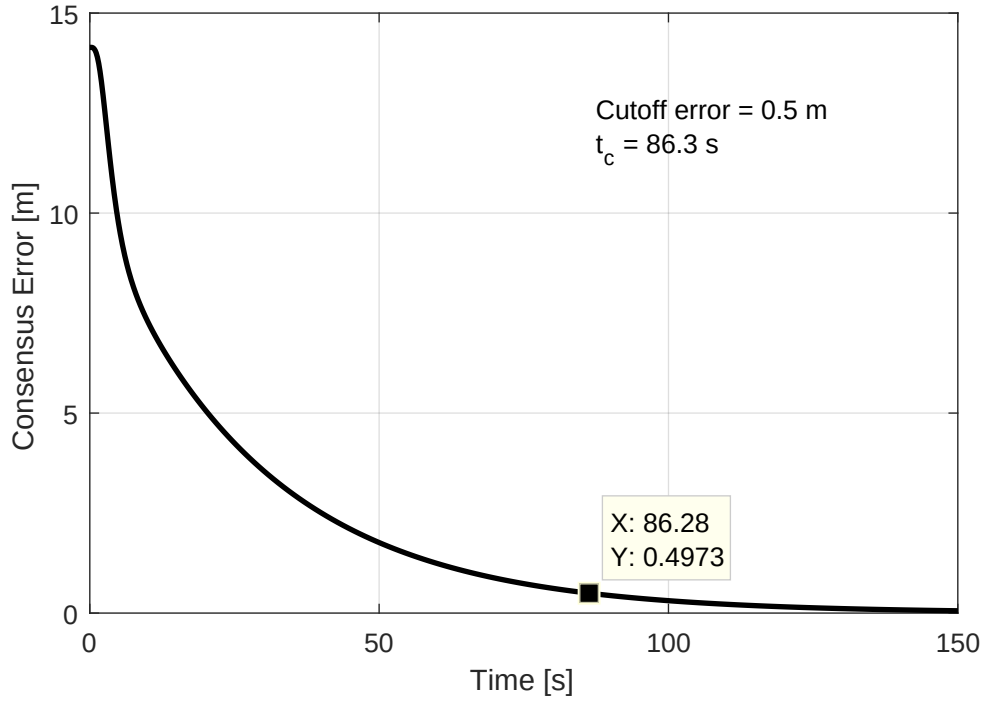

Figure 5.16: Consensus error of the six-agent system under $\tau = 0.3$ s

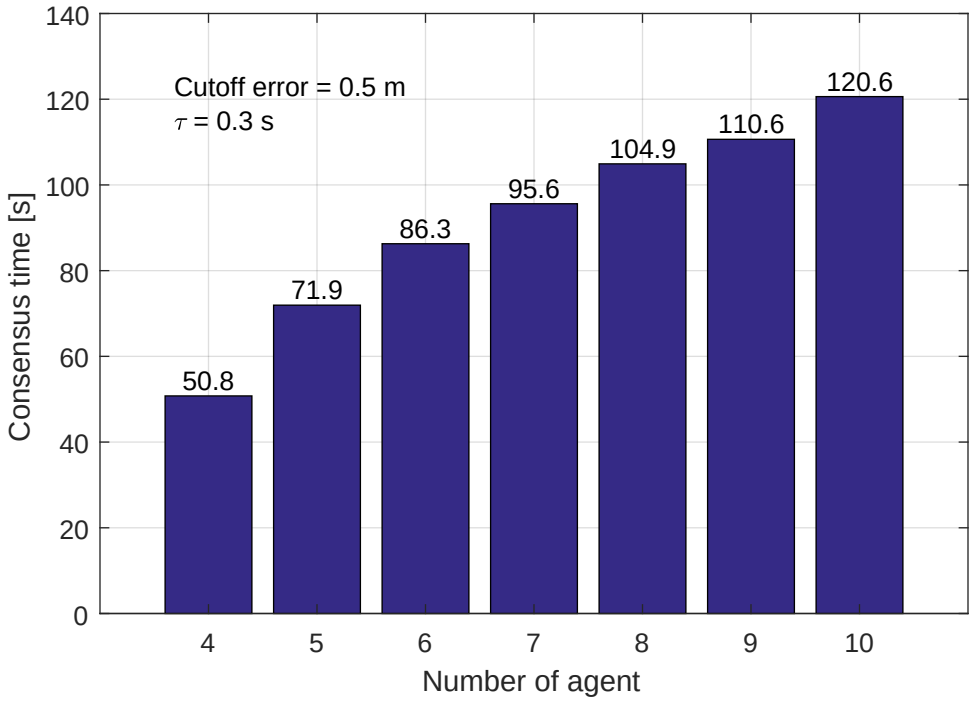

Figure 5.17: Consensus time of the ring type topology associated with four to ten agents under $\tau = 0.3$ s

5.4 Effect of Constant Delay

Two cases are studied regarding the effects of constant delays on the consensus time t_c in this section. Specifically, in the first case the controller is designed based on the actual delay that exists in the system, whereas in the second case the controller is designed based on a prescribed delay which is independent of the actual system delay. Communication topology (e) in Fig. 5.2 is chosen for all the simulations in this section. θ_i, $i = 1, \cdots, 5$ are set as 1, and the agents' initial conditions are given as (5.6).

5.4.1 Case I: Controller Designed Based on Actual System Delay

Consensus time t_c of the system with respect to a set of delays is summarized in Fig. 5.18. It is surprising that t_c reduces as the delay increases. This observation can be analyzed by examining the controller gains given in Table 5.5 where the position error gains (in bold) increase as the delays increase. It means that the feasible LMI in (4.11) condition actually tries to compensate for the effect of large delays by introducing large gains. Nevertheless, large controller gains imply large control effort which is often not realizable in practice. Furthermore, as shown in Fig. 5.19, large gains resulted from large delays also imply agile maneuvering, thus large roll angle response.

Table 5.5: Control gains with respect to delays (case I)

Delay τ (s)	$\mathbf{k}_2$	$\mathbf{k}_4$
0.1	**0.1833** 0.2801	**0.1294** 0.1066 −0.1865 −0.1471
0.2	**0.2486** 0.4212	**0.3340** 0.5799 0.1064 −0.0694
0.3	**0.2557** 0.4264	**0.5895** 1.0359 0.3839 0.0052
0.4	**0.2752** 0.4189	**1.0179** 1.8294 0.8860 0.1397
0.5	**0.2914** 0.4152	**1.5870** 2.9000 1.5395 0.2835
0.6	**0.3716** 0.4180	**2.2235** 4.0792 2.2243 0.3918

5.4.2 Case II: Controller Designed Based on Prescribed Delay

In this case the controller is designed based on the prescribed delay of 0.5 s and subsequently applied to the systems under various delays from 0.05 s to 0.6 s. The

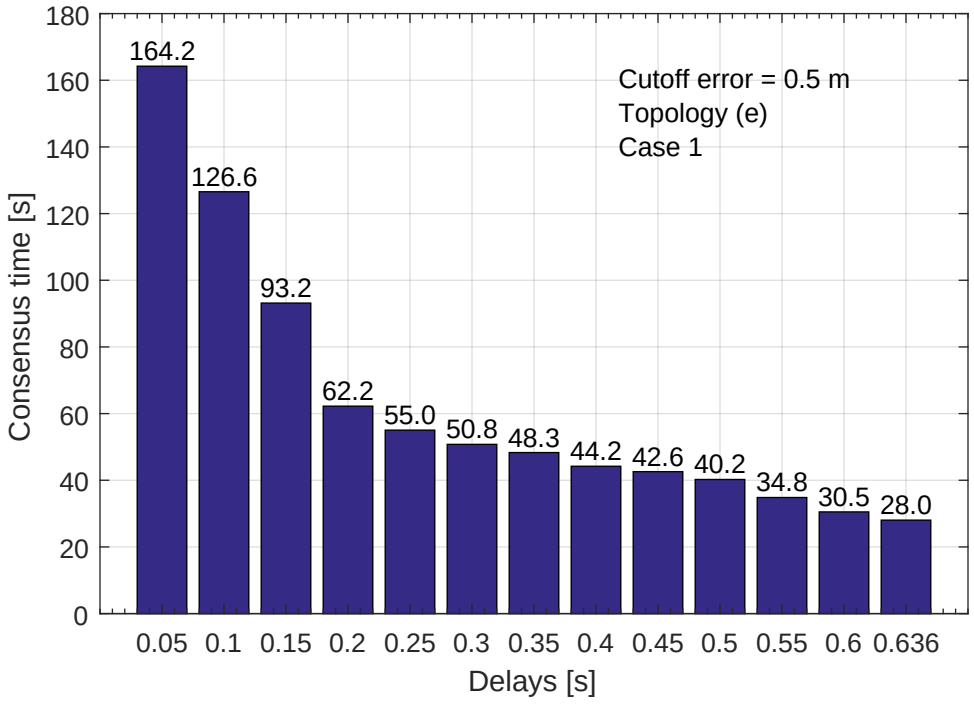

Figure 5.18: t_c under topology (e) with respect to delays (Case I)

simulation results given in Fig. 5.20 indicate that the systems converge at approximately same speed under such controller disregarding the actual delay in the system to certain magnitude. However, larger system delay tends to induce more oscillations. To illustrate, the roll responses for a delays of 0.5 s and 1 s are compared and given in Fig. 5.21, and pitch responses in Fig. 5.22. In addition, it has been manually tested that the simulation actually fails when the delay in the system exceeds 1.33 s and the consensus profile for the system with delay of 1.33 s is shown in Fig. 5.23. Similar results, as shown in Fig. 5.24, can be obtained for controllers designed at other prescribed delays.

Finally, the conclusions obtained from both cases with the system under communication topology (e) are also valid for systems under other topologies. As an illustration, the results hold, as shown in Figs. 5.25 and 5.26, for the system under communication topology (a).

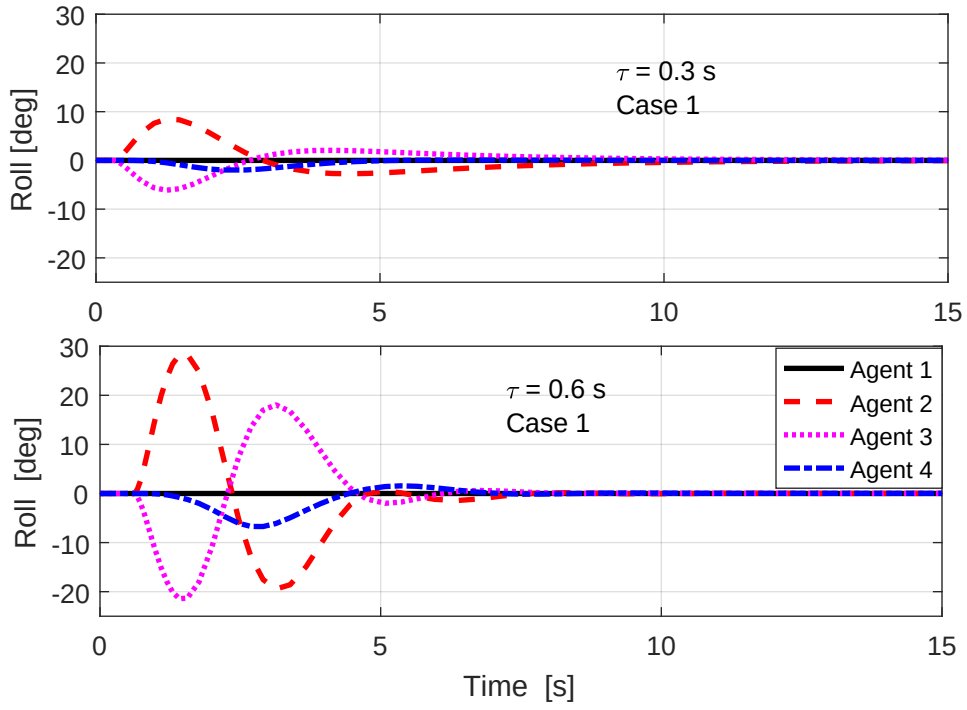

Figure 5.19: Roll angle responses with respect to delays (Case I)

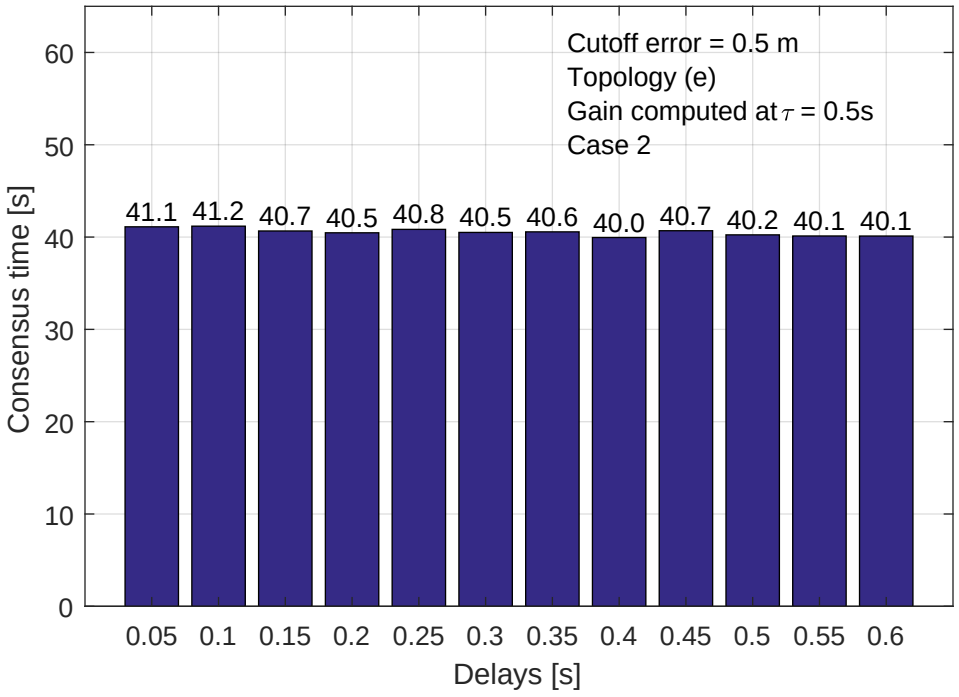

Figure 5.20: t_c under controller designed at $\tau = 0.5$ s (Case II)

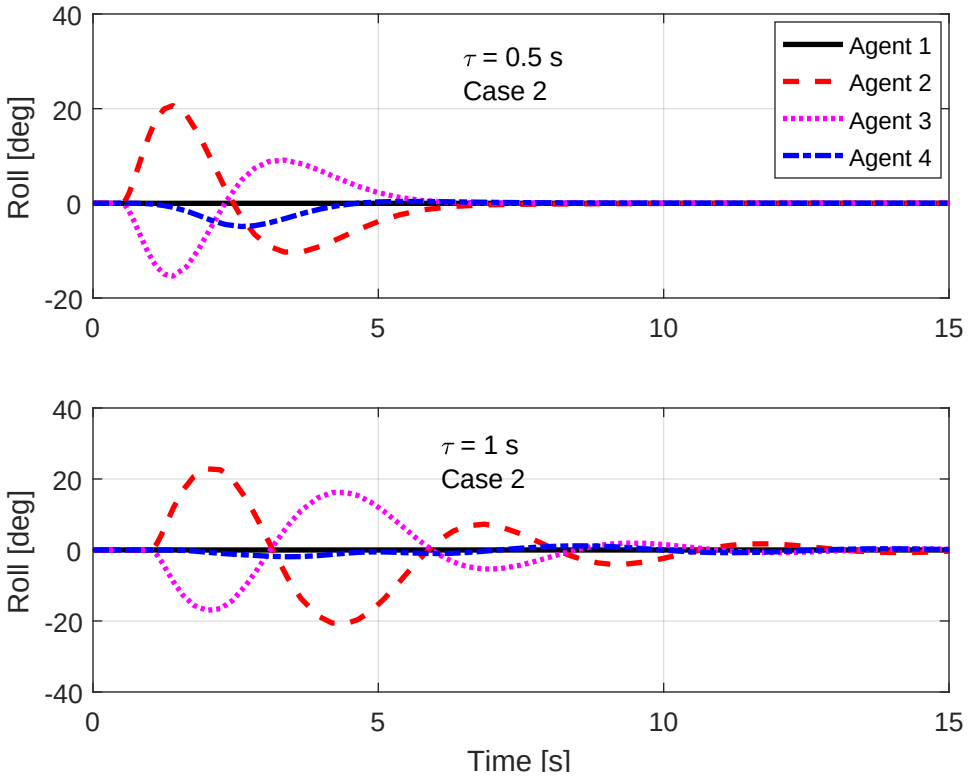

Figure 5.21: Roll responses at $\tau = 0.5\ s$ and $1\ s$ (Case II)

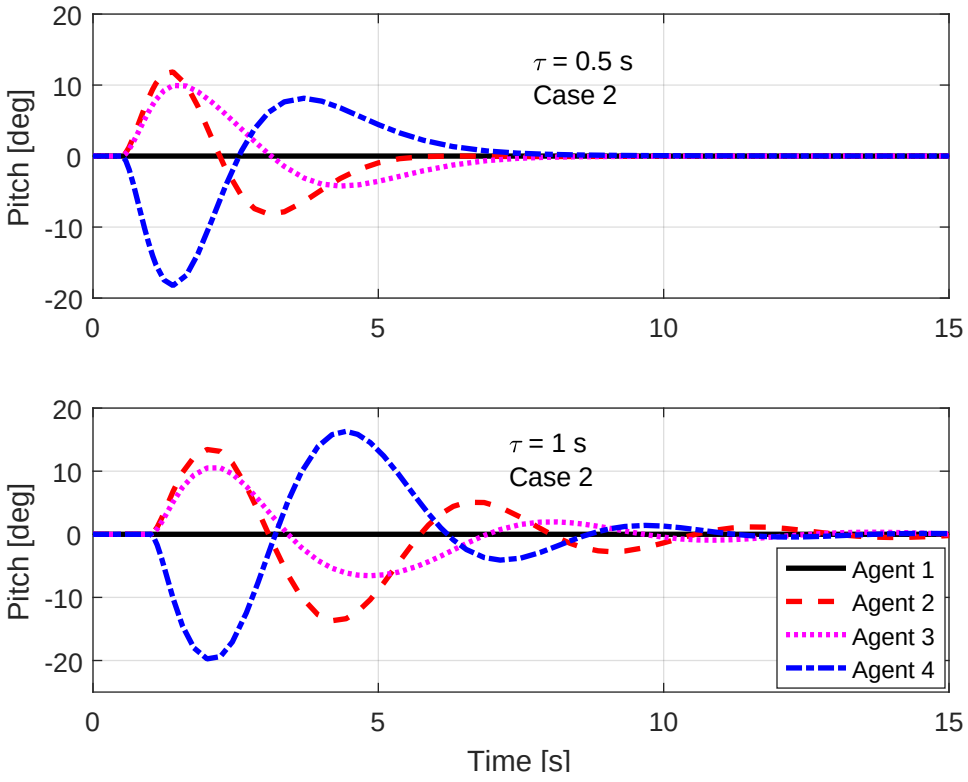

Figure 5.22: Pitch responses at $\tau = 0.5\ s$ and $1\ s$ (Case II)

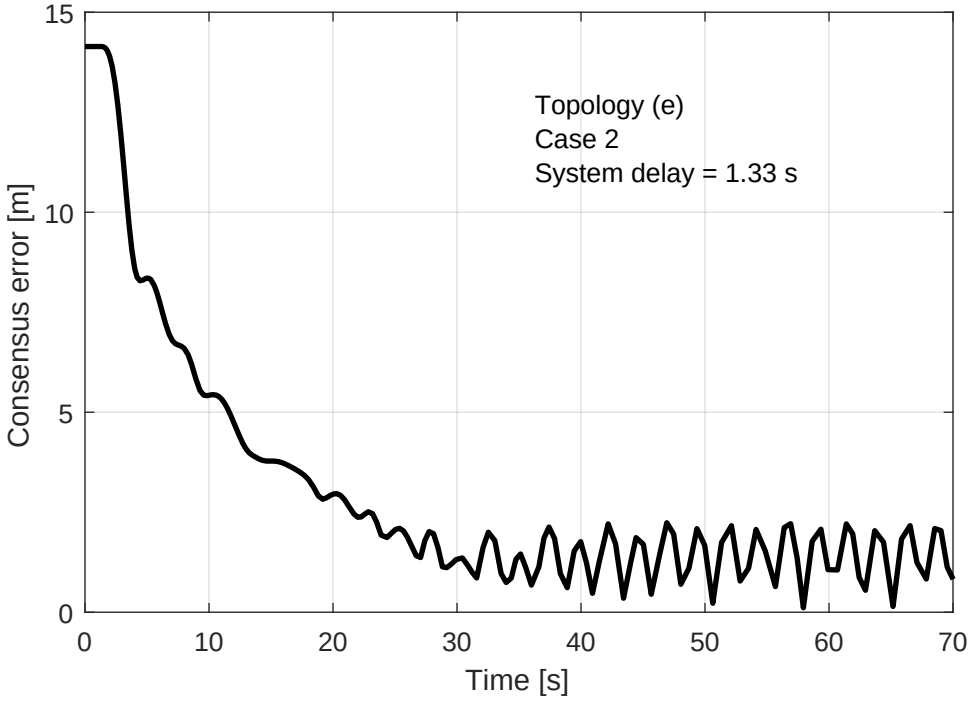

Figure 5.23: Consensus error profile under $\tau = 1.33\ s$ (Case II)

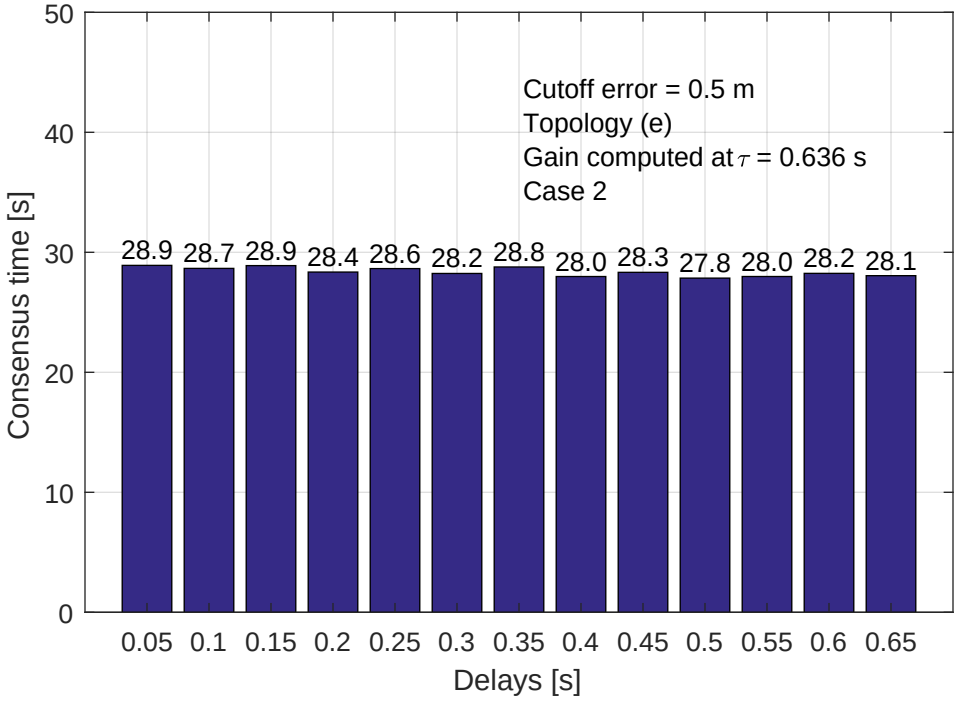

Figure 5.24: Consensus time under controller designed at $\tau = 0.636$ s (Case II)

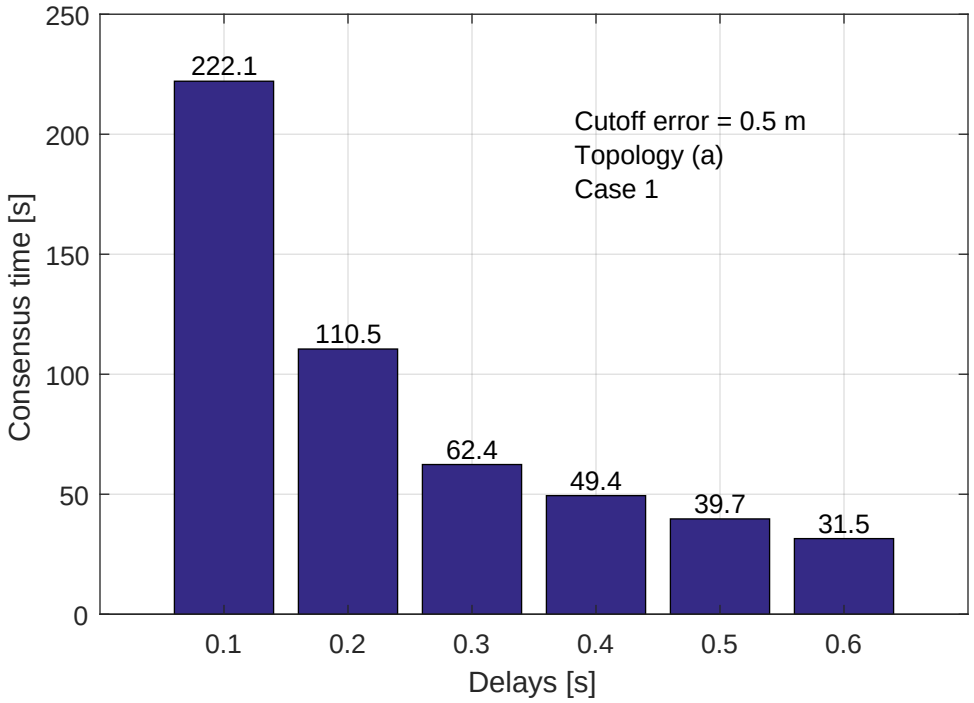

Figure 5.25: Consensus time under topology (a) with respect to delays (Case I)

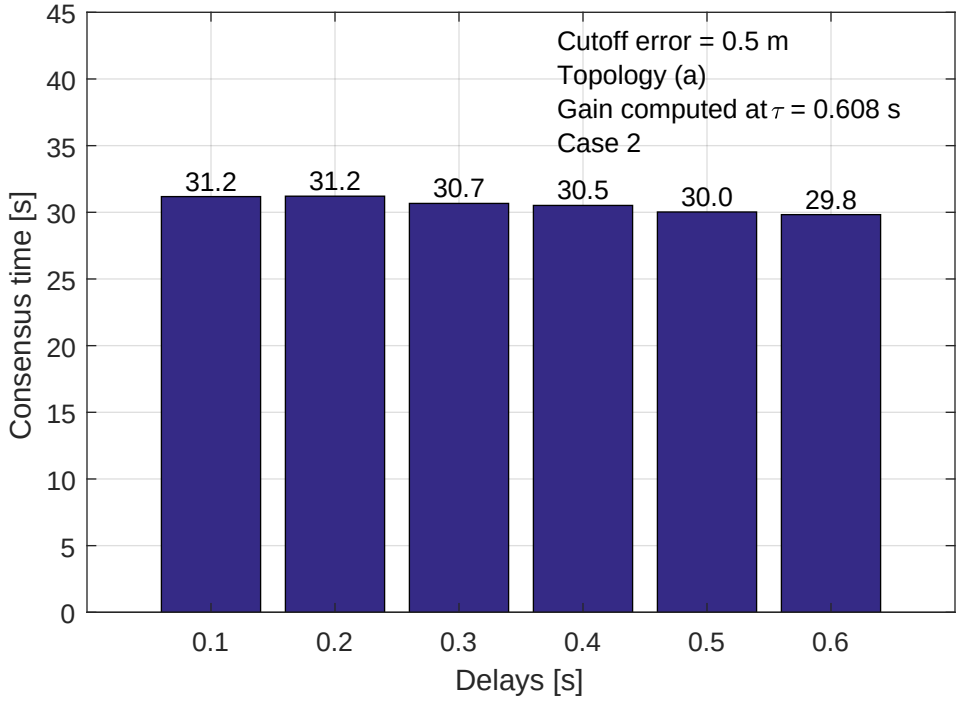

Figure 5.26: Consensus time under topology (a) with respect to delays (Case II)

5.5 Effect of Arbitrary Design Parameters

LMI in (4.11) has four arbitrary design parameters θ_1, θ_2, θ_4 and θ_5. To investigate their impact on the system performance, simulations are run for each combination of two parameters with values from 0.5 to 1.5 while others are fixed at 1. τ_{max} and t_c are then calculated and discussed. The communication topology shown in Fig. 5.2(e) is chosen for all simulation cases. In calculating the consensus time, the system delay and the cutoff error are set as 0.4 s and 0.1 m.

The relation between τ_{max} and parameter set (θ_1, θ_2) is given in Fig. 5.27. It can be observed that τ_{max} increases as both θ_1 and θ_2 increase, and eventually reach a maximum when $\theta_1 = 1.5$ and $\theta_2 = 1.5$. Besides, the gradients are getting smaller in both directions as θ_1 and θ_2 getting larger, thus resulting a flatter surface. The relation between t_c and the same parameter set is shown in Fig. 5.28, and t_c reaches a minimum, thus achieves fastest consensus, when θ_1 and θ_2 reach minimum values. Therefore, there is a trade-off between the robustness against delays and the consensus speed in this parameter set. Similar conclusions can be drawn from conducting the discussions for parameter set (θ_2, θ_5) and relation between τ_{max} and this parameter set is given in Fig. 5.29, and Fig. 5.30 for t_c.

The relation between τ_{max} and parameter set (θ_1, θ_4) is given in Fig. 5.31. It shows that τ_{max} is relatively insensitive to θ_1 and it increases as θ_4 decreases. As shown in Fig. 5.32, similar results exist for the relation between t_c and the same parameter set. As a result, the trade-off still exists in this parameter set. Similar conclusions can be drawn for parameter sets (θ_2, θ_4) and (θ_4, θ_5). The relations between system performance indexes (τ_{max}, t_c) and these parameter sets are given in Figs. 5.33 to 5.36.

The relations between system performance indexes and the parameter set (θ_1, θ_5) are shown in Figs. 5.37 and 5.38. It can be observed that τ_{max} increases when θ_1 increases and θ_5 reduces while t_c is relatively insensitive to θ_5 and decreases as θ_1 decreases. In addition, τ_{max} and t_c all reach the maximums when θ_1 reaches maximum value, thus the trade-off retains. However, it can be observed that the gradient of τ_{max} in θ_1 direction is relatively small, thus the parameters of $\theta_1 = 0.5$ and $\theta_5 = 0.5$ can be chosen as suboptimal parameter set where the system has the fastest consensus speed and also has some robustness against delays.

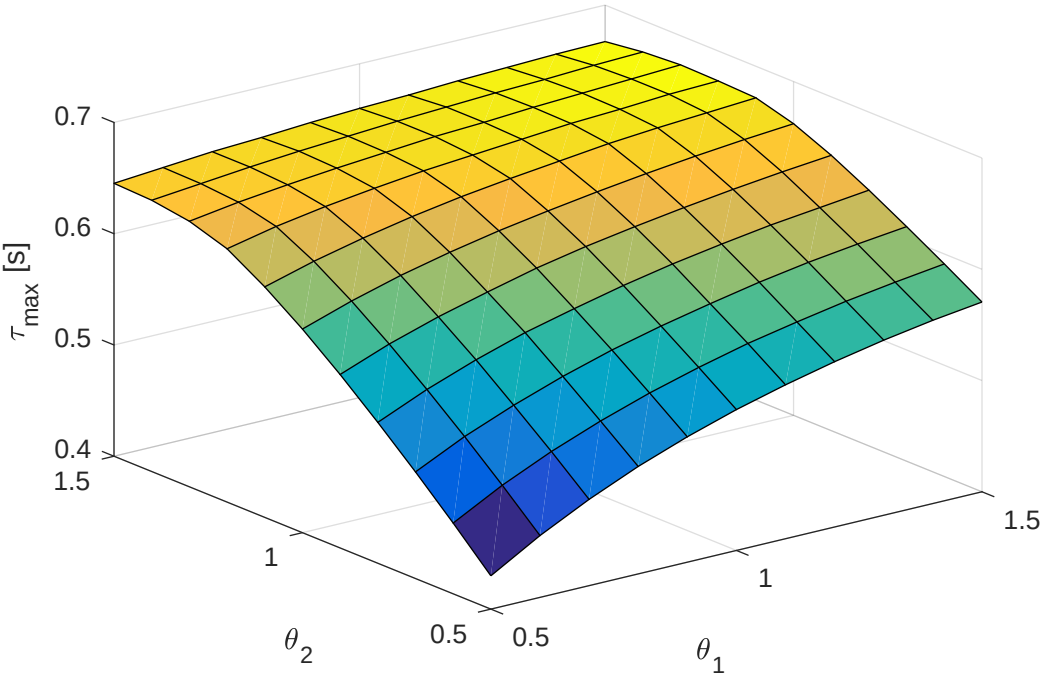

Figure 5.27: Maximum allowable delay vs. θ_1 and θ_2

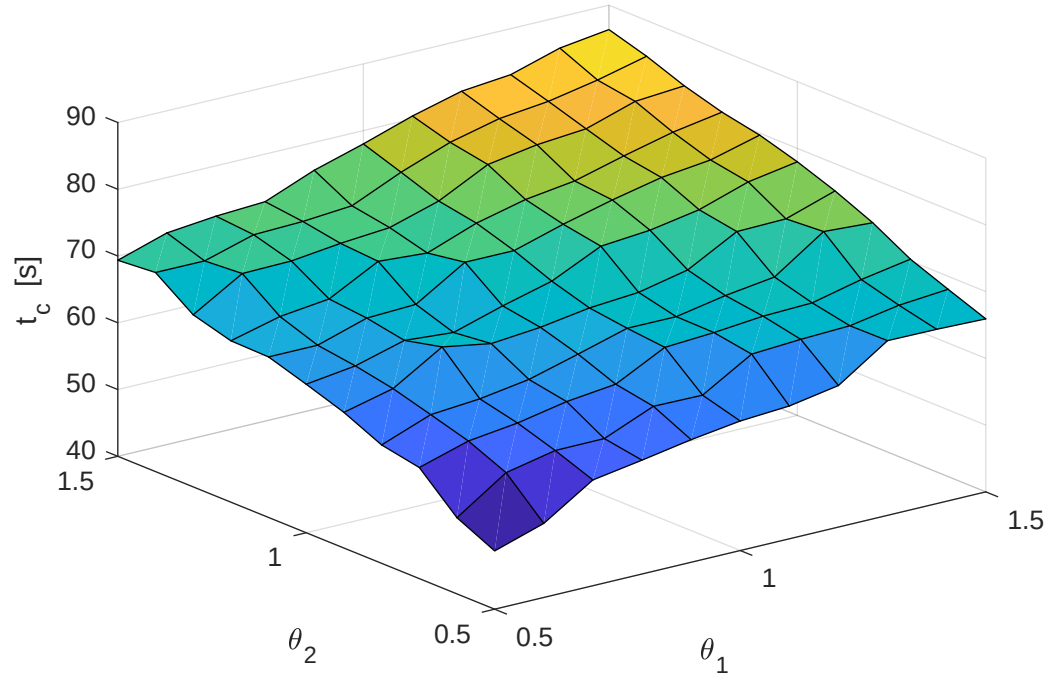

Figure 5.28: Consensus time vs. θ_1 and θ_2

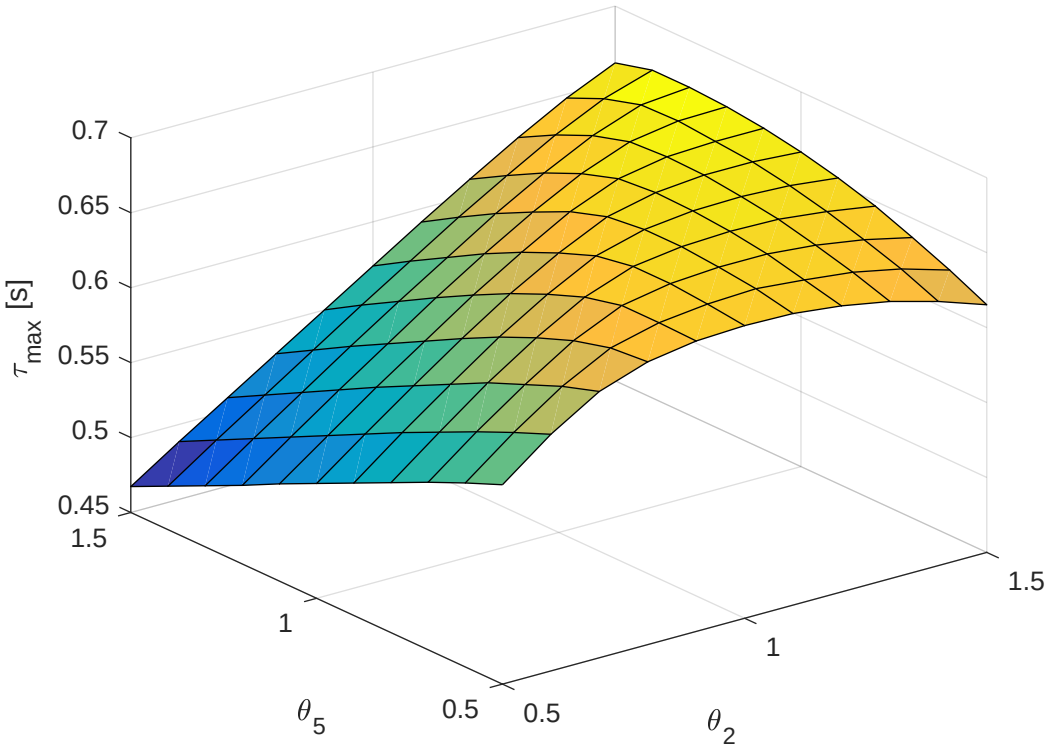

Figure 5.29: Maximum allowable delay vs. θ_2 and θ_5

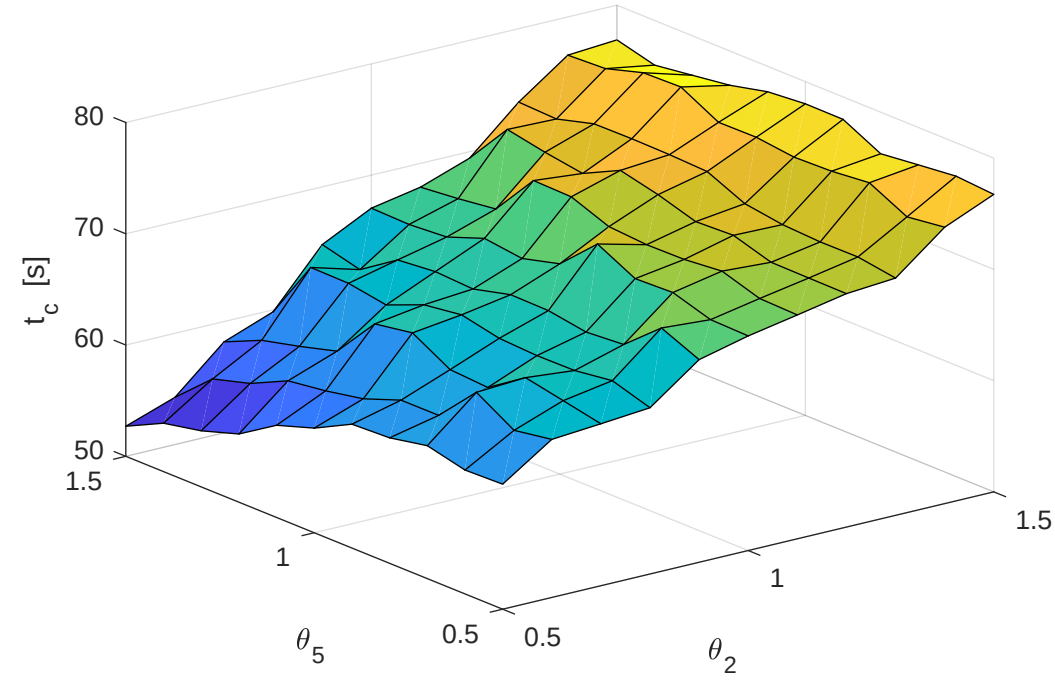

Figure 5.30: Consensus time vs. θ_2 and θ_5

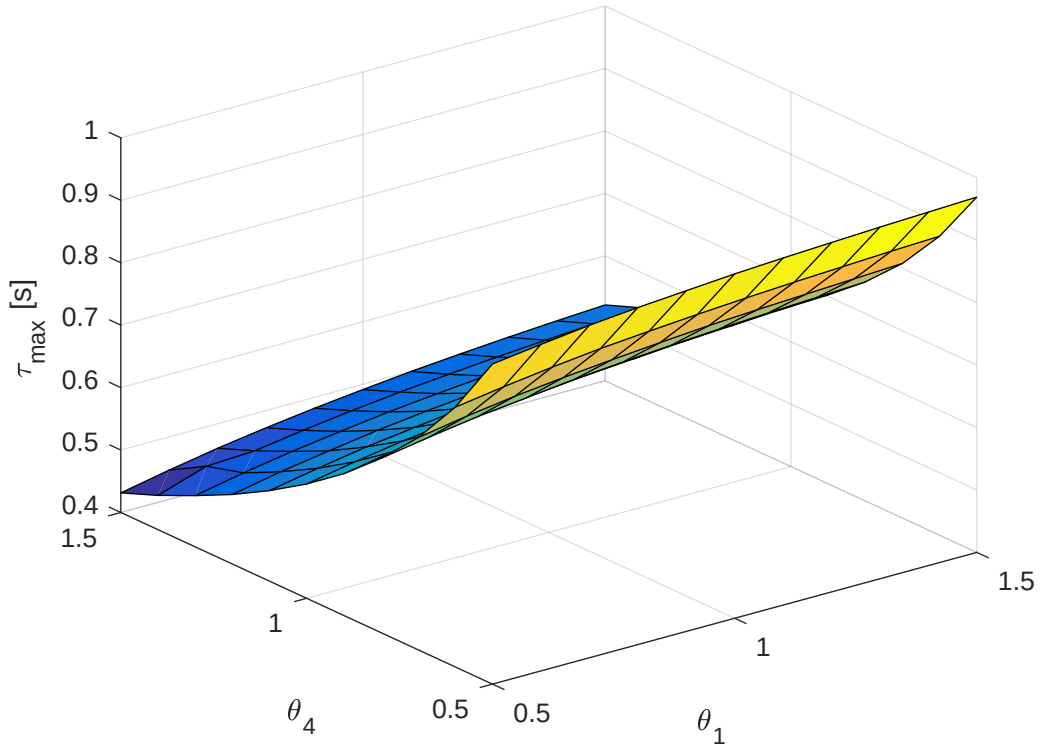

Figure 5.31: Maximum allowable delay vs. θ_1 and θ_4

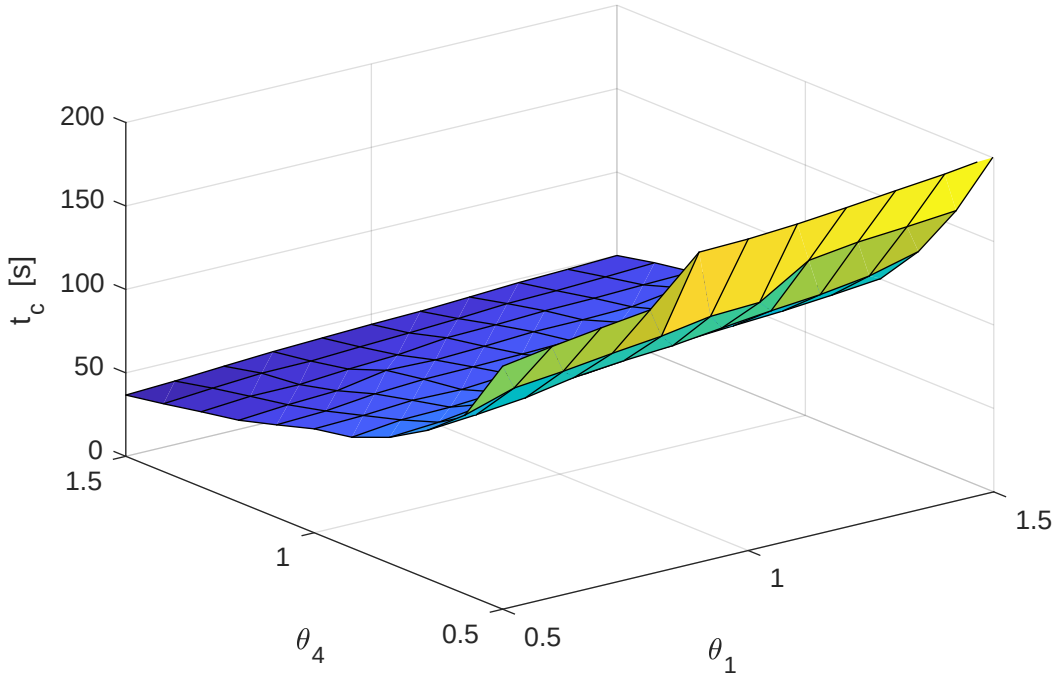

Figure 5.32: Consensus time vs. θ_1 and θ_4

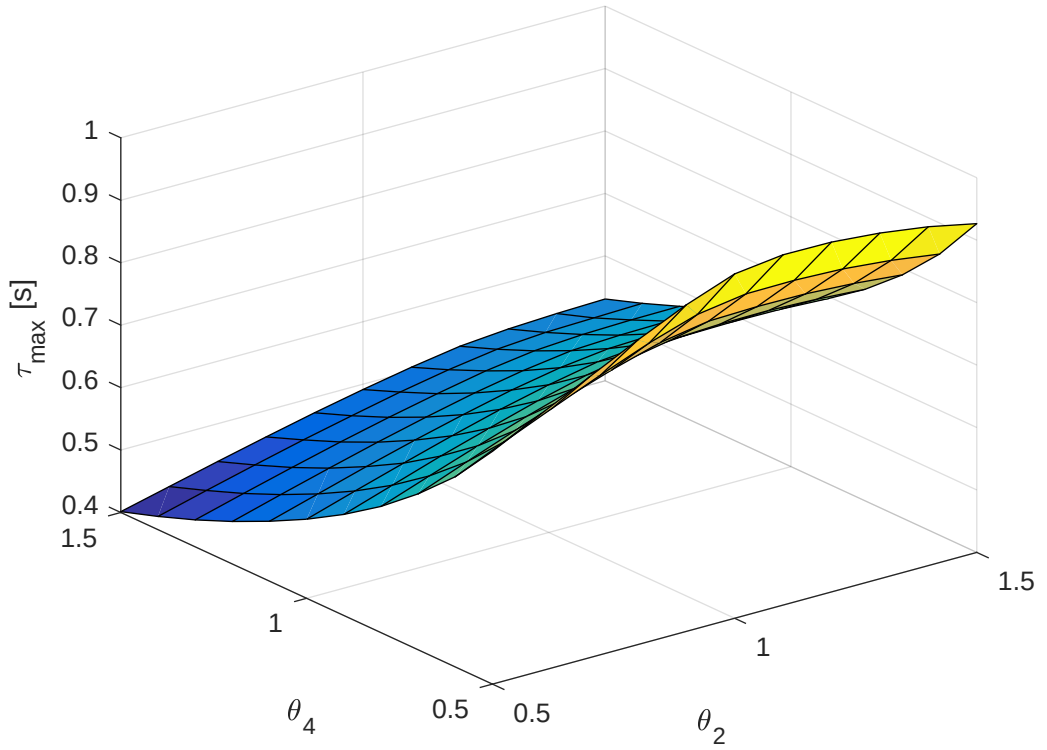

Figure 5.33: Maximum allowable delay vs. θ_2 and θ_4

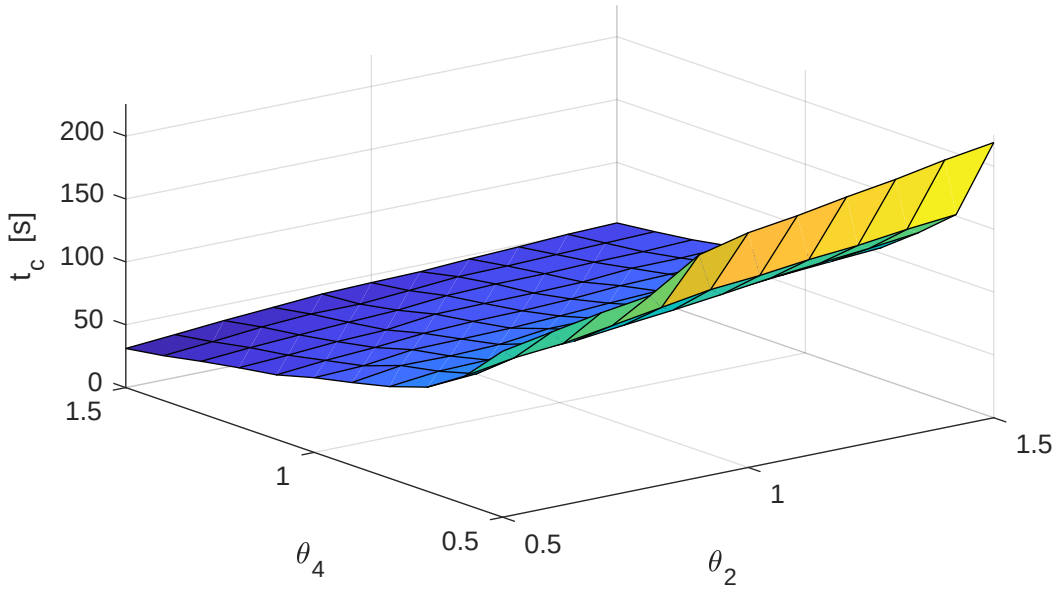

Figure 5.34: Consensus time vs. θ_2 and θ_4

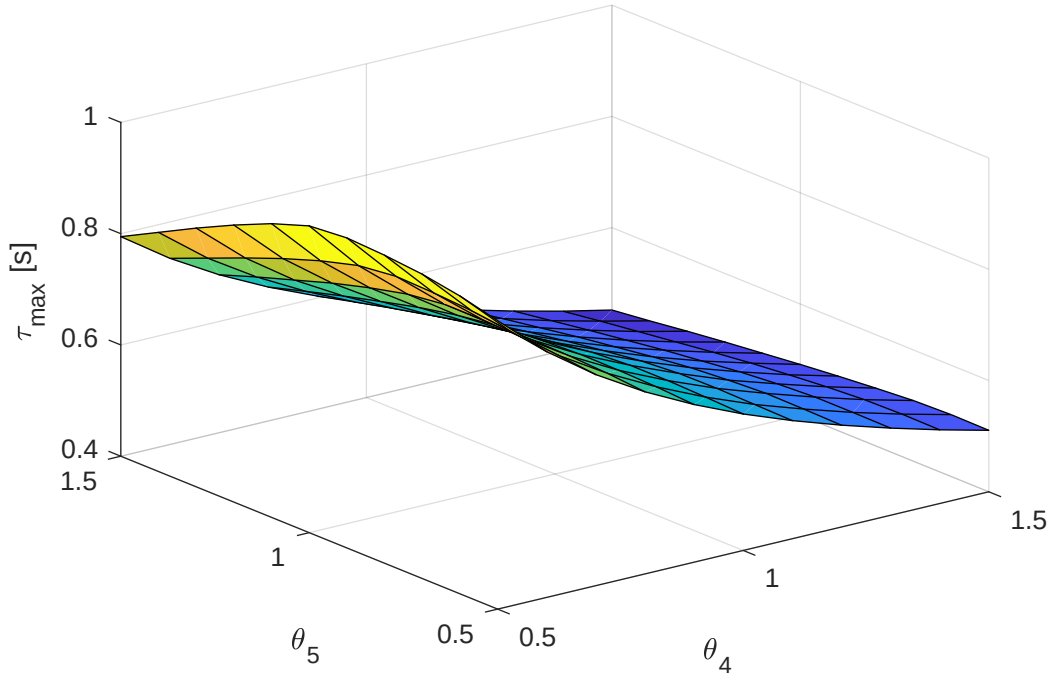

Figure 5.35: Maximum allowable delay vs. θ_4 and θ_5

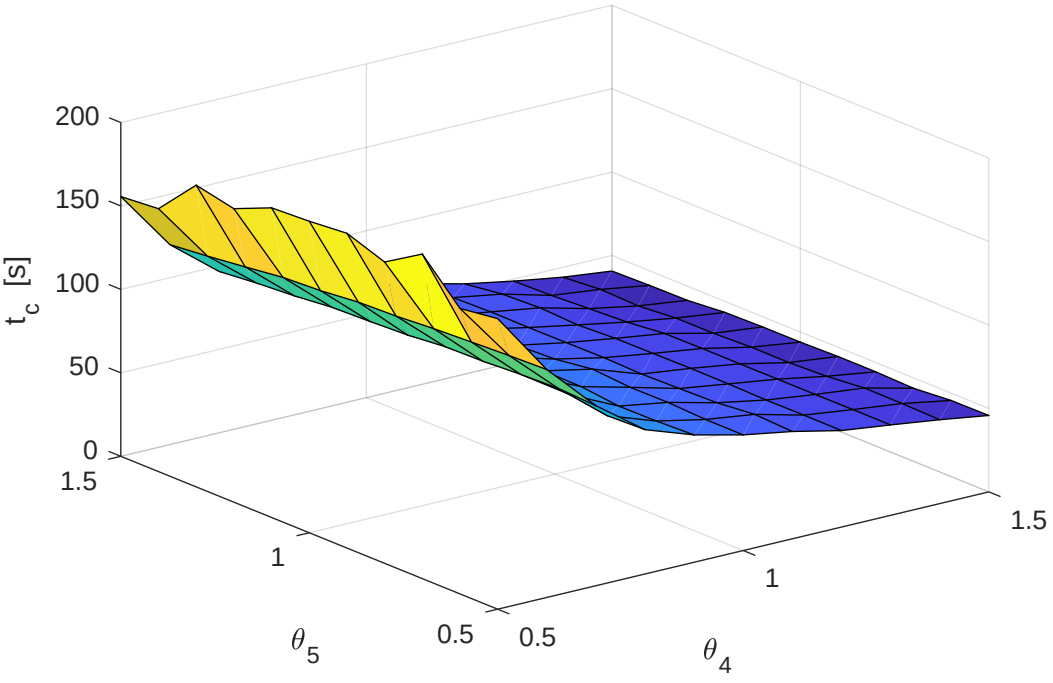

Figure 5.36: Consensus time vs. θ_4 and θ_5

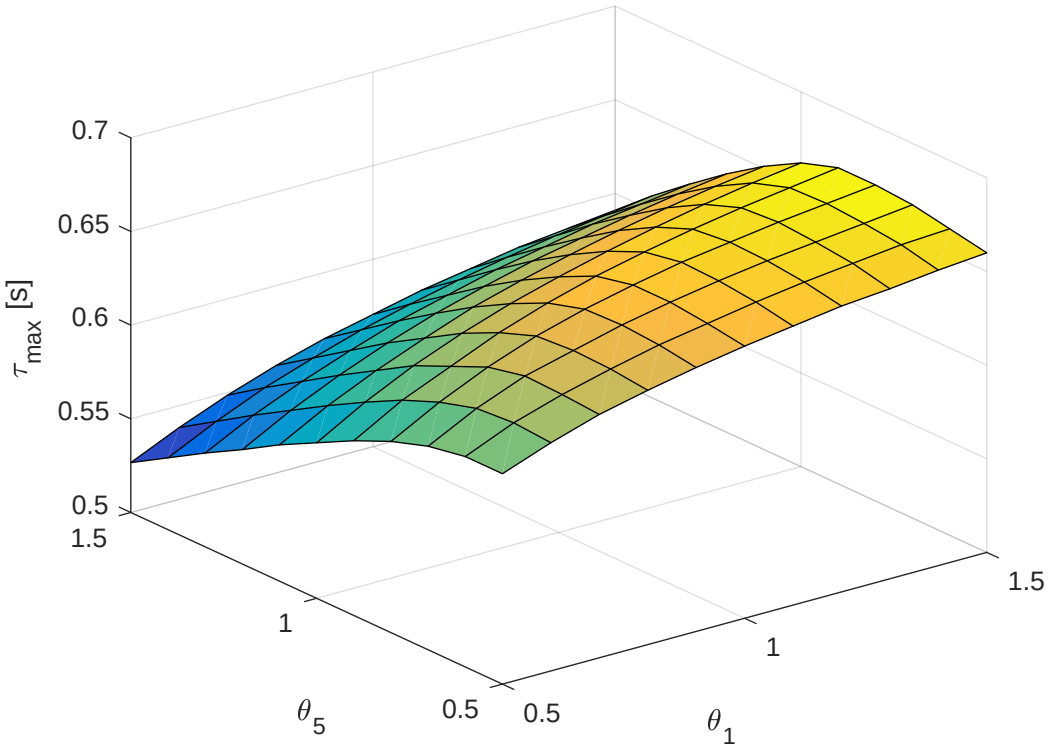

Figure 5.37: Maximum allowable delay vs. θ_1 and θ_5

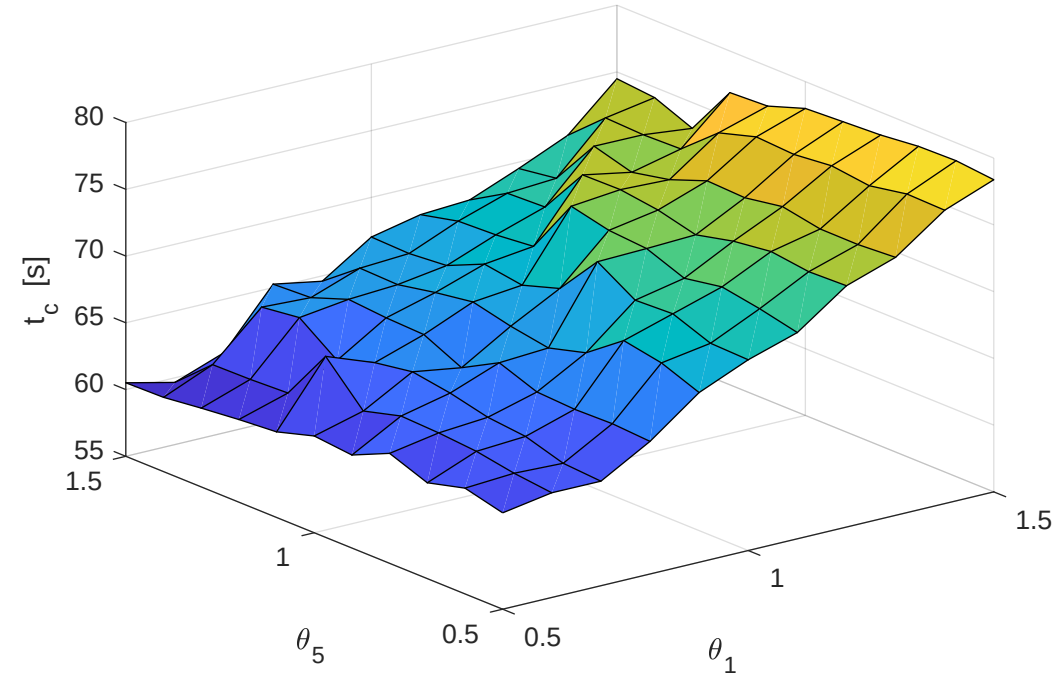

Figure 5.38: Consensus time vs. θ_1 and θ_5

5.6 Summary

Discussions on the system performance with respect to different communication topologies, system scales, system delays and arbitrary design parameters are successfully conducted. The centralized communication topology outperforms other topologies. Large scale systems are more susceptible to communication delays and tend to converge slowly. Systems actually achieve faster consensus speed under larger delays. Finally, there is a trade-off between the robustness against delays and the consensus speed when choosing the arbitrary design parameters.

Chapter 6

Simulation Results of the Time-Varying Delay Case

MATLAB simulation results of the system under asynchronous time-varying communication delays are discussed in this chapter. In particular, discussions will be focusing on: the optimal error bound parameter design; the effect of time-varying delays; the system performance comparison; and the trajectory planning. Communication topology shown in Fig. 5.2(e) is adopted for all simulation cases and the arbitrary design parameters θ_i, $i = 1, \cdots, 5$ are set as 1.

6.1 Optimal Error Bound Parameter Design

Different from the controller design in the constant delay case, two additional design parameters γ and $\bar{S}$ are introduced for compressing the system error bound in the time-varying delay case. From Theorem 2, if the LMIs in (4.19) are satisfied, the consensus error will be bounded by (4.27). However, as described in Section 4.2.2, there is a trade-off between the feasibility of the controller and the error bound. Therefore, it is necessary to design an optimal controller over a prescribed region such that the feasibility is ensured and the error bound is minimized. In the feasibility test of LMIs in (4.19), the $\bar{S}$ is modified as $s * I$, where s is an arbitrary positive scalar and then $\lambda_{min}(\bar{S})$ becomes s. Thus, the error bound is minimized when the product of γ and s is maximized. The feasibility test is conducted for γ and s range from 0.01 to 1 and the results are given in Fig. 6.1, where the yellow regions are guaranteed to be feasible. The enlarged results is given in Fig. 6.2 and within these feasible regions, the product of γ and s is maximized when $\gamma = 0.01$ and $s = 0.18$. These values are used subsequently for all simulations in this chapter.

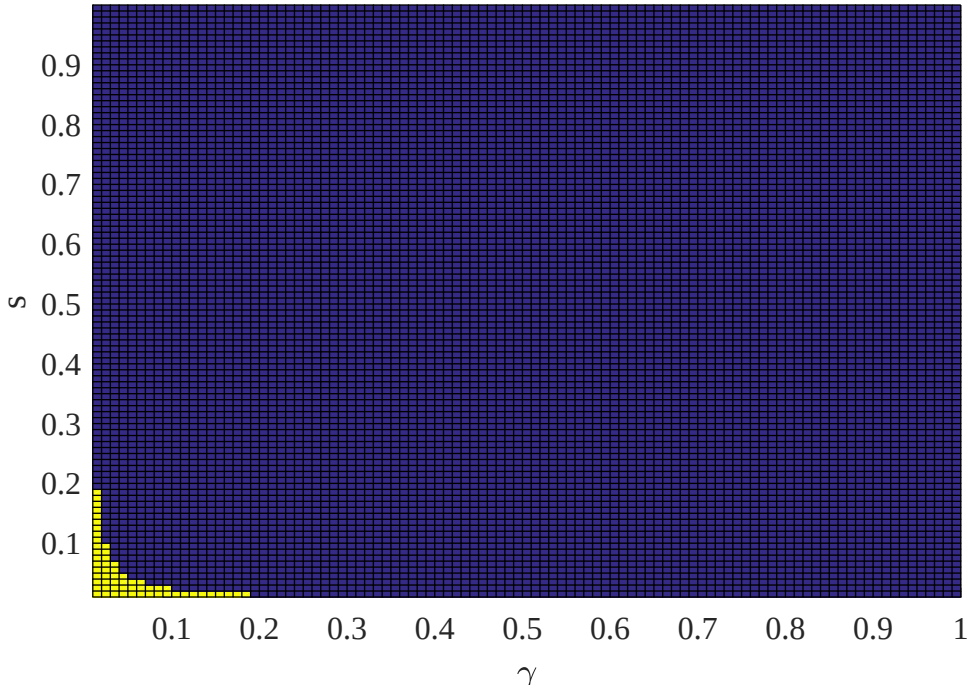

Figure 6.1: Feasibility region of stable controller, where the yellow areas are guaranteed to be feasible

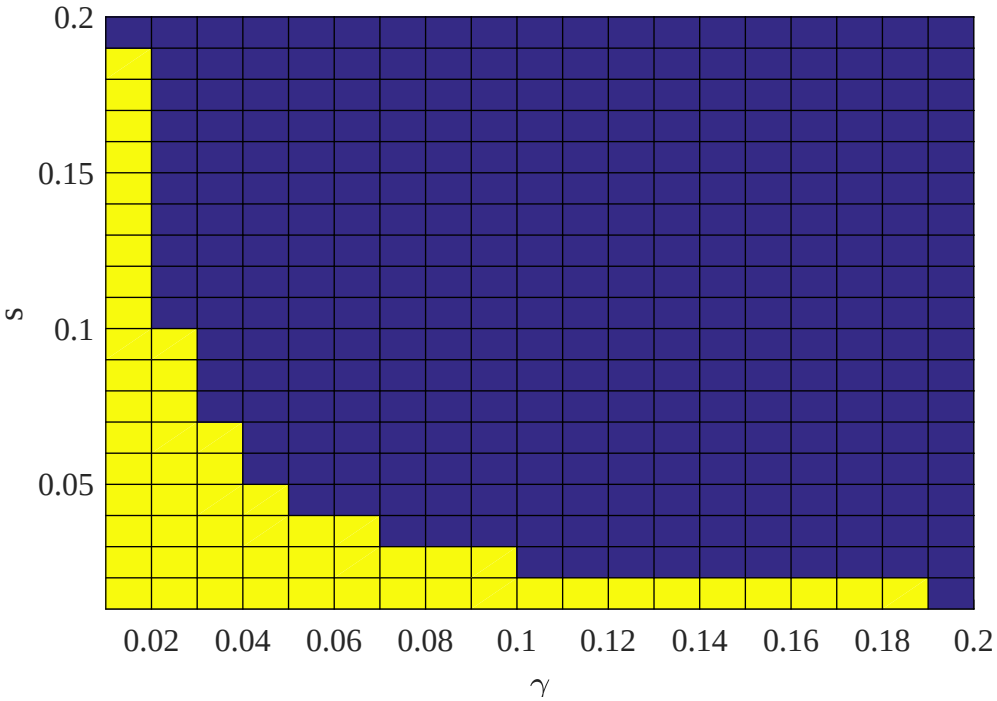

Figure 6.2: Enlarged feasibility region of stable controller, where the yellow areas are guaranteed to be feasible

6.2 Effect of Time-Varying Delays

This section discusses the system performance under the effect of nominal delay τ, time-varying potion delay bound d and types of time-varying delay. The leader sits stationary at the origin, with the followers initially hovering at locations evenly distributed in the space. l_2-norm of the consensus error is introduced as an extra criterion for evaluating the system performance in order to take account of the oscillations induced by time-varying delays.

6.2.1 Effect of Nominal Delays

In studying the effect of nominal delay, a controller is designed at $\tau = 0.1\,s$ and $d = 0.01\,s$, which then subsequently implemented to systems with τ ranging from $0.01\,s$ to $0.4\,s$, without changing d. In addition, for a smooth system response, the time-varying portion of the delay $d_{ij}(t)$ in each communication channel is implemented as a sinusoidal signal $d * \sin(2\pi f_{ij} t)$ where f_{ij} denotes the signal frequency in channel from agent j to agent i. Delay signals with $\tau = 0.1\,s$ and $d = 0.01\,s$ for communication topology (e) are given in Figs. 6.3 and 6.4.

It is shown in Fig. 6.5, as expected, that the l_2-norm increases thus the system performance degrades, as the nominal delay increases. Moreover, it increases rapidly when the nominal delay gets closer to $0.4\,s$ and eventually simulation fails when the nominal delay exceeds $0.43\,s$.

6.2.2 Effect of Varying Delay Bound

The same controller designed in Section 6.2.1 is adopted in studying the effect of the varying delay bound on system performance. However, in this subsection, the controller is implemented to systems with d ranging from $0.01\,s$ to $0.09\,s$ and a fixed τ. l_2-norm with respect to various d is shown in Fig. 6.6. Although increasing d tends to slightly reduce the l_2-norm, there is only 2.5% difference between the maximum and minimum values. Therefore, it can be concluded that the system performance is relatively insensitive to d given an unchanged τ.

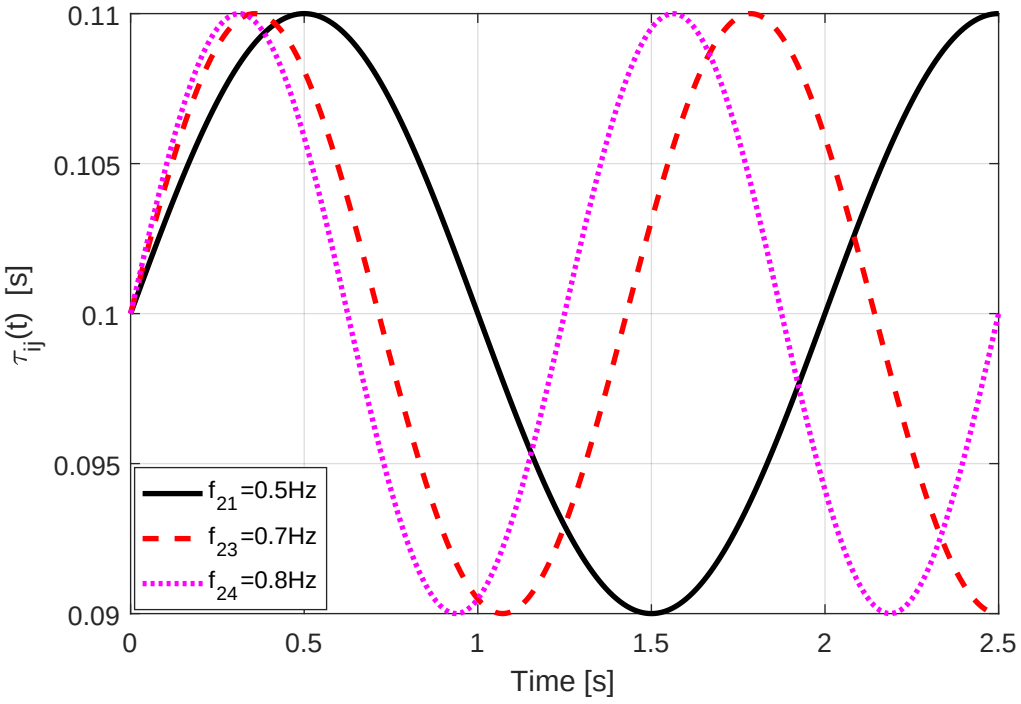

Figure 6.3: Delay signals in channels to agent two

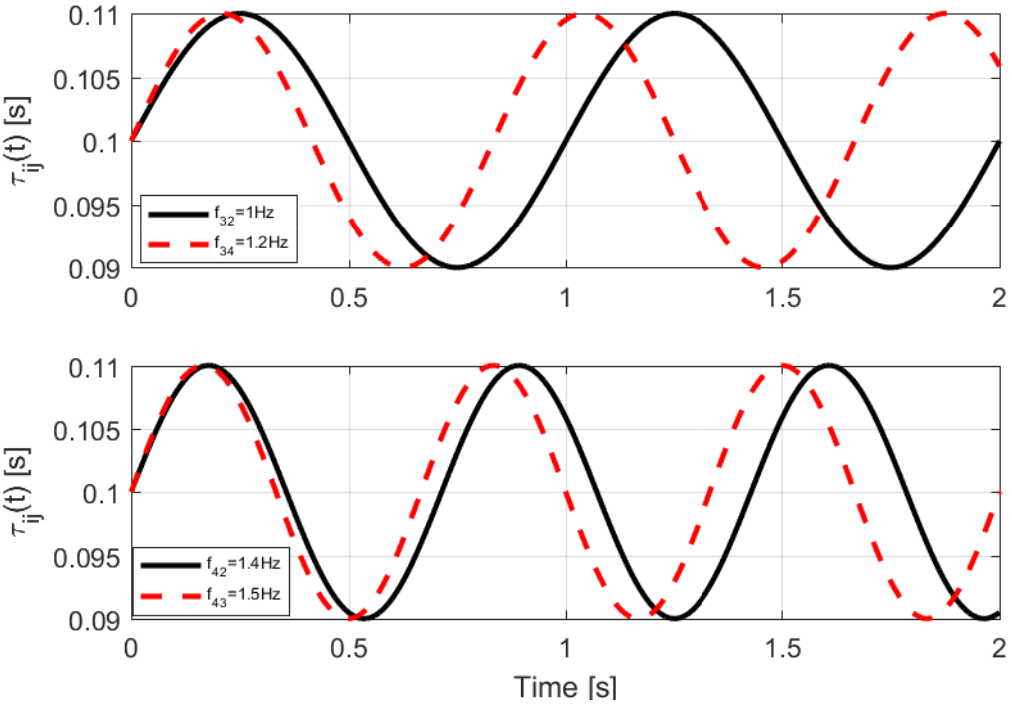

Figure 6.4: Delay signals in channels to agent three and four

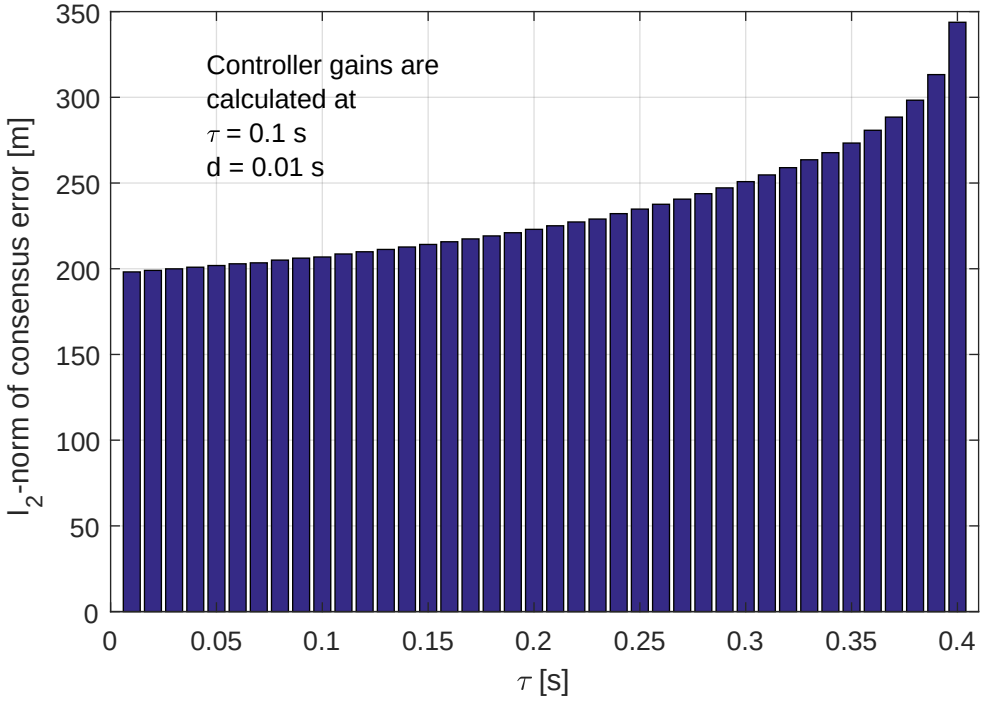

Figure 6.5: l_2-norm of consensus error vs. nominal system delay

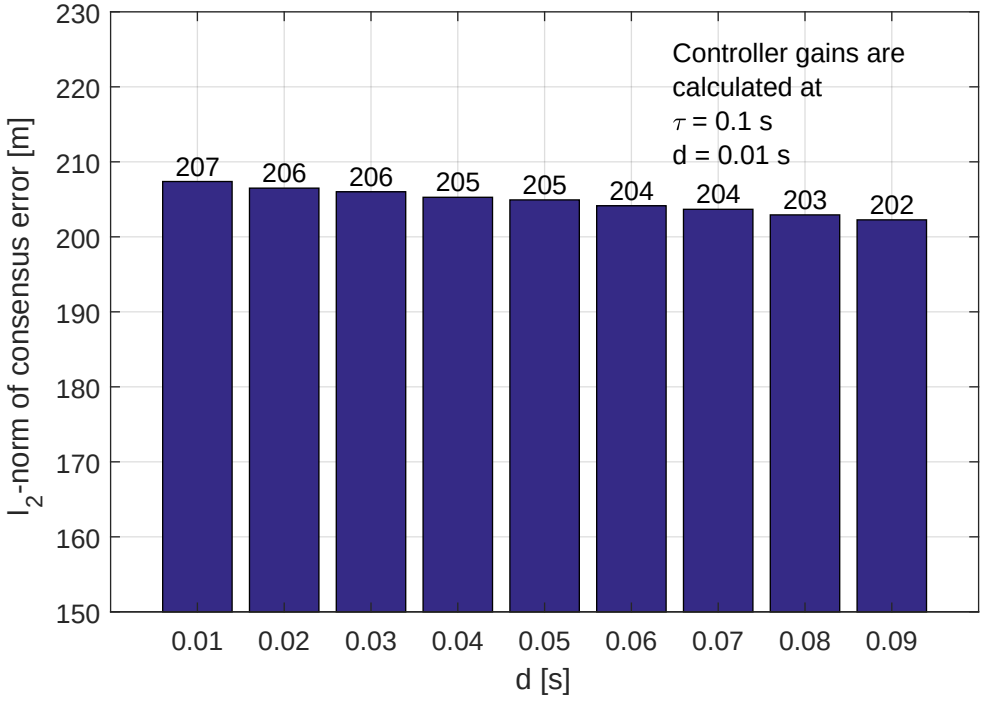

Figure 6.6: l_2-norm of consensus error vs. varying delay bound

6.2.3 Effect of Type of Time-Varying Delays

This subsection discusses the effect of type of time-varying delays on system performance. The random delay signal with a bound of $0.05\,s$, as shown in Fig. 6.7, is implemented in simulation for performance comparison with the sinusoidal type of delays. It is discovered that the type of delay generally has very little impact on system performance given a same delay bound. To demonstrate, the system consensus error profiles under random and sinusoidal delays, given in Fig. 6.8, are almost overlapped. In addition, as shown in Fig. 6.9, the l_2-norm of consensus error under two types of delay are similar and the system has slightly better performance under smooth sinusoidal delays.

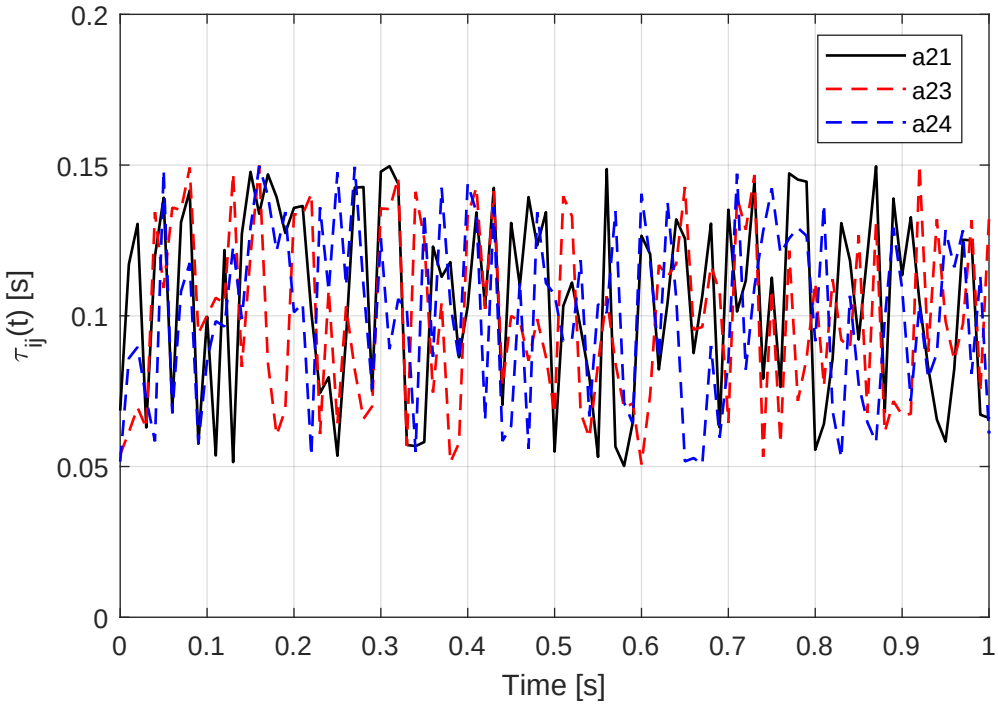

Figure 6.7: Random delays in channels to agent two

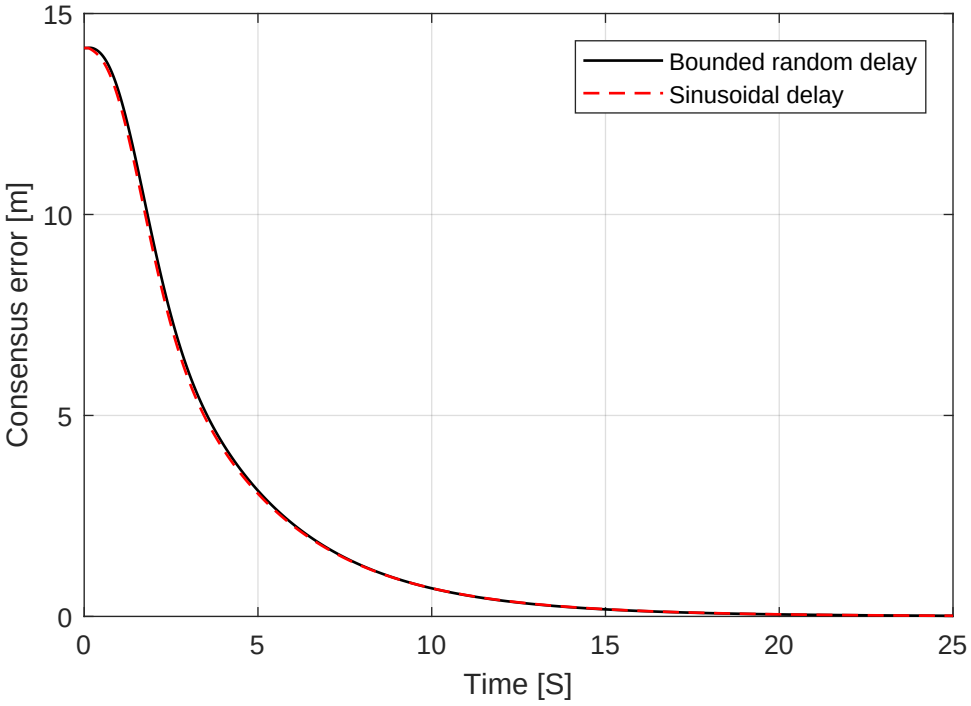

Figure 6.8: Consensus error under random and sinusoidal delays

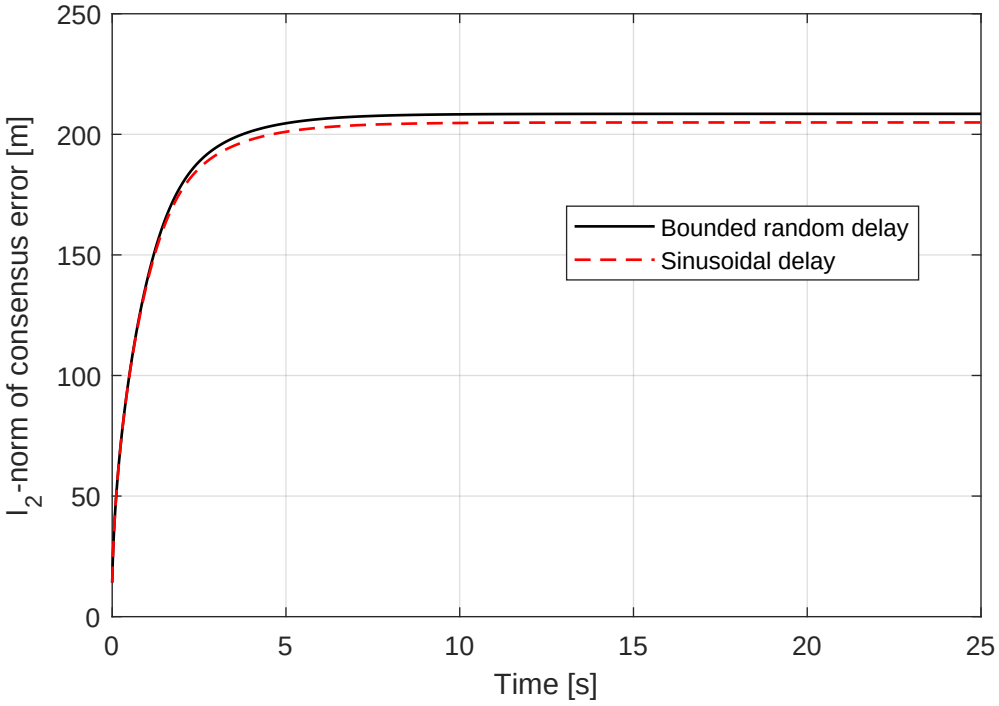

Figure 6.9: l_2-norm of consensus error under two types of delays

6.3 System Performance Comparison

To compare the system performance, a controller is designed based on the constant delay condition (4.11) and implemented to the system under a nominal delay of 0.1 s and a varying-delay bound of 0.01 s afterwards. It is then compared with the controller designed based on the time-varying delay condition (4.19). The l_2-norm results for both controllers, as shown in Fig. 6.10, indicate that the controller designed based on (4.19) outperforms the controller designed based on (4.11) since the former one tends to induce a faster convergence and a smaller l_2-norm. Similar result can be obtained from the consensus error profile shown in Fig. 6.11, where the consensus error bound is larger with the controller designed based on (4.11). It is also interesting to investigate the controller gains, as listed in Table 6.1, for both controllers. It can be observed that control gains designed under (4.19) are generally larger than the case under (4.11). It means that the LMI in condition (4.19) actually tries to compensate the effect of time-varying portion of the delay by outputting large gains. However, large controller gains imply large control effort which might not be realizable in the actual implementation.

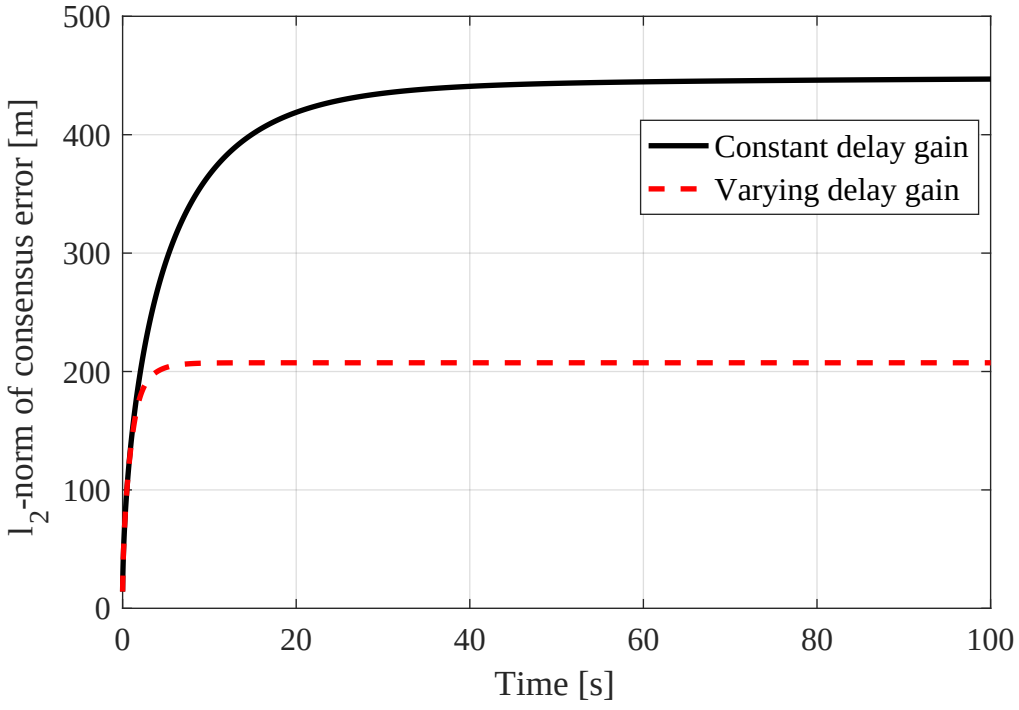

Figure 6.10: l_2-norm of consensus error under two described controllers

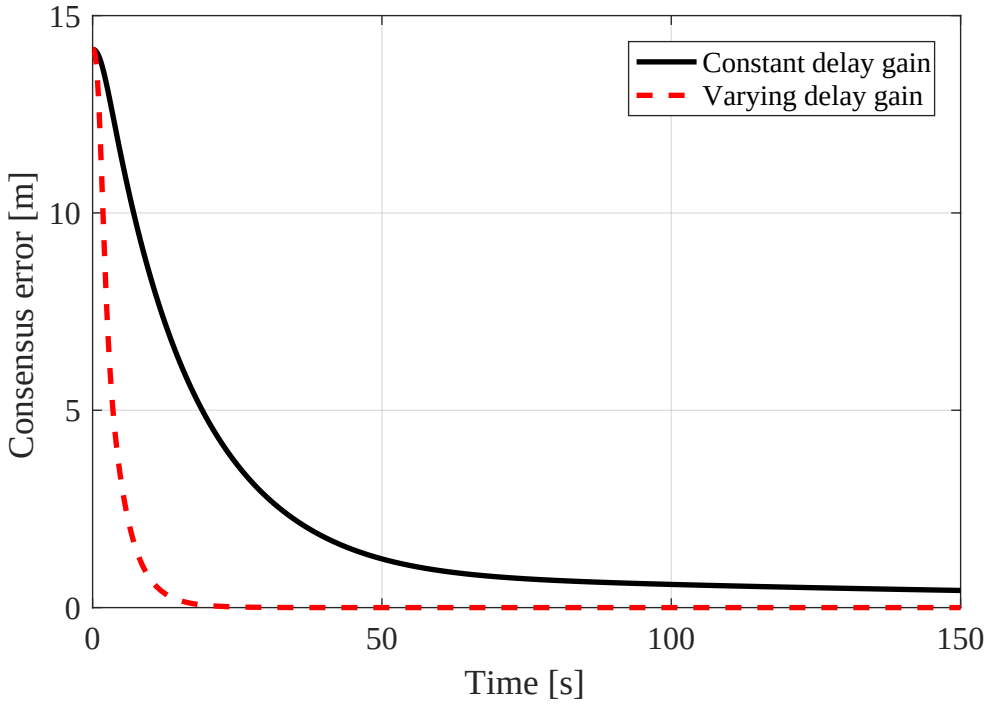

Figure 6.11: Consensus error under two described controllers

Table 6.1: Control gains under constant and time-varying delays

Gains	$\mathbf{k}_2$	$\mathbf{k}_4$
Constant delay	$\begin{bmatrix} 0.1833 & 0.2801 \end{bmatrix}$	$\begin{bmatrix} 0.1294 & 0.1066 & -0.1865 & -0.1471 \end{bmatrix}$
Time-varying delay	$\begin{bmatrix} 1.2419 & 1.4259 \end{bmatrix}$	$\begin{bmatrix} 4.7298 & 8.5353 & 4.7316 & 1.0009 \end{bmatrix}$

6.4 Trajectory Planning

This section presents two trajectory planning methods by implementing the designed controllers: the waypoints and the virtual leader.

6.4.1 Trajectory Planning with Waypoints

The proposed controllers can be directly applied to the trajectory planning with waypoints since each waypoint can be treated as a stationary leader and under the proposed controllers, all the followers are able to converge to the leader. As a consequence, the followers are able to converge the trajectory specified by the waypoints given long enough time interval between the two waypoints.

The system trajectory in Fig. 6.12 is obtained by setting a sequence of waypoints of $(0, 0, 20)$, $(20, 0, 20)$, $(20, 20, 20)$,$(0, 20, 20)$ and a time interval of 30 s between two waypoints. The controller is designed under $\tau = 0.1$ s and $d = 0.01$ s. Nevertheless, for complex trajectories, more waypoints are required and the waiting time between waypoints are hard to determine.

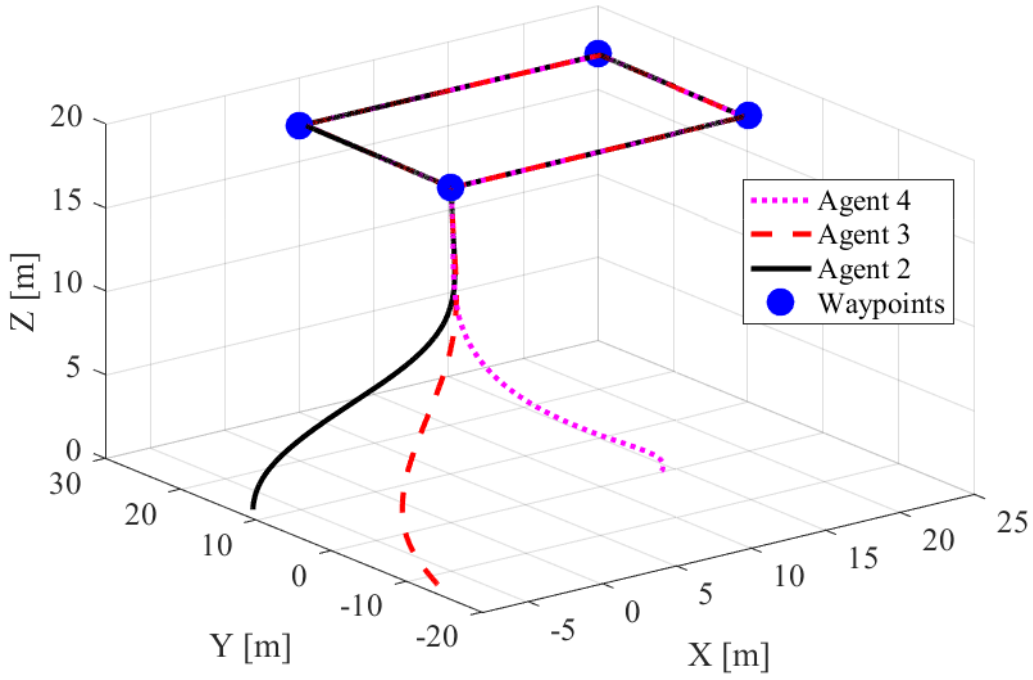

Figure 6.12: Trajectory planning with waypoints

6.4.2 Trajectory Planning with a Virtual Leader

The system under proposed controller is capable of tracking complex trajectories properly planned by a virtual leader which is independent from any followers. Fig. 6.13 shows the position profile of the system under controller designed based on constant delay of 0.1 s, and with a virtual leader which has a helix trajectory given below

$$\begin{cases} x(t) = & r\cos(\frac{2\pi}{T}t) \\ y(t) = & r\sin(\frac{2\pi}{T}t) \\ z(t) = & 0.1t, \end{cases} \tag{6.1}$$

where, r denotes the radius and T is the period. The translational responses are given in Fig. 6.14.

The consensus error profile in Fig. 6.16 shows that the consensus can only be

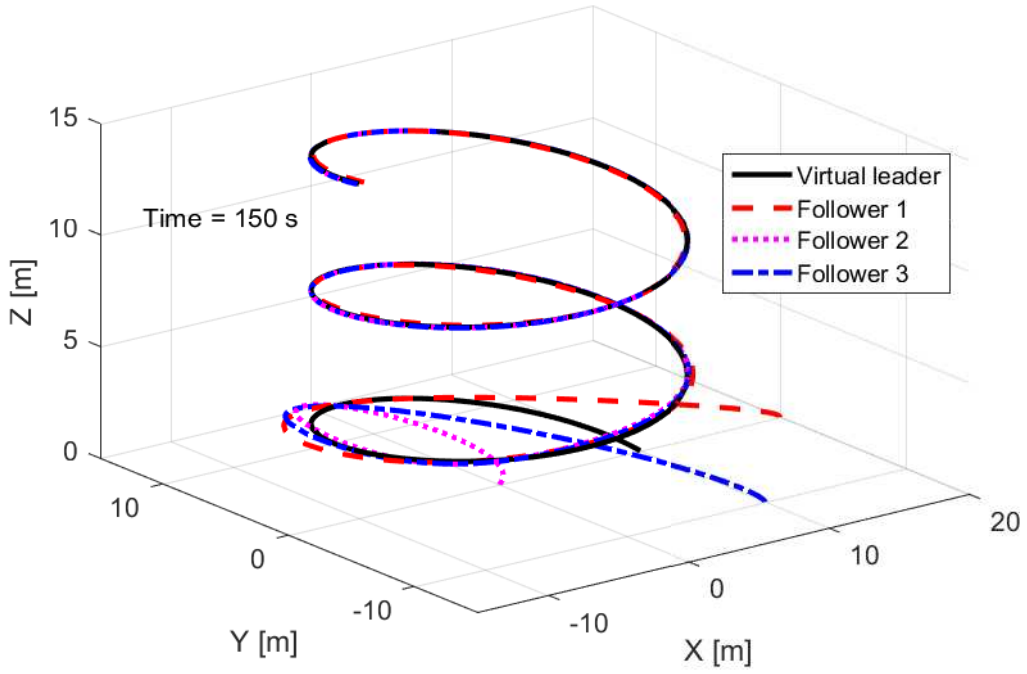

Figure 6.13: Trajectory planning with a virtual leader

reached with a bounded error. This can be analyzed by considering the controller design which is based on the assumption that roll and pitch angles will be converging to zero. However, with the circular trajectory, the roll and pitch responses as shown in Fig. 6.15 will oscillate around the origin rather than converging to it. Consequently, there always exists a model mismatch for this kind of trajectory. Still, the effect of the model mismatch can be reduced by properly choosing the helix radius and period. To illustrate, with $r = 10\,m$ and $T = 60\,s$ chosen here, the roll and pitch angles shown in Fig. 6.15 only oscillate with an amplitude less than 1 deg, thus the model mismatch is minor. As a result, the consensus error bound is almost negligible.

Lastly, the system is not fully distributed, even though the virtual leader's states are only available to the followers that directly connected to it, its virtual control input is assumed to be available to all followers.

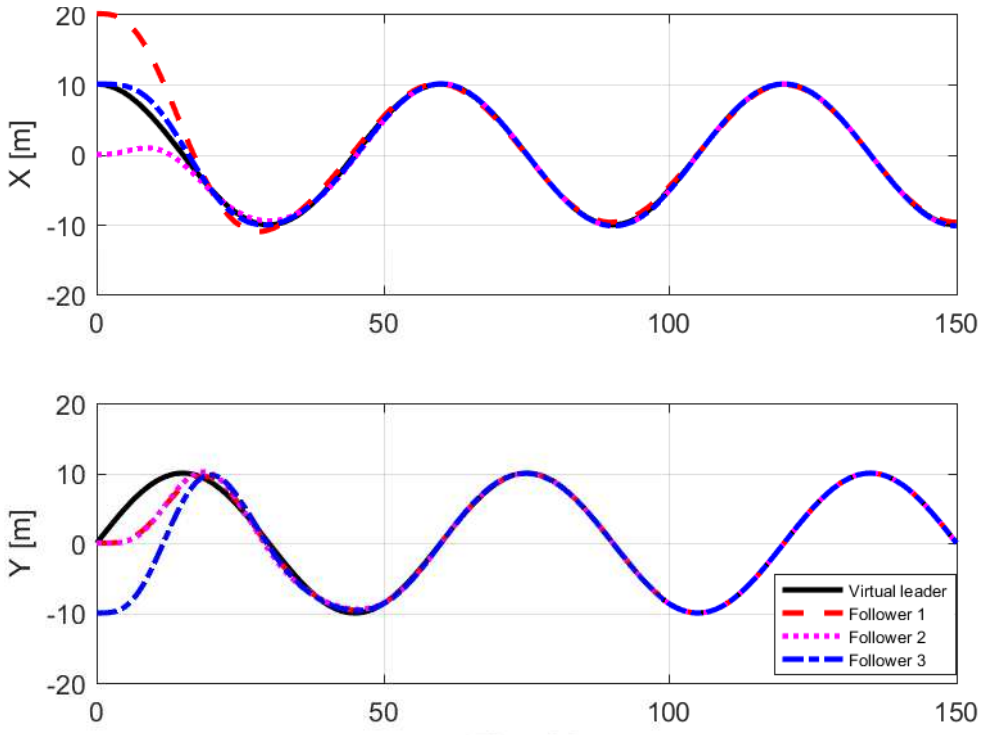

Figure 6.14: X and Y responses of the system with trajectory planning with a virtual leader

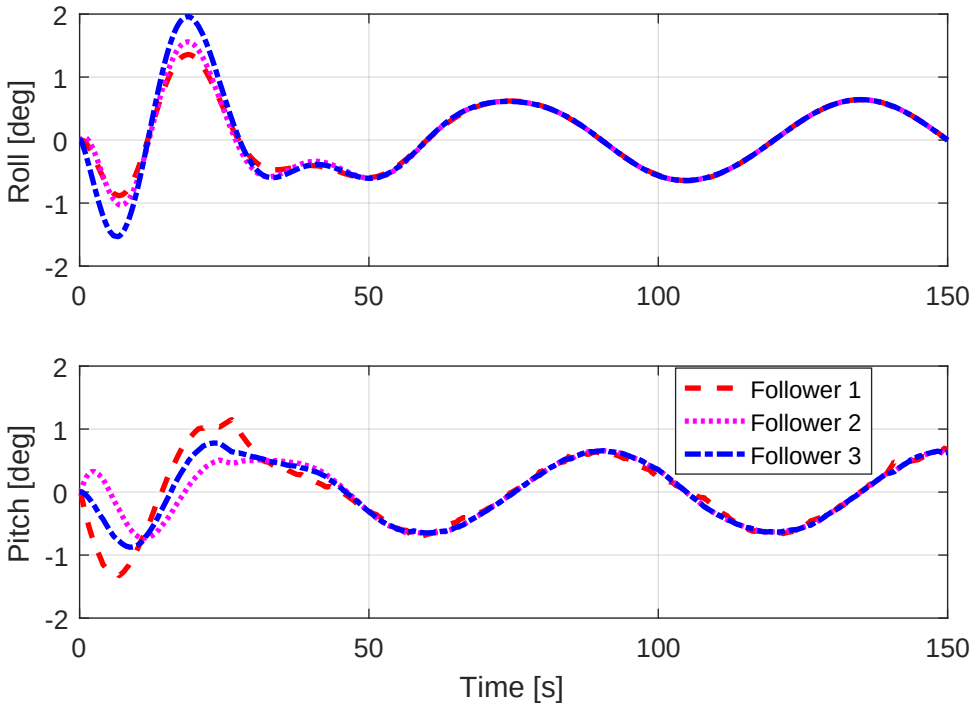

Figure 6.15: Roll and pitch responses of the trajectory planning with a virtual leader

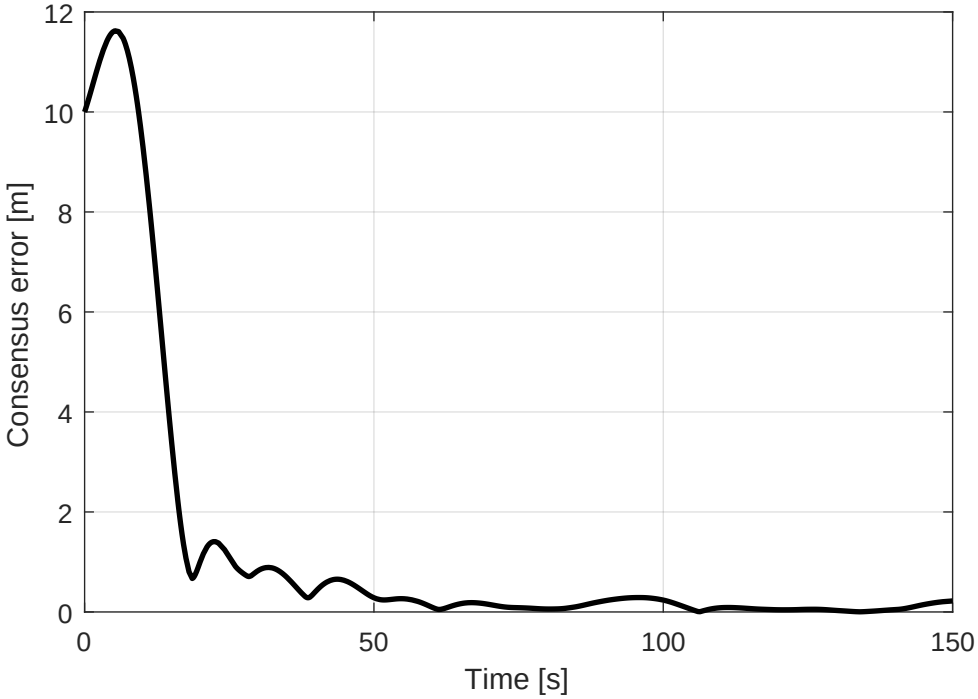

Figure 6.16: Consensus error with a virtual leader

6.5 Summary

This section presents the simulation conducted on systems under time-varying delay. The optimal error bound compressing parameters are determined firstly and applied to all the following simulations. The effect of nominal delay and time-varying delay bound have been discussed. The system performance under constant delayed controller and time-varying delayed controller are also compared. At last, the trajectory planning methods of waypoints and virtual leader using the proposed controllers are presented.

Chapter 7

Conclusions and Future Works

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

7.1 Conclusions

In conclusion, sufficient conditions for static feedback control gain, that solves the consensus problem of multiple-quadcopter system asymptotically under synchronous constant delays, and with bounded error under asynchronous time-varying delays, are successfully derived using Lyapunov-based methodologies. Low complexity conditions independent of the network size are proposed when dealing with a system with large number of agents under certain communication topologies. Extensive simulations are conducted and the results in constant delay case indicate that the centralized communication topology has the best system performance, large scale systems are more susceptible to communication delays, systems actually converge faster under larger delays, and in general there is trade-off between the robustness and consensus speed in choosing the arbitrary design parameters. In time-varying delay case, the system performance degrades as the nominal delay increases, and is relatively insensitive to delay bound and types of delays. In addition, the controller designed based on the time-varying condition outperforms the controller based constant delay condition. The proposed consensus algorithms are then successfully applied to trajectory planning for multiple-quadcopter systems.

7.2 Future Works

There are many extensions that are interesting and challenging on this work for future studies.

1. Apply the proposed controllers to simulations with actual nonlinear systems.

2. It is interesting to apply the proposed controllers to real systems. There are some popular experimental platforms, such as Quanser Qball, Asctech Humingbird that may be used to achieve this objective.

3. The controllers in this work are designed based on full state feedback. It is interesting to investigate that if a output feedback controller could achieve control objective.

4. It is necessary to develop a collision avoidance mechanism before actually implementing the algorithms to real systems.

5. It is worthwhile to investigate control design under other communication constraints, such as packet losses, and the coexistence of packet losses and time delays.

6. The hardware limitations should be addressed (e.g. large control input might not be realized by the actual system).

Bibliography

[1] K. A. Ghamry and Y. Zhang, "Cooperative control of multiple uavs for forest fire monitoring and detection," in *Mechatronic and Embedded Systems and Applications (MESA), 2016 12th IEEE/ASME International Conference on*, pp. 1–6, IEEE, 2016.

[2] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems*, pp. 545–558, Springer, 2013.

[3] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, "Mathematical modelling and parameter identification of quadrotor (a survey)," *Procedia Engineering*, vol. 96, pp. 172–181, 2014.

[4] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[5] M. Kumar, K. Cohen, and B. HomChaudhuri, "Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires.," *JACIC*, vol. 8, no. 1, pp. 1–16, 2011.

[6] T. Lee, K. Sreenath, and V. Kumar, "Geometric control of cooperating multiple quadrotor uavs with a suspended payload," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 5510–5515, IEEE, 2013.

[7] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[8] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE transactions on industrial electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.

[9] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems—a survey," *IEEE transactions on industrial informatics*, vol. 9, no. 1, pp. 403–416, 2013.

[10] D. Yue, Q.-L. Han, and J. Lam, "Network-based robust h-inifinity control of systems with uncertainty," *Automatica*, vol. 41, no. 6, pp. 999–1007, 2005.

[11] M. B. Cloosterman, N. Van de Wouw, W. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1575–1580, 2009.

[12] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1177–1181, 2005.

[13] D. Yue, Q.-L. Han, and C. Peng, "State feedback controller design of networked control systems," in *Control Applications, 2004. Proceedings of the 2004 IEEE International Conference on*, vol. 1, pp. 242–247, IEEE, 2004.

[14] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[15] C.-L. Liu and F. Liu, *Consensus Problem of Delayed Linear Multi-agent Systems: Analysis and Design*. Springer, 2016.

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

[17] W. Ren, K. L. Moore, and Y. Chen, "High-order and model reference consensus algorithms in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 678–688, 2007.

[18] Y. Kuriki and T. Namerikawa, "Formation control of uavs with a fourth-order flight dynamics," in *52nd IEEE Conference on Decision and Control*, pp. 6706–6711, IEEE, 2013.

[19] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 213–224, 2010.

[20] Z. Li, X. Liu, P. Lin, and W. Ren, "Consensus of linear multi-agent systems with reduced-order observer-based protocols," *Systems & Control Letters*, vol. 60, no. 7, pp. 510–516, 2011.

[21] Y. Zhao, G. Wen, Z. Duan, X. Xu, and G. Chen, "A new observer-type consensus protocol for linear multi-agent dynamical systems," *Asian Journal of Control*, vol. 15, no. 2, pp. 571–582, 2013.

[22] E. Nuno, R. Ortega, L. Basanez, and D. Hill, "Synchronization of networks of nonidentical euler-lagrange systems with uncertain parameters and communication delays," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 935–941, 2011.

[23] N. Chopra and M. W. Spong, "On exponential synchronization of kuramoto oscillators," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 353–357, 2009.

[24] G. S. Seyboth, G. S. Schmidt, and F. Allgöwer, "Cooperative control of linear parameter-varying systems," in *2012 American Control Conference (ACC)*, pp. 2407–2412, IEEE, 2012.

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

[26] X. Gong, Y.-J. Pan, J.-N. Li, and H. Su, "Leader following consensus for multi-agent systems with stochastic packet dropout," in *2013 10th IEEE International Conference on Control and Automation (ICCA)*, pp. 1160–1165, IEEE, 2013.

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

[28] W. Yang, A. L. Bertozzi, and X. Wang, "Stability of a second order consensus algorithm with time delay," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 2926–2931, IEEE, 2008.

[29] J. Hu and Y. Hong, "Leader-following coordination of multi-agent systems with coupling time delays," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 2, pp. 853–863, 2007.

[30] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[31] W. Ren and Y. Cao, *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer Science & Business Media, 2010.

[32] W. Ren, H. Chao, W. Bourgeous, N. Sorensen, and Y. Chen, "Experimental implementation and validation of consensus algorithms on a mobile actuator and sensor network platform," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pp. 171–176, IEEE, 2007.

[33] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.

[34] H. K. Khalil, *Noninear Systems*. Prentice-Hall, New Jersey, 2002.

[35] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *Journal of process control*, vol. 10, no. 4, pp. 363–385, 2000.

[36] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of time-delay systems*. Springer Science & Business Media, 2003.

[37] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2013.

[38] J. J. Craig, *Introduction to robotics: mechanics and control*, vol. 3. Pearson Prentice Hall Upper Saddle River, 2005.

[39] T. S. Alderete, "Simulator aero model implementation," *NASA Ames Research Center, Moffett Field, California*, 1995.

[40] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, 2011.

[41] T. Madani and A. Benallegue, "Backstepping control with exact 2-sliding mode estimation for a quadrotor unmanned aerial vehicle," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 141–146, IEEE, 2007.

[42] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 3255–3260, IEEE, 2006.

[43] T. Madani and A. Benallegue, "Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles," in *American Control Conference, 2007. ACC'07*, pp. 5887–5892, IEEE, 2007.

[44] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear h-infinity control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.

[45] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four rotor mini-rotorcraft," in *European Control Conference (ECC), 2003*, pp. 3123–3128, IEEE, 2003.

[46] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, pp. 72–77, IEEE, 2002.

[47] R. Xu and U. Ozguner, "Sliding mode control of a quadrotor helicopter," in *Decision and Control, 2006 45th IEEE Conference on*, pp. 4957–4962, IEEE, 2006.

[48] Y. Wang, Q. Wu, and Y. Wang, "Distributed consensus protocols for coordinated control of multiple quadrotors under a directed topology," *IET Control Theory & Applications*, vol. 7, no. 14, pp. 1780–1792, 2013.

[49] Y. Wang, Q. Wu, Y. Wang, and D. Yu, "Consensus algorithm for multiple quadrotor systems under fixed and switching topologies," *Journal of Systems Engineering and Electronics*, vol. 24, no. 5, pp. 818–827, 2013.

[50] Y. Wang, Q. Wu, and Y. Wang, "Containment control for multiple quadrotors with stationary leaders under directed graphs," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 2781–2786, IEEE, 2012.

[51] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems*, vol. 7, no. 3, pp. 419–428, 2009.

[52] D. Lara, A. Sanchez, R. Lozano, and P. Castillo, "Real-time embedded control system for vtol aircrafts: Application to stabilize a quad-rotor helicopter," in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pp. 2553–2558, IEEE, 2006.

[53] U. Pilz, A. P. Popov, and H. Werner, "Robust controller design for formation flight of quad-rotor helicopters," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 8322–8327, IEEE, 2009.

[54] U. Pilz and H. Werner, "Convergence speed in formation control of multi-agent systems-a robust control approach," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 6067–6072, IEEE, 2013.

[55] Z. Gosiewski and L. Ambroziak, "Formation flight control scheme for unmanned aerial vehicles," *Lecture Notes in Control and Information Science*, vol. 422, pp. 331–340, 2012.

[56] Y. Kuriki and T. Namerikawa, "Formation control with collision avoidance for a multi-uav system using decentralized mpc and consensus-based control," in *Control Conference (ECC), 2015 European*, pp. 3079–3084, IEEE, 2015.

[57] Y. Kuriki and T. Namerikawa, "Formation control of uavs with a fourth-order flight dynamics," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 6706–6711, IEEE, 2013.

[58] H. Bouadi, A. Aoudjif, and M. Guenifi, "Adaptive flight control for quadrotor uav in the presence of external disturbances," in *Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on*, pp. 1–6, IEEE, 2015.

[59] J.-J. Xiong and G. Zhang, "Sliding mode control for a quadrotor uav with parameter uncertainties," in *Control, Automation and Robotics (ICCAR), 2016 2nd International Conference on*, pp. 207–212, IEEE, 2016.

[60] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding mode control based on backstepping approach for an uav type-quadrotor," *World Academy of Science, Engineering and Technology*, vol. 26, no. 5, pp. 22–27, 2007.

[61] H. Bouadi and M. Tadjine, "Nonlinear observer design and sliding mode control of four rotors helicopter," *Vorld Academy of Science, Engineering and Technology*, vol. 25, pp. 225–229, 2007.

[62] A. Villanueva, B. Castillo-Toledo, E. Bayro-Corrochano, L. F. Luque-Vega, and L. E. González-Jiménez, "Multi-mode flight sliding mode control system for a quadrotor," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pp. 861–870, IEEE, 2015.

[63] W. Ren, K. L. Moore, and Y. Chen, "High-order and model reference consensus algorithms in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 678–688, 2007.

[64] L. Sheng, Y.-J. Pan, and X. Gong, "Consensus formation control for a class of networked multiple mobile robot systems," *Journal of Control Science and Engineering*, vol. 2012, p. 1, 2012.

Appendix A

Sample Matlab Code and Simulink Block Diagram

A.1 Simulink Block Diagram

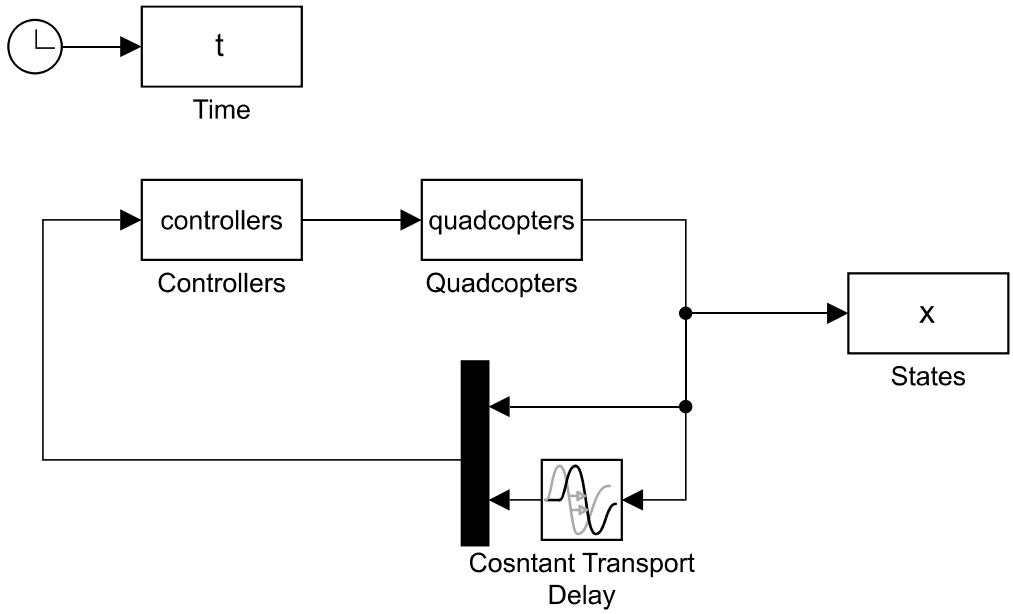

Figure A.1: Simulink diagram for multiple quadcopter system

A.2 Quadcopter S-function

```
function [sys,x0,str,ts] = quadcopters(t,x,u,flag,init,Adj)

%DEFINE GLOBAL VARIABLES THAT WILL BE SHARED B/C FUNCTIONS
global g Num
g = 9.8;            % Gravity
[Num,~]=size(Adj);      % # of agent
```

```
%DISPATCH THE FLAG

switch flag
case 0
[sys,x0,str,ts]=mdlInitializeSizes(init); % Initialization
case 1
sys = mdlDerivatives(t,x,u); % Calculate derivatives
case 3
sys = mdlOutputs(t,x); % Calculate outputs
case { 2, 4, 9 }
sys = []; % Unused flags
otherwise
error(['Unhandled flag = ',num2str(flag)]); % Error handling
end
end % End of quadcopters

function [sys,x0,str,ts] = mdlInitializeSizes(init)
% RECALL THE GLOBAL VARIABLE
global Num
% Call simsizes for a sizes structure, fill it in and convert it
% to a sizes array.
sizes = simsizes;
sizes.NumContStates  = 12*Num; % number of continuous states
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 12*Num;
sizes.NumInputs      = 4*Num;
sizes.DirFeedthrough = 0; % direct means y=k*u,and set = 1
sizes.NumSampleTimes = 1; % number of sample times, 1
%This passes the information in the sizes structure to sys
sys = simsizes(sizes);
```

```
% Initialize the initial conditions.
x0 = init; %[x1_0;x2_0;...;xn_0]
% str is an empty matrix, reserved for future use
str = [];
% Generic timesample, [SampleTime Offset], [0 0] for continuous
ts = [0 0];
end % End of mdlInitializeSizes.

% Calculate the state derivatives for the next timestep
function sys = mdlDerivatives(t,x,u)
%RECALL THE GLOBAL VARIABLE, THIS IS NECESSARY
global g Num

%DEFINED AS EMPTY MATRIX FOR LOOPING
sys = [];

%LOOPING
for i=1:Num
%DISSECT x INTO X{i}, cell array
X{i} = x(12*(i-1)+1:12*i);

%DISSECT u INTO U{I}
U{i} = u(4*(i-1)+1:4*i);

%EXTRACT STATES FROM x{i}
Z{i} = X{i}(1:3);   % position
N{i} = X{i}(4:6);   % RPY angles
V{i} = X{i}(7:9);   % velocity
O{i} = X{i}(10:12); % RPY angles derivative

%EXTRACT ROLL/PITCH/YAW ANGLES
Phi{i} = N{i}(1);    % roll
```

```
The{i} = N{i}(2);    % pitch
Psi{i} = N{i}(3);    % yaW

%RIGID BODY DYNAMIC MODEL
Dz{i} = V{i}; % Velocity
Dn{i} = O{i}; % RPY Derivative
Dv{i} =
[U{i}(1)*(cos(Psi{i})*sin(The{i})*cos(Phi{i})...
...+sin(Psi{i})*sin(Phi{i}));
U{i}(1)*(sin(Psi{i})*sin(The{i})*cos(Phi{i})...
...-cos(Psi{i})*sin(Phi{i}));
U{i}(1)*cos(The{i})*cos(Phi{i})-g                          ];
Do{i} = U{i}(2:4); %
sys = [sys;Dz{i};Dn{i};Dv{i};Do{i}];% state derivative vector
end

end %End of mdlDerivatives.

%Calculate the output vector for this timestep
function sys = mdlOutputs(t,x)
sys = x; % output is the system state
end
% End of mdlOutputs.
```

A.3 Controller S-function

```
function [sys,x0,str,ts] = controllers(t,x,u,flag,Adj,l,k,b)

%DEFINE GLOBAL VARIABLES THAT WILL BE SHARED B/C FUNCTIONS
global  Num g L
[Num,~]=size(Adj);      % # of agent,
g = 9.8;              % Gravity
```

```
L = diag(sum(Adj,2))-Adj;   % L is the Laplacian matrix

%DISPATCH THE FLAG
switch flag
case 0
[sys,x0,str,ts]=mdlInitializeSizes; % Initialization
case 3
sys=mdlOutputs(t,x,u,l,k,b); % Outputs
case {1, 2, 4, 9 }
sys = []; % Unused flags
otherwise
error('Simulink:blocks:unhandledFlag', num2str(flag));
end
end % End of controllers


function [sys,x0,str,ts]=mdlInitializeSizes

global Num

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4*Num;
sizes.NumInputs      = 12*Num*2; % states & delayed states
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);

x0  = []; % initialize the initial conditions
str = []; % str is always an empty matrix
ts  = [0 0]; % initialize the array of sample times
```

```
end % End mdlInitializeSizes

% mdlOutputs
% Return the block outputs.
function sys=mdlOutputs(t,x,u,l,k,b)
%RECALL THE GLOBAL VARIABLE
global Num g L

%INITIALIZING EMPTY MATRIX FOR STATES LOOPING
sys = [];
v = []; w = []; % v(column vector)
Dv =[]; Dw = []; % \dot{v}
vd = []; wd = []; % v(t-\tau)(...)
Dvd = []; Dwd = []; % \dot{v(t-\tau)}(...)
x_pos = []; y_pos = []; z_pos = []; % x,y,z(...)
x_posd = []; y_posd = []; z_posd = []; % x(t-tau)(...)
Dx =[]; Dy =[]; Dz = []; % \dot{x}(...)
Dxd =[]; Dyd =[]; Dzd = []; % \dot{x(t-\tau)}(...)

%ISOLATING STATES AND DELAYED STATES FROM INPUT
states = u(1:12*Num);
statesd = u(12*Num+1:end); % constant delayed states

%LOOPING AND EXTRACTING ALL THE STATES AND DELAYED STATES
%DEFINE UPPER CASE AS CELL ARRAY, LOWER CASE AS VECTOR
for i = 1:Num
%EXTRACT STATES AND DELAYED STATES FOR ith Agent
X{i} = states(12*(i-1)+1:12*i);
Xd{i} = statesd(12*(i-1)+1:12*i);

%EXTRACT STATES FROM X{i}
Z{i} = X{i}(1:3);   % position
```

```
N{i} = X{i}(4:6);   % RPY angles
Vel{i} = X{i}(7:9);   % velocity
O{i} = X{i}(10:12); % RPY angles derivative

%EXTRACT STATES FROM Xd{i}
Zd{i} = Xd{i}(1:3);   % position delayed
Nd{i} = Xd{i}(4:6);   % RPY angles ...
Veld{i} = Xd{i}(7:9);   % velocity ...
Od{i} = Xd{i}(10:12); % RPY angles derivative ...

%EXTRACT ROLL/PITCH/YAW ANGLES
Phi{i} = N{i}(1);    % roll
The{i} = N{i}(2);    % pitch
Psi{i} = N{i}(3);    % yaw
DPhi{i} = O{i}(1);    % roll derivative, D as derivative
DThe{i} = O{i}(2);    % pitch ...
DPsi{i} = O{i}(3);    % yaw ...

%EXTRACT ROLL/PITCH/YAW ANGLES
Phid{i} = Nd{i}(1);    % roll delayed
Thed{i} = Nd{i}(2);    % pitch ...
Psid{i} = Nd{i}(3);    % yaw ...
DPhid{i} = Od{i}(1);    % delayed roll derivative
DThed{i} = Od{i}(2);    % pitch ...
DPsid{i} = Od{i}(3);    % yaw ...

%EXTRACT v,w,vd,wd, (vd means delayed v)
V{i} = g*tan(The{i});
W{i} = -g*tan(Phi{i});
Vd{i} = g*tan(Thed{i});
Wd{i} = -g*tan(Phid{i});
```

```
%COMPUTING DELAYED/NONDELAYED DERIVATIVE OF Vd,Wd
DV{i} = g*sec(The{i})^2*DThe{i};
DW{i} = -g*sec(Phi{i})^2*DPhi{i};
DVd{i} = g*sec(Thed{i})^2*DThed{i};
DWd{i} = -g*sec(Phid{i})^2*DPhid{i};

%EXTART ALL THE STATES VECTORS USED IN THE CONTROLLER
v = [v;V{i}]; w = [w;W{i}]; % v...
Dv = [Dv;DV{i}]; Dw=[Dw;DW{i}]; % \dot{v}...
vd = [vd;Vd{i}]; wd = [wd;Wd{i}]; % v(t-\tau)...
Dvd = [Dvd;DVd{i}]; Dwd = [Dwd;DWd{i}]; % \dot{v(t-\tau)}...
x_pos = [x_pos;Z{i}(1)]; y_pos = [y_pos;Z{i}(2)];
z_pos = [z_pos;Z{i}(3)];
x_posd = [x_posd;Zd{i}(1)]; y_posd = [y_posd;Zd{i}(2)];
z_posd = [z_posd;Zd{i}(3)];
Dx =[Dx;Vel{i}(1)]; Dy =[Dy;Vel{i}(2)];
Dz = [Dz;Vel{i}(3)];
Dxd =[Dxd;Veld{i}(1)]; Dyd =[Dyd;Veld{i}(2)];
Dzd = [Dzd;Veld{i}(3)];
end

%CONTROLLER DESIGN
for i=1:Num
U{i}(1) = l{1}*[z_pos(i);Dz(i)]...
...-k{1}*(kron(eye(2),L(i,:))*[z_posd;Dzd]);
U{i}(1) = (U{i}(1)+g)/(cos(Phi{i})*cos(The{i}));

U{i}(2) = l{2}*[y_pos(i);Dy(i);w(i);Dw(i)]...
...-k{2}*(kron(eye(4),L(i,:))*[y_posd;Dyd;wd;Dwd]);
U{i}(2) = (U{i}(2)-2*Dw(i)^2*w(i)/(g^2+w(i)^2))/(-(g+w(i)^2/g));

U{i}(3) = l{2}*[x_pos(i);Dx(i);v(i);Dv(i)]...
```

```
...-k{2}*(kron(eye(4),L(i,:))*[x_posd;Dxd;vd;Dvd]);
U{i}(3) = (U{i}(3)-2*Dv(i)^2*v(i)/(g^2+v(i)^2))/(g+v(i)^2/g);

U{i}(4) = -b(1)*Psi{i}-b(2)*DPsi{i}; % local controller for psi
sys = [sys;[U{i}(1);U{i}(2);U{i}(3);U{i}(4)]];
end
end %End mdlOutputs
```

Appendix B

Simplified Quadcopter Model

B.1 Partially Linearized Model

$$\begin{cases} \ddot{x} &= -\frac{T}{m}\sin\theta \\ \ddot{y} &= \frac{T}{m}\cos\theta\sin\phi \\ \ddot{x} &= \frac{T}{m}\cos\theta\cos\phi - g \\ \ddot{\phi} &= \tilde{\tau}_\phi \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\psi} &= \tilde{\tau}_\psi \end{cases} \tag{B.1}$$

B.2 Fully Linearized Model

$$\dot{x} = A_L x + B_L u, \tag{B.2}$$

where

$$A_L = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B_L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.3}$$

Appendix C

Rotation Matrix in Left-handed Coordinate System

Consider a Left-handed coordinate system as shown below The elemental rotation

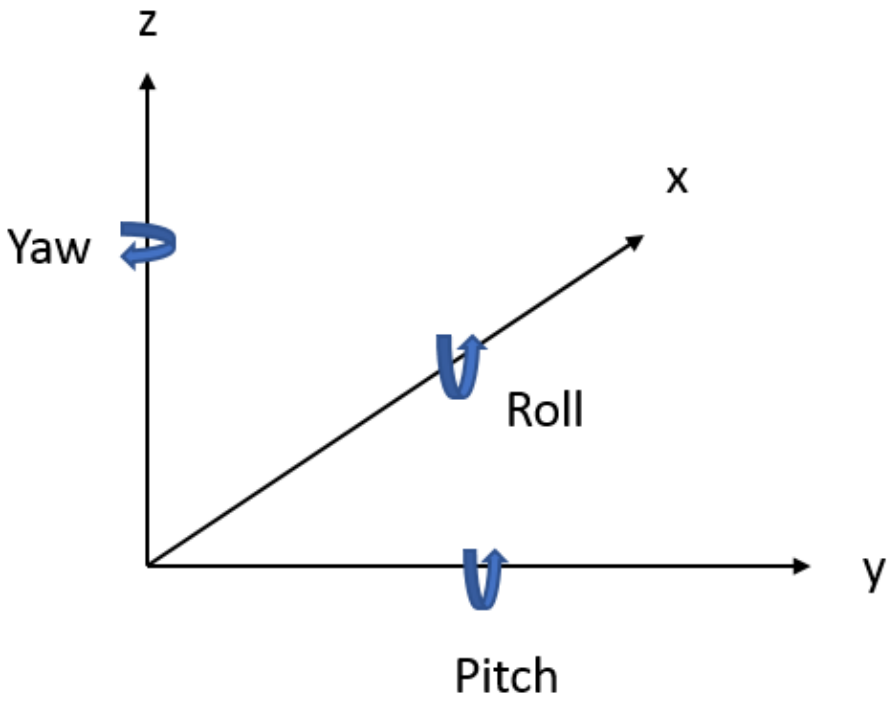

Figure C.1: Left-handed coordinate system.

matrices are given as

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix}, R_y(\theta) = \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix}, R_z(\psi) = \begin{bmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then under ZYX (fixed yaw-pitch-roll angles) or $X'Y'Z'$ (Euler roll-pitch-yaw angles) frame, the rotation matrix R is given as

$$\begin{aligned} R &= R_x(\phi)R_y(\theta)R_z(\psi) \\ &= \begin{bmatrix} C_\theta C_\psi & S_\psi C_\theta & -S_\theta \\ C_\psi S_\theta S_\phi - S_\psi C_\phi & S_\psi S_\theta S_\phi + C_\psi C_\phi & C_\theta S_\phi \\ C_\psi S_\theta C_\phi + S_\psi S_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi & C_\theta C_\phi \end{bmatrix}. \end{aligned}$$

Appendix D

Author's Publication List

Peer-Reviewed

Z. Huang and Y.J. Pan, "Observer based Leader Following Consensus for Multi-Agent Systems with Random Packet Loss", In *Control Technology and Applications (CCTA), 2017 IEEE Conference on*, pp. 1698-1703, IEEE, 2017.
Y.J. Pan, H. Werner, **Z. Huang** and M. Bartel, "Distributed Cooperative Control of Leader-Follower Multi-Agent Systems under Packet Dropouts for Quadcopters", *Systems and Control Letters (SCL)*, 2017.

Other Publications

Z. Huang and Y.J. Pan, "Observer based Leader Following Consensus Control for Multi-Agent Systems with Random Packet Loss", (Presentation) *2017 Dalhousie University Mechanical and Material Engineering Conference (DUMMEC)*, 2017.
Z. Huang and Y.J. Pan, "Formation Control of Multi-agent Systems with Applications to Multiple Quadcopters", (Poster) *2016 Dalhousie University Mechanical Engineering Research Conference (MERC)*, 2016.