DESIGN AND IMPLEMENTATION OF CONSENSUS IN MULTI-AGENT SYSTEMS WITH INPUT AND COMMUNICATION TIME-DELAYS

A

Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

By

YANUMULA VENKATA KARTEEK

Department of Electronics and Electrical Engineering

Indian Institute of Technology Guwahati

Guwahati - 781 039, INDIA.

March, 2018.

DESIGN AND IMPLEMENTATION OF CONSENSUS IN MULTI-AGENT SYSTEMS WITH INPUT AND COMMUNICATION TIME-DELAYS

YANUMULA VENKATA KARTEEK

Certificate

This is to certify that the thesis entitled "**DESIGN AND IMPLEMENTATION OF CONSENSUS IN MULTI-AGENT SYSTEMS WITH INPUT AND COMMUNICATION TIME-DELAYS**", submitted by **Yanumula Venkata Karteek** (10610219), a research scholar in the *Department of Electronics & Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:
Guwahati.

Dr. Indrani Kar
Dept. of Electronics & Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781039, Assam, India.

Dated:
Guwahati.

Prof. Somanath Majhi
Dept. of Electronics & Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781039, Assam, India.

Acknowledgements

I would like to express my wholehearted gratitude to my supervisors, Dr. Indrani Kar and Prof. Somanath Majhi, for their excellent guidance and support throughout the course of thesis work. I am very thankful to their patience in correcting my mistakes in the thesis work as well as in all my manuscripts.

I would like to thank my doctoral committee members Prof. Harsha B. Nemade, Dr. Pravin Kumar and Dr. Atanu Banerjee for evaluating the progress of my work. Their valuable suggestions are very helpful in the thesis work. I would like to thank Dr. Krishnaswamy Srinivasan for the fruitful discussions. I would like to thank all the faculty members for their support at various times. My special thanks to Mr. Sidananda Sonawal and Mr. Syyed Mazid for their help while developing the hardware setup. I would like to thank all the members responsible for establishing and maintaining Control & instrumentation Laboratory, which served as a platform to perform my work.

During my entire stay at IIT Guwahati, friends played a great role in helping me stay relaxed. I would like to thank Mridul, Tousif, Arghya and Saurabh for helping me to stay focussed and for the discussions on various topics. My thanks to other lab members (but not limited to) Sanjoy, Mandar and Karnika. With Prathap, Pavan, Vinay and Sivanshu, the hostel stay has been very pleasant. I would like to thank my other friends (but not limited to) Sravani, Rama Krishna, Dileep, Swetha, Harathi, Satish, Varma, Dhamodar, Satheesh and Ranganath. I would like to extend my special thanks to Dileep and Sivanshu for being my badminton partners.

I am grateful and indebted to IIT Guwahati and MHRD for providing all the facilities, research funding and financial assistance. I would like to thank my parents for their constant support and patience during my thesis work. Finally, I would like to thank the God for giving this opportunity.

(Yanumula Venkata Karteek)

Abstract

Achieving consensus is a major problem in multi-agent systems(MASs). Analysis, design and implementation of consensus in MASs is the main focus of this thesis. Motivated by recent surge in distributed control applications with networked mobile robots and sensor networks which are MASs, consensus problems in autonomous mobile robots and a sensor network are researched. In particular, faults in communication like complete loss and time-delay effects on consensus are studied. Complete loss of communication will prevent the multi-agent system in reaching consensus. Time-delay within certain limit deteriorate the performance of system by increasing the convergence time. Beyond the limit, the system never reaches consensus.

Two novel algorithms, namely “back-tracking” and “history following”, to reach consensus in case of communication loss for a multi-agent system(MAS) with switching topologies are proposed for mobile agents. A strongly connected topology is considered to make decision on usage of “back-tracking” algorithm whenever an agent loses communication. It ensures the mobile multi-agent consensus with intermittent communication loss by changing path of the agents to maintain a strongly connected topology. In the case of “history following” algorithm, the agents store past data and continue with most recent target position even after the loss of communication. Since the communication loss that we have considered is intermittent at few places, the agents regain communication after some displacement. Then the data is updated with new calculation of target position and stored for further usage that might arise. A simple obstacle avoidance algorithm also used while simulating a network of six mobile agents with intermittent communication loss and static obstacles. The pros and cons of both algorithms are discussed. Hardware implementation is performed for a three robot system without obstacles along with corresponding simulation to demonstrate the effectiveness of the algorithms.

In the case of sensor network application, necessary and sufficient conditions are derived for consensus with saturated agents, asynchronous input and communication time-delays, which are sometimes practically unavoidable. A multi-agent system of linear second order agents

with saturation is considered. The distributed system is connected under fixed directed graph coupled with input and communication time-delays. Describing function analysis is used to approximate each nonlinear agent into cascade of linear and nonlinear elements. Nyquist stability criterion is used for analysing stability of the linear element, thereby estimating the existence of limit cycles. Explicit necessary and sufficient conditions are derived for stability of the linear element. Instability of limit cycles is estimated using those conditions, implying the consensus.

Further, stability analysis is performed using Lyapunov-Krasovskii functions instead of Nyquist stability criterion with multiple control laws. A set of linear matrix inequalities (LMIs) are derived to prove the stability of linear element. The stability range of time-delays are obtained using expressions derived from both Nyquist stability criterion and Lyapunov-Krasovskii approaches. A comparison of pros and cons with respect to time-delay ranges, applicability and computational complexity is presented. Simulations and corresponding hardware implementation are performed on a network of four agents and five agents to support the theoretical results with multiple control laws. Furthermore, a general nonlinear input is considered as input to a multi-agent system that is connected by an undirected network. It has been proved that, consensus can be achieved with any nonlinear function g that is symmetric to origin and $xg(x) > 0$. Simulations and implementations are performed on a network of five agents.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE AND MATHEMATICAL NOTATIONS

MAS	Multi-agent system
LMI	Linear matrix inequality
PD	Proportional derivative controller
PID	Proportional integral derivative controller
$\mathscr{G}$	Graph to represent communication topology
V	Vertex set representing nodes of graph
E	Edge set representing communication among agents
$\mathscr{A}$	Adjacency matrix of graph
τ_1	Input time-delay
τ_2	Communication time-delay

CHAPTER 1

INTRODUCTION

1.1 Background

Control theory and its applications have made significant progress in the last century. During the World War-II, advancements in control theory happened at a rapid pace. Classical control uses frequency domain approach and transfer functions while analysing single-input-single-output (SISO) systems. For multiple-input-multiple-output (MIMO) system, state-space approach is being used. Due to the versatility of state-space approach, it has been used in modern control for analysing SISO, MIMO, linear and nonlinear systems. In the past several decades, modern control theory and automation were further advanced by addressing the requirements in various sectors. From the past few decades, automation is receiving lot of interest from researchers and industry [3–6]. Automation and robotics are playing a pivotal role in the manufacturing and supply lines of chip design (microprocessors or microcontrollers), auto-mobile industry and so on.

Robots and software-bots are playing a major role in automation industry and have been tasked in large number of high end applications [7–11]. Robot assisted surgery to minimize the human intervention is receiving a lot of attention [12, 13]. Recovery time of patient and human errors are minimized with surgical robots [14, 15]. Unmanned vehicles are being used in applications where human intervention is not safe and/or not economical. Sending a robot to greater depths in an ocean is far safer and economical than sending a diver. Using a drone or robot is preferable in a war zone with an unpredictable enemy. Human errors are a major cause of accidents [16–18]. Driverless vehicles and semi-autonomous vehicles are being developed

to make the roads safer [19–21]. With applications in multiple domains, unmanned vehicles are technologically evolving with the fusion of various control strategies, sensors and data handling. Controlling a single vehicle has become relatively mature with the help of strategies like autopilot [22, 23]. The autopilot function of a modern aircraft is very advanced and it is getting developed continuously for application in driverless cars and semi-autonomous cars.

A team of robots working in a cooperative manner can be highly efficient compared to a single complex robot [24]. Map-making, scanning a cluttered environment etc., are performed efficiently by a team of robots [25–30] as depicted in Fig. 1.1. In the past two decades, control of multiple robots has received considerable attention [24, 31]. An agent can represent a robot, an unmanned vehicle (land, air, floating on water and underwater) [32–36], a spacecraft [37–40], an internet router [41], or even a sensor [42–44]. A system of multiple agents acting as a single unit with the help of communication is termed as a multi-agent system (MAS). Improvements in communication and computation technologies have contributed to generate interest on multi-agent systems among the research community. During the last two decades, multi-agent systems have received significant attention. Multi-agent systems have applications in robotics, sensor networks, computational economics [45–50] and so on.

Control and coordination are major problems in dealing with the multi-agent system while performing certain tasks. Control of the multi-agent system can be performed by two approaches, centralized approach and distributed approach. In centralized approach, there is a central station that controls the whole system. The basic assumption in the centralized approach is that the central station is sufficiently powerful to control the whole group of agents. In distributed approach, each agent has the freedom to take its own decisions and control itself, without the need of a powerful central station. The freedom in distributed approach is coupled with the increase in complexity of the system in terms of structure and organization. Although both the approaches are being used, distributed approach is advantageous. The drawbacks of centralized approach include, limited capability of central station, short range of wireless communication, narrow bandwidth etc.. The drawbacks can become bottlenecks when the centralized system is scaled up. If there is a failure in central station, the whole system is rendered

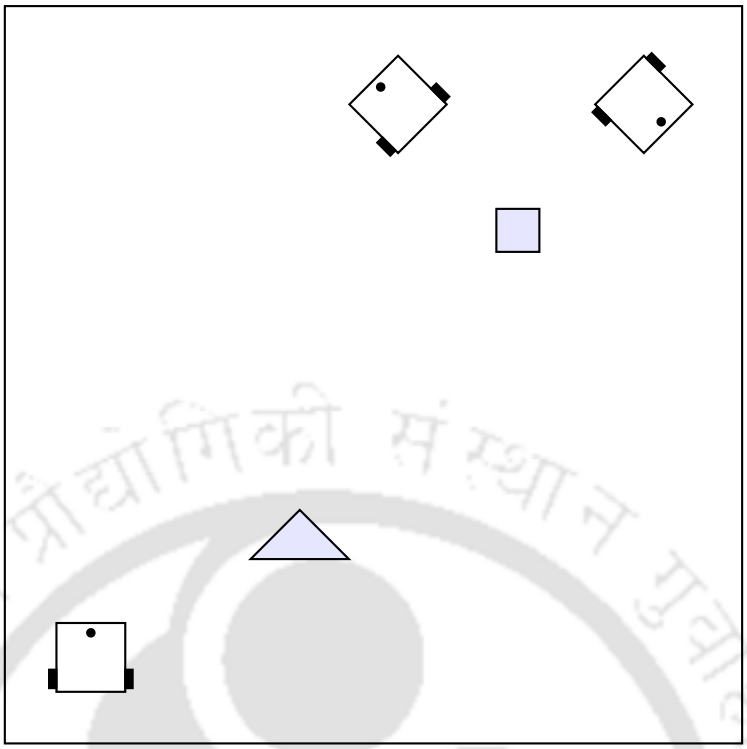


Figure 1.1: Arena with mobile agents and obstacles.

useless, whereas, in distributed system there is certain degree of redundancy, which makes the system robust and have greater fault tolerance. If one of the agents fail in a distributed system, the task can be re-distributed.

In a distributed approach, the main aim of a multi-agent system is to have the whole group of autonomous agents working in a cooperative manner [51]. Since the central station is absent in distributed approach, information sharing and communication play the central role of controlling the multi-agent system. Distributed approach has advantages like low operational cost, enhanced functionality, flexible scalability, fault tolerance and less system requirements. Due to these advantages, it has been receiving the attention from research community from last two decades [24, 31].

Stability of the system is dependent on how well the agents are connected with each other by a communication network. If the communication among agents is bidirectional, it is called undirected network and if it is unidirectional, it is called directed network. If the communication topology is not changing in the course of achieving a goal, the multi-agent system is said to have a fixed topology. If the communication topology is changing periodically or aperiodically,

it is called switching topology. The communication network topology of a multi-agent system can be fixed or switching. A lot of research on multi-agent systems has been focussed on the stability and various factors that affect the stability [52–56]. One such factor is time-delay, which is primarily of two types, communication time-delay and input time-delay. Communication time-delay is caused by the delay in receiving the information, input time-delay is due to computational burden on agents. Stability analysis of multi-agent systems can be performed with frequency domain approach and Lyapunov approach [57, 58].

In a multi-agent system, when the states agree on some common value, we call them to have reached consensus. Consensus in multi-agent systems are biologically inspired and have been reported initially by Tams Vicsek and Shochet [59]. The consensus problems in multi-agent systems are receiving lot of attention recently due to the applications in grouping, formation control, synchronization, etc.. A multi-agent system while grouping at a common position or while maintaining a formation, have to reach consensus on at least one variable. The variable can be position with or without separation, velocity or acceleration or any such variable depending on the requirement. We call the system to have reached consensus only when corresponding difference of all the state variables is zero. The agents exchange information locally and a cooperative distributed control strategy is implemented without the need of external supervision for reaching consensus. The consensus in multi-agent system is biologically inspired and follow the distributed approach to a large extent. In some of the applications, leader following and centralized control strategies are also used to guide agents [60].

1.2 A Survey of Previous Works

The research in multi-agent systems is inspired by the observations made by Tams Vicsek and Shochet [59]. The authors have investigated self-ordered motion in systems of particles with biologically motivated interactions. In this model, particles are driven with a constant absolute velocity. At each time-step, particles assume the average direction of motion of the particles in their neighbourhood with some random perturbation added. The algorithms proposed by

the authors consider the velocity of other particles in the neighbourhood. These algorithms are used extensively in the later articles with modifications according to various requirements. Ali Jadbabaie *et al.* [61] gave the theoretical explanation to the behaviour of particles reported in [59]. Convergence analysis is performed mathematically with the help of theorems, the results in [61] paved the way for further research in the domain of multi-agent systems.

The results reported in [61] are followed by an extensive analysis of multi-agent systems by Saber and Murray [57, 62] with time-delay effects taken into consideration. The subsequent research reported in the literature is extended to higher order agents based on the approach and results provided by the authors in [57, 61, 62]. In [62], linear and nonlinear consensus protocols are considered in a multi-agent system with time-delay. The network considered is an undirected one with distributed control law. In [57], the authors have further extended the results in [62] and analysed various cases with directed fixed topology, directed switching topology and undirected fixed topology with time-delays. The authors have provided a tight upper limit on the time-delay that the undirected fixed topology is able to tolerate. They proved that the maximum time-delay that can be tolerated by a network of integrators with the linear consensus protocol is inversely proportional to the largest eigenvalue of a network topology or the maximum degree of the nodes of a network. Agents considered in the analysis are first order and single integrator type and time-delay is uniform. It has been shown that there is a tradeoff between time-delay robustness and linear protocol performance. Another tradeoff is between low communication cost and high performance.

A linear time varying system is considered in [63], where the authors derived sufficient conditions guaranteeing uniform exponential convergence for consensus of the multi-agent system. They have considered a linear time varying system with communication time-delay without considering any input time-delay. Few theoretical results are obtained to prove the convergence of multi-agent system. Guangming Xie and Long Wang [64, 65] have considered a multi-agent system second order dynamics and time-delays are not considered. In [64], a fixed topology is considered and in [65], a switching topology is considered. The analysis is performed with the help of Lyapunov functions defined in terms of disagreement in state variables. In the review

paper by Olfati-Saber *et al.* [66], various algorithms used to reach consensus of first order systems in continuous as well as discrete time systems with and without time-delay are discussed. They have given a brief overview and mainly focused on their past research in [57]. Order of the agents exceeding two are considered by Wie Ren *et al.* [67] and sufficient condition for convergence is given, but communication time-delays are not considered. Multi-vehicle cooperative control strategy, that mimics flocking behaviour in birds is taken into account. The bird flock, sometimes change the direction abruptly due to danger perception by one of the birds. For such abrupt changes to be implemented in unmanned vehicles, second-order model is not sufficient. The authors in [67] have considered a system with order of each agent exceeding two (≥ 3). This flocking behavior can be applied in unmanned air vehicles or with autonomous underwater vehicles working in a hostile environment, where abrupt changes in heading and acceleration are required.

In [68], the authors have considered nonlinear input for a multi-agent system with second order linear agents. An undirected network without time-delays is studied and the existing results are extended for a system with input saturation. A discrete model with time-delays is studied by Bliman *et al.* [69]. The analysis is based on an enlarged system without delay, new virtual agents are introduced to compensate the delay. The virtual agents are non-computing and add a single step delay. The number of virtual agents is calculated on delay requirement. Cao *et al.* [70] have reported consensus of various versions the flocking problems with emphasis on the underlying graph structures. Few special graphs like quotient graph and an agent subgraph are studied. These graphs are used in the analysis of the consensus problem with delays.

Time-varying delays are analysed by Feng Xia and Long Wang [71] with continuous-time agents updated at discrete times. The update times of each agent are independent of each other and asynchronous. The communication topology is switching and delays are bounded with time-varying nature. In [72], the consensus algorithm for discrete-time second-order multi-agent systems with diverse communication delays is considered. Leader following control strategy is proposed with a static leader and the results are extended to the leaderless consensus problem of the discrete-time second-order multi-agent systems. Liu *et al.* [73] have analysed

average consensus problem for first order multi-agents of undirected network with limited data rate and communication time-delays. They proved that a finite-level quantizer for a connected network is able to solve the average consensus problem when the time-delays are bounded. In [74], the authors have considered autonomous mobile agents with double-integrator dynamics. The interconnection delays are time-varying and communication topology is directed and a leaderless control is considered. The consensus stability is studied with both fixed and switching interconnection topologies.

Sampled control is used by Liu *et al.* [75] to analyse a continuous-time multi-agent system. The time-delays considered in the article are assumed to be uncertain, time-varying and both larger and smaller than the sampling interval. The authors in [76] have reported results on a directed and switching network with communication delays. The boundedness of a network with first order agents is proved and extended to higher order systems. Meng *et al.* [77] have studied leader-following and leaderless consensus algorithms coupled with uneven input and communication time-delays. The stability analysis is performed by using both Lyapunov approach and Nyquist stability criterion. Multiple cases involving static virtual leader, dynamic virtual leader, with full and partial access to the virtual leader are analysed. The network topology considered is directed and fixed. Higher order systems with undirected topology are studied in [78]. A sufficient and necessary condition for reaching consensus is obtained, but the affect of time-delay is ignored. A nearest-neighbour rule is used for higher-order discrete time agents in [79]. Switching topologies with communication time-delays are analysed, however, input time-delays are ignored.

Frequency domain approach is used in [58] to derive stability conditions for a second-order multi-agent system with time-delays. A fractional order system is considered in [80, 81]. Nyquist stability criterion is used in [80] and Lyapunov approach is used in [81] to obtain necessary and sufficient conditions to reach consensus. A second-order multi-agent system with time-varying and multiple asynchronous time-delays are considered in [82, 83]. Event triggered consensus is explored in [84, 85] and the consensus problems with time-delayed impulsive controls and periodic sampling are studied in [86].

Recently, the nonlinearities are being taken into account since most of the practical systems are inherently nonlinear in nature [85,87,88]. In [89], pulse width modulation is used to control the multi-agent system. The control is inherently nonlinear with user defined maximum magnitude based on saturation constraints of the agents. Actuator saturation is commonly seen in many practical scenarios where extremely large control input is not feasible, like the velocity of a mobile robot cannot go beyond certain limit. Actuator saturation is receiving attention recently [90–96]. Nonlinearities like saturation have been considered in multi-agent systems with first-order dynamics in [90] and second order dynamics in [91–96]. Various control laws with differential gain feedback [91, 92], observer-based adaptive consensus [93], containment control [94] on linear agents and nonlinear agents [95] are studied to solve the consensus problem with the help of Lyapunov-Krasovskii analysis. Considering the time-delays in nonlinear multi-agent systems started recently, You *et al.* [96] have considered a second order system with switching topologies and a single time-varying delay.

Most of the work, discussed above, is confined to simulation rather than implementation. Simulations are performed with different algorithms which deal with or without time-delays [57, 65, 66, 68, 77, 79]. The implementations are done without considering time-delays in the algorithms [97–99]. In [97], Mondada *et al.* have discussed about self-configurable swarm robots which are equipped with 50 sensors and controlled by Intel Xscale processor with 13 PIC slave processors. All the robots are autonomous with decentralized control and perform different tasks as shown in the results of the paper. Time-delays are not considered. A map-making multi-agent system is studied by Churavy *et al.* [100] to update the map of a given space. The robots are controlled by centralized mechanism and they use metric localization which use worldcentric co-ordinates to localize themselves on a plane. In [98], the authors explain how a swarm of robots are used to inspect a turbine system using the existing path-planning and co-ordination algorithms. They don't deal with time delays since they use reliable 2.4GHz radio module for communication. Localization problem is studied in [101], where the authors implemented Extended Kalman filter technique to localize the robots on a plane and the algorithms are implemented on LEGO NXT robot kits. Chang *et al.* [99] implemented a

different approach of Fuzzy Sliding mode to control the multi-agent systems. However, time-delays are not considered while modelling the system.

1.3 Motivation

From the above discussion on previous works, it can be observed that there is little focus on hardware implementation of multi-agent systems. Multiple robots working in a cluttered environment has applications in map-making, surveying, search and rescue operations, etc.. While working in a cluttered environment with limited resources, complete loss of communication may be encountered frequently. In wireless communication, blind spots can happen due to various factors like reflection, refraction, diffraction, absorption etc.. When a robot enters into one such blind spot, the robot is affected by complete loss of communication. Robots getting lost while performing a task is not desirable. This triggered the need for developing strategies to counter the complete loss of communication and led to a motivation behind developing such algorithms.

Moreover, majority of the research is confined to linear agents with time-delays. Nonlinear agents affected by time-delays have received very little attention in the literature, except for a small number of articles [96] with simulation results. Implementation with time-delay effects has not received enough attention. Nonlinearities are common in mobile agents and actuator saturation is the most frequent hard nonlinearity affecting them. For example, acceleration of an agent is constant over certain range and cannot be maintained after the agent attains its maximum velocity.

Recently, saturation nonlinearity is receiving considerable attention. Li *et al.* [90] considered a first order system with saturation and without time-delays. For second order agents with saturation and without time-delay, a differential gain feedback control is used by authors in [91,92]. Adaptive control laws with observer are used by Chu *et al.* [93] and nonlinear agents are considered by Cui *et al.* [95]. The effects of synchronous time-delays are taken into consideration by You *et al.* [96] for a network of second order saturated agents. It is evident from

the literature that, there is very little focus on consensus of second order saturated multi-agent system with asynchronous communication and input time-delays. This created an interest in considering a multi-agent system with asynchronous time-delays and hard saturation nonlinearities.

After the results in [68], there is little focus on general nonlinear input to a linear system. There are many nonlinear functions other than tanh(), that give saturated input. Moreover, heterogeneous systems with nonlinear input have not been investigated [102]. This motivated us to consider a general nonlinear input for achieving consensus in a multi-agent system.

1.4 Problem statement

According to the earlier discussion on previous works, the simulation work is done with and without time-delayed cases, but the implementation is done only without time-delay till now. Even in the simulation work, time-delayed cases are not giving satisfactory results [77, 79] with the existing algorithms, especially in the event of complete loss of communication. One of the objectives of this thesis is to develop algorithms for addressing the issue of communication loss and test it on mobile robots. We aim to test the developed algorithms on a simulation platform and implement on a multi-agent system with mobile robots. It is expected that some of the mobile robots are developed in laboratory to have more control on their motion. The target is to cross-check the results of both simulation and implementation for testing the effectiveness of algorithms.

The second objective it to perform stability analysis of a saturated multi-agent system with different approaches. By using the theoretical results of the stability analysis, we aim to obtain the range of time-delays that the network can tolerate. The theoretical results will be tested with simulation of different communication network topologies. Coming to implementation part, only cases without time-delay are considered till now [97–99]. The goal is to develop a simple multi-agent system of non-mobile agents, which is analogous to a sensor network. The results obtained from the stability approach will be tested on the hardware setup and also

cross-checked with simulation results. We aim to perform a comparison on advantages and disadvantages of the different approaches used in the stability analysis.

The third objective is to analyse a general nonlinear input applied to heterogeneous linear multi-agent system for reaching consensus. The general nonlinear function considered here is involved of a $tanh()$ function along with velocity saturation in the control law. The goal is to prove the convergence of heterogeneous multi-agent system with such input. The heterogeneous agents have different maximum velocities and various scaling factors for control inputs. We aim to demonstrate the effectiveness of theoretical results by performing simulations and implementing on the hardware setup of non-mobile multi-agent system.

1.5 Contribution of the Thesis

Two algorithms namely, back-tracking and history following algorithms are proposed to overcome the communication loss in multi-agent systems. The algorithms are tested with simulations and implemented on a hardware system. The hardware system consists of three robots, two of them are developed in laboratory. It has been observed from the results, that the algorithms work effectively in tackling the total loss of communication. With back-tracking algorithm, the multi-agent system is able to reach consensus by diverting the mobile robots when there is a loss of communication. With history following algorithm, the consensus is achieved in most of the cases. Consensus is not achieved with history following algorithm in few special cases, the explanation is provided for such behaviour with the help of a simulation. Back-tracking algorithm has larger convergence time compared to history following algorithm, but the consensus is always guaranteed. Advantages and disadvantages of both the algorithms are explained. Tackling the complete loss of communications and corresponding methods to deal with the issues received very little attention in the literature. The algorithms try to solve the practical problems.

Nonlinear multi-agent system is analysed with the help of describing function analysis to rearrange the system as a separate linear and nonlinear blocks. Asynchronous time-delays are considered while analysing the multi-agent system. When dealing with time-delays, mathemat-

ical analysis of nonlinear system is highly complex. With the help of describing function, the analysis is simplified to a great extent. Two approaches are used to estimate the existence of limit cycles in the multi-agent system. One of them is Nyquist approach and the other is Lyapunov approach. The results are tested with simulations and implementations on a network of agents with different topologies. It has been observed that Nyquist approach is not applicable to wide range of control laws, but the Lyapunov approach is applicable. The second observation is that the Lyapunov approach is more conservative compared to Nyquist approach with time-delay ranges. The results are simulated and validated on a network of non-mobile agents, the simulation and hardware platforms are same for both the approaches. The simulation and implementation results demonstrate the effectiveness of the theoretical results obtained in both the approaches. Pros and cons of both the approaches are discussed in detail.

A heterogeneous multi-agent system is considered with different peak velocities for different agents. Moreover, the control law is involved with scaling factors that are different for different agents. It has been proved that, with undirected communication topology, the multi-agent system is able to reach consensus using the general nonlinear input. Simulation and hardware implementation are performed on the non-mobile multi-agent system to prove the effectiveness of theoretical results.

1.6 Thesis Organization

The thesis is organized into six chapters and the brief details of each chapter as given below.

- **Chapter 2**: In this chapter, preliminaries which are essential in understanding the subsequent chapters are described. Basic concepts of algebraic graph theory are explained. Few definitions are reviewed and associate matrix representation of graphs is discussed.

- **Chapter 3**: Two new algorithms are proposed in this chapter to tackle the complete loss of communication in a multi-agent system. The algorithms are tested in simulation environment and implemented on a hardware setup. Two of the robots in the hardware setup are developed in laboratory and the design procedure and working are detailed. The re-

sults of both simulation and implementation are provided to demonstrate the effectiveness of algorithms in dealing with complete loss of communication.

- **Chapter 4**: In this chapter, a multi-agent system with saturation nonlinearity is considered. Describing function analysis is used to rearrange the system into linear and nonlinear elements. Nyquist stability theory is used to prove the stability of linear element. Necessary and sufficient conditions for the consensus are derived with two control laws. The existence of limit cycles in the system is estimated using describing function of saturation nonlinearity and the stability of linear element. Simulation and implementation results are furnished to show the effectiveness of theoretical results. Hardware implementations are performed on a non-mobile setup, so that the limit cycles are observed properly.

- **Chapter 5**: Lyapunov approach is used in this chapter to prove the stability of linear element and further estimate the existence of limit cycles in this chapter. Four control laws are used to drive the multi-agent system towards consensus. Advantages and disadvantages of both Lyapunov approach and Nyquist approach are discussed in detail. Few comparative results are given and tested on hardware setup similar to the one used in Chapter 4. Analysis of consensus in the multi-agent system with a general nonlinear input is also performed. Lyapunov approach is used to prove the stability. Simulation and hardware implementation are performed to show the effectiveness of theoretical results.

- **Chapter 6**: Summary of results is presented in this chapter. Future scope and research directions in the multi-agent systems are discussed.

- **Appendix**: Basics of describing function analysis and discussion on how to estimate the existence of limit cycles in a nonlinear system are provided. Lyapunov stability theory and expansion of characteristic equation are explained briefly. It is concluded with technical specifications of the hardware setup used for performing the experiments.

CHAPTER 2

PRELIMINARIES

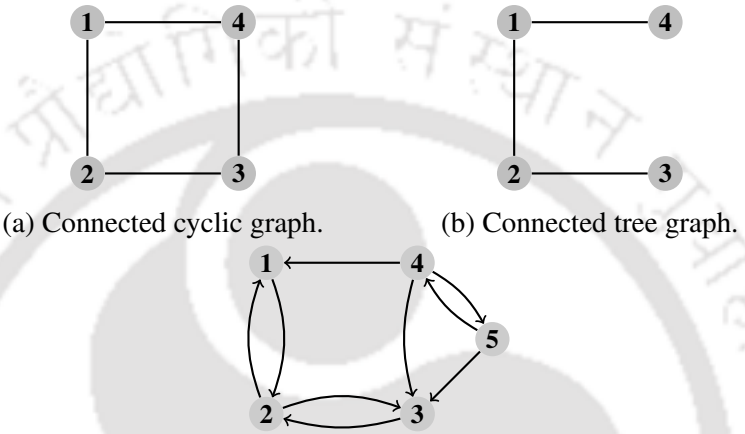
Basics of graph theory, notations and mathematical representation of multi-agent system are discussed in this chapter. The aim is to provide necessary concepts for clear understanding of the following chapters. We start with graph theory notation, proceed with adjacency matrix representation. Further discussion on obtaining a Laplacian matrix from the adjacency matrix is presented and some properties of the Laplacian matrix are discussed. Few definitions, terminology and notations with respect to graph are discussed. Additionally, describing function analysis and Lyapunov stability theory that are used in the subsequent chapters are reviewed. Readers who are familiar with aforementioned topics in the context of multi-agent systems can safely skip this chapter.

2.1 Algebraic Graph Theory

2.1.1 Basics

Graph theory is widely used to represent multi-agent systems. A network of n-agents connected by communication topology can be expressed as a graph $\mathscr{G} = (V, E, \mathscr{A})$. The vertex set $V = \{v_1, v_2, ..., v_n\}$ is generated by nodes of a graph, where nodes are analogous to agents. The branches of a graph bring out an edge set $E = \{(i, j) : i, j \in V\}$, that represents the underlying communication topology. Edge set has distinct ordered pairs of vertices which depict existence and direction of information flow among the vertices. If there is information flow from node j to node i, then the ordered pair $(i, j) \in E$ and vice versa.

If the communication among agents is bidirectional, the resulting graph is undirected as shown in Figs. 2.1a and 2.1b. A directed graph as depicted in Figs. 2.1c and 2.2 are used to represent communication topologies with one or more unidirectional information among agents. Edges $(i, j) \in E$ can have positive numerical weights w_{ij}. Unity weight is considered if no weight is given on edges. If two vertices v_i and v_j are connected by an edge $\big((i, j) \in E$ or $(j, i) \in E\big)$, then they are called *adjacent* vertices.



(a) Connected cyclic graph. (b) Connected tree graph.

(c) Directed graph.

Figure 2.1: Graphs of communication topologies.

Definition 1. *Path*: A path from vertex v_i to vertex v_j is a sequence of vertices connected by edges in an undirected graph, representing the information flow from v_i to v_j. For a directed graph, we have *directed paths* [103].

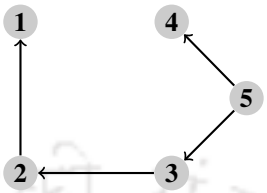
Definition 2. *Connected and strongly connected graphs*: An undirected graph is called *connected* if there exists a path between any pair of vertices $(\mathrm{v}_i, \mathrm{v}_j), \forall (i, j) \in [1, n]$ and $i \neq j$. A directed graph is called *strongly connected* if there exists a directed path between any pair of vertices $(\mathrm{v}_i, \mathrm{v}_j), \forall (i, j) \in [1, n]$ and $i \neq j$ [103].

Definition 3. *Tree and spanning tree*: A graph is said to be a *tree* if there exists only one path between any vertices without any loops. A tree is called a spanning tree, if there exists at least one *root* node that has a directed path to all the other nodes [103].

Cyclic graphs contain loops as shown in Figs. 2.1a, 2.1c and 2.2a and trees are acyclic by default as depicted in Figs. 2.1b and 2.2c. A graph is said to contain a spanning tree if a subset

(a) Strongly connected graph.

(b) Directed tree graph.



(c) Directed spanning tree graph.

Figure 2.2: Graphs of communication topologies.

of graph (subgraph) forms a spanning tree. Connected and strongly connected graphs have spanning tree as subgraph, Fig. 2.1b depicts a connected tree (also a spanning tree) which is a subset of graph in Fig. 2.1a. A graph which is not strongly connected can have a spanning tree as subgraph. The graph shown in Fig. 2.2c is a spanning tree with node-5 as its root node and it is a subset of the graph depicted in Fig. 2.1c.

2.1.2 Matrix representation of graphs

$\mathscr{A} = (a_{ij})_{n\times n}$ is an adjacency matrix of the graph with n nodes, it also represents communication topology with $a_{ij} = w_{ij}$ if $(i, j) \in E$ and $a_{ij} = 0$ otherwise. A weighted adjacency matrix will also have entries other than zero and unity weights, which depend on various assumptions. The adjacency matrices of graphs depicted in Figs. 2.1a and 2.1c with unity weights are as given below.

$$\mathscr{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \quad \mathscr{A}_2 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The number of nodes from which a node receives information is called in-degree (d_{in}) of the node. The number of nodes to which a node sends information is called out-degree (d_{out})

of the node. A diagonal matrix D is constructed with elements as given in Eqn. (2.1). Diagonal elements of D matrix (d_{ii}) are row sums of adjacency matrix and equivalent to in-degree of i^{th} node.

$$d_{ij} = \begin{cases} 0 & \forall i \neq j \\ \sum_{j=1}^{n} a_{ij}, & otherwise \end{cases} \tag{2.1}$$

The Laplacian L of an adjacency matrix $\mathscr{A}$ is defined in Eqn. (2.2).

$$L = D - \mathscr{A} \tag{2.2}$$

A Laplacian matrix has zero row sums and it is diagonally dominant ($|a_{ii}| \geq \sum_{i \neq j} |a_{ij}|$) with nonnegative diagonal elements. Let $\mathbf{1}_n$ and $\mathbf{0}_n$ be column vectors of ones and zeros respectively, then $L\mathbf{1}_n = \mathbf{0}_n$. For an undirected graph, both adjacency and Laplacian matrices are symmetric, $\mathscr{A} = \mathscr{A}^T$ and $L = L^T$. The in-degree and out-degree are equal at each node for an undirected graph. A directed graph is considered to be balanced if, in-degree of each node is same as out-degree of the node.

Lemma 1. [57] *If an undirected graph is connected, then the rank of Laplacian* $\rho(L_{n\times n}) = n-1$. *The right eigenvector associated with zero eigenvalue is* $\mathbf{1}_n$, $L\mathbf{1}_n = \mathbf{0}_n$ *and* $\mathbf{1}_n^T L = \mathbf{0}_n^T$.

Lemma 2. [57, 78]*If a directed graph* $\mathscr{G}$ *is strongly connected, then its Laplacian has one eigenvalue as zero (with right eigenvector* $\mathbf{1}_n$*) and rest of the eigenvalues have positive real parts. Converse is not true.*

Definition 4. [104] A matrix $A_{n\times n} \in \Re^{n\times n}$ is called *positive semidefinite* if $\mathbf{x}^T A\mathbf{x} \geq 0$, $\forall \mathbf{x} \in \Re^n$ and it is denoted by $A \succeq 0$. The matrix A is called *positive definite* if $\mathbf{x}^T A\mathbf{x} > 0$, $\forall \mathbf{x} \in \Re^n$ and it is denoted by $A \succ 0$.

$A_s = \frac{1}{2}(A + A^T)$ is called symmetric part of A. A necessary and sufficient condition for the matrix A to be positive semidefinite (or definite) is that the symmetric part should be positive semidefinite (or definite) [104, 105].

Property 1. A symmetric positive semidefinite matrix A_s has all non-negative eigenvalues ($0 \leq \lambda_1 \leq \lambda_2 \leq ...\lambda_n$). A symmetric positive definite matrix A_s has all positive eigenvalues ($0 < \lambda_1 \leq \lambda_2 \leq ...\lambda_n$).

Rank of a strongly connected graph Laplacian is $n-1$, zero is a simple (multiplicity 1) eigenvalue. If the graph is symmetric and connected, its Laplacian is symmetric and positive semidefinite with only one eigenvalue as zero, $(0 = \lambda_1 < \lambda_2 \leq ...\lambda_n)$

If each row of the adjacency matrix is normalized, we get a normalized adjacency matrix $\widetilde{\mathscr{A}}$, whose elements are given in Eqn. (2.3).

$$\widetilde{a}_{ij} = \begin{cases} \frac{a_{ij}}{d_{ii}} & if d_{ii} \neq 0 \\ 0 & if d_{ii} = 0 \end{cases} \tag{2.3}$$

The corresponding elements of normalized Laplacian matrix $\widetilde{L}$ is given in Eqn. (2.4).

$$\widetilde{l}_{ij} = \begin{cases} \frac{l_{ij}}{d_{ii}} & if d_{ii} \neq 0 \\ 0 & if d_{ii} = 0 \end{cases} \tag{2.4}$$

For a strongly connected network, the eigenvalues of normalized adjacency and normalized Laplacian matrices have the following relation given in Eqn. (2.5).

$$1 - \lambda(\widetilde{\mathscr{A}}) = \lambda(\widetilde{L}) \tag{2.5}$$

The Laplacian matrix, normalized adjacency matrix and normalized Laplacian matrix of graph shown in Fig. 2.1a are given below.

$$L_1 = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 2 & -1 & 0 \end{bmatrix}, \widetilde{\mathscr{A}_1} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}, \widetilde{L}_1 = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \end{bmatrix}$$

CHAPTER 3

CONSENSUS OF MULTI-AGENT SYSTEMS IN PRESENCE COMMUNICATION LOSS

3.1 Introduction

Communication loss is a very common phenomenon while implementing consensus algorithms on practical system such as a group of mobile robots. This loss of communication is attributable to various factors. Big obstacles like walls, robot away form antenna range of other robots etc., can cause communication loss. Multi-agent systems are connected by communication topologies, the changes in topologies are referred as switching topologies. Loss of communication and regain results in switching topologies. In some advanced applications, there can be delay in execution but failure is not desirable. Examples include a group of robots used for map making in a new environment, search operations by a robotic swarm in non-emergency situations etc.. Complete loss of communication is not studied in the existing literature [31, 106]. In this chapter, two algorithms, namely "back-tracking" and "history following", are proposed to reach consensus in case of communication loss with switching topologies. To reach consensus by a linear system in distributed control, a topology with spanning tree is sufficient [79]. However, in this work the proposed back-tracking algorithm always bring back the agents to a strongly connected topology and the reason for that is explained in Section 3.2.

The proposed back-tracking algorithm ensures that the agent backtracks its position unless the communication is re-established and a new path is taken to reach consensus. In history following, the agents use the values stored in memory and move towards consensus point calcu-

lated using those values until the communication is regained. Upon regaining communication, a new consensus point is calculated depending on the current positions of the agents and they change their trajectories accordingly. Simulation results, for a network of six agents, show that when the agents follow the previous history, the average consensus time is less than that of back-tracking. However, situation may arise in history following where a false notion of reaching consensus makes some of the agents stop at a point other than the actual consensus point. The agents fail to reach consensus in such a scenario.

An obstacle avoidance algorithm is integrated with the proposed algorithms in simulation to avoid collisions. Hardware implementation for a three robots system demonstrates the effectiveness of the algorithms. Presence of time-delays due to processing and communication may affect the stability of the agents. The proposed algorithms are able to withstand time-delays up to a certain limit. The time-delays considered in this chapter are primarily bounded communication time-delays. The attention in this chapter is focused on overcoming communication loss while consensus is being achieved. Simulation and implementation on a group of robots are performed to demonstrate the working of the proposed algorithms.

Organization of the chapter is as follows, Section 3.2 explains the system model mathematically and algorithms are defined in Sections 3.3 and 3.4. A sensor based obstacle avoidance algorithm is explained in Section 3.5, followed by simulation results in Section 3.6. A detailed description of hardware implementation is given in Section 3.7.

3.2 System Model

The position of mobile robot in a flat arena is represented by the position coordinates (x, y) with respect to the origin and the orientation (heading of robot) by ϕ with respect to X-axis. The linear velocity is represented by v and the angular velocity by ω. The linear acceleration u_1 and angular acceleration u_2 are the corresponding control inputs. The dynamics of each robot in a network is given by fifth order state space representation in Eqn. (3.1). The agent dynamics are chosen to represent both simulation and implementation models. A consensus

circle with center as the mean position of all agents is assumed to be a circular area, in which the robots once entered are expected to stay. The multi-agent system with switching topologies and time-delays is considered with an aim of robots gathering inside the consensus circle with radius *ccr*. The topologies with loss of communication (not strongly connected) and strongly connected topologies are considered for switching. Topology switching occurs in the case of communication loss and regain, which is a very common phenomenon in any communication network. Presence of big obstacles like wall may cause loss of link and some situations may arise when the link is never regained unless the position of the agent is changed. Loss of communication can result in failure of reaching consensus or flocking.

Two new algorithms called ”back-tracking” and ”history-following” are proposed to efficiently handle such situations when the communication topology is not strongly connected mainly because of the positions of the agents. According to the proposed back-tracking algorithm, an agent backtracks its path if it loses communication with other agents and choose a new path to reach the consensus. However, in the history-following case, the agents use their memory to follow the previous consensus point once the communication is lost. A new consensus point is calculated after the communication is re-established and the network is strongly connected again.

A group of n agents and the underlying communication topology is represented by a directed graph with n nodes and corresponding branches representing the communication topology. Branch weights of the directed graph are considered to be unity rather than non-uniform weights. The switching topologies based on position of agents are represented by corresponding directed graphs and adjacency matrices. Static obstacles are considered to be present in the paths of the agents which necessitates an obstacle avoidance algorithm to be incorporated with.

The i^{th} agent dynamics in a coordinate system positioned at (x_i, y_i) with an orientation of ϕ radians is given by Eqn. (3.1).

$$\begin{aligned}\dot{x}_i &= v_i \cos(\phi_i)\\ \dot{y}_i &= v_i \sin(\phi_i)\\ \dot{\phi}_i &= \omega_i\\ \dot{v}_i &= u_{i1}\\ \dot{\omega}_i &= u_{i2}\end{aligned} \tag{3.1}$$

Where $v_i \in \Re^n$ and $\omega_i \in \Re^n$ are linear velocity and angular velocity of the agent with dimension $n \times 1$. Control inputs $u_{i1}, u_{i2} \in \Re^n$ correspond to linear acceleration and angular acceleration. The control inputs u_{i1}, u_{i2} are designed using Eqn. (3.2).

$$\begin{aligned}u_{i1} &= f_1(u_{ix}, u_{iy}, u_{iv})\\ u_{i2} &= f_2(u_{ix}, u_{iy}, u_{iv})\end{aligned} \tag{3.2}$$

where, the feedback inputs, u_{ix}, u_{iy}, corresponding to the local interactions with respect to each agent, are given by Eqns. (3.3) and (3.4).

$$u_{ix} = \frac{1}{\sum_{j=1}^{n} a_{ij}} \left(\sum_{j=1}^{n} a_{ij} \Big(x_j\big((k-g_j)\,T \big) - x_i\big((k-g_i)\,T \big) \Big) \right) \tag{3.3}$$

$$u_{iy} = \frac{1}{\sum_{j=1}^{n} a_{ij}} \left(\sum_{j=1}^{n} a_{ij} \Big(y_j\big((k-g_j)\,T \big) - y_i\big((k-g_i)\,T \big) \Big) \right) \tag{3.4}$$

The feedback information about velocity of the agent is given by Eqn. (3.5).

$$u_{iv} = v_i\left((k-g_i)\,T\right) \tag{3.5}$$

For an undirected graph, $a_{ij} = a_{ji}$. It is assumed that position and velocity are constant in the interval $[kT, (k+1)\,T)$. The scalar k represent the step number, T represent the time duration of each step and kT represent the current time. Communication and input time-delays are represented by g_jT and g_iT respectively at current time kT. Since the focus is on overcoming communication loss, the time-delays are chosen to be single step (T). The single step time-delay will not affect the performance of system when the network topology is strongly connected.

Time-delays are incorporated in the systems by using the past values of position instead of the present position. Linear control laws in Eqns. (3.3) and (3.4) result in large acceleration and velocity when the agents are far from each other. The linear control laws cannot be used directly in simulation since obstacle avoidance is designed for low velocity scenario and also due to limitations in hardware. Robots are nonlinear in nature, they cannot handle large values of velocity and acceleration due to actuator saturation. Control laws in Eqn. (3.2) are similar to saturation function with constraints on acceleration and velocity. Most of the consensus control laws in literature are linear and switching topologies are periodic in nature. To emulate the situation where topology changes due to specific positions of the agents, algorithms are used to stop the robots from communicating with other robots when they enter into those specific positions.

The robot motion control is performed through the control of linear and angular accelerations with respect to origin. The algorithms follow a nonlinear control when they are strongly connected. The robots are equipped with multi-hop communication and enables each agent to get information of all the agents when strongly connected. Consequently, the control law defined in Eqns. (3.2), (3.3), (3.5) and (3.7) coupled with algorithms is a special case of the one described in Eqns. (5.70) and (5.71). In section 3.2, each robot is described by fifth order dynamics. When observed carefully, each robot has two subsystems, one for position and other for orientation with respect to origin. With $r_i = \sqrt{x_i^2 + y_i^2}$, it can be rewritten as,

$$\begin{aligned} \dot{r}_i &= v_i \\ \dot{\phi}_i &= \omega_i \\ \dot{v}_i &= u_{i1} \\ \dot{\omega}_i &= u_{i2} \end{aligned} \tag{3.6}$$

When compared with the system defined in section 5.4, the system in Eqn. (3.6) has double the number of states with similar relation among the states. The stability analysis of the above system follows a similar procedure as described in section 5.4.

The control inputs in Eqn. (3.2), $f_1(u_{ix}, u_{iy}, u_{iv})$ and $f_2(u_{ix}, u_{iy}, u_{iv})$ are computed by the

following algorithmic steps.

1. If $step = 1$, the agents broadcast their positions without moving. If $1 < step \leq step_{min}$, the agents broadcast messages to incorporate user defined time-delay in the system.

2. The heading of each agent is updated by the control law $f_2(u_{ix}, u_{iy}, u_{iv})$ based on feedback from other agents when the agent is not inside the consensus circle radius ccr ($\sqrt{{u_{ix}}^2 + {u_{iy}}^2} > ccr$) with center as mean position of other agents.

3. $f_1(u_{ix}, u_{iy}, u_{iv}) = \pm a$ if $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} > ccr$ and $u_{iv} < v_{max}$ i.e., if an agent is not inside the consensus circle radius ccr, set acceleration as $\pm a$ until velocity of the agent reaches v_{max}.

4. $f_2(u_{ix}, u_{iy}, u_{iv})$ continuously updates the heading using obstacle avoidance algorithm in section 3.5.

5. Repeat above algorithmic steps until $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} \geq ccr$.

6. When an agent loses communication with other agents, the agent is brought to halt by setting deceleration d until velocity becomes zero, i.e., $f_1(u_{ix}, u_{iy}, u_{iv}) = -d$ if $v_i > 0$.

7. When $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} < ccr$, $f_1(u_{ix}, u_{iy}, u_{iv}) = -d$ if $v_i > 0$ and $f_1(u_{ix}, u_{iy}, u_{iv}) = 0$ if $v_i = 0$ i.e., the agent is brought to halt by decelerating until velocity becomes zero.

8. The simulation is continued till $step_{max}$ number of steps.

The parameters $step_{min}$, $step_{max}$, ccr, a, d and v_{max} are to be chosen judiciously by the users based on the requirement and limitations of hardware. The parameter $step_{min}$ is used in the cases where time-delay effects are tested. The above position based switching algorithm without back-tracking or history following is elaborated in a flow chart as shown in Fig. 3.1.

Control inputs u_{i1}, u_{i2} in Eqn. (3.2) are able to solve consensus problem of linear multi-agent system if the graph has a spanning tree [74] [77]. A graph becomes strongly connected when every node has a directed path to all the other nodes. Likewise, the graph has a spanning tree when all nodes have a directed path from at least one node. In distributed control, the

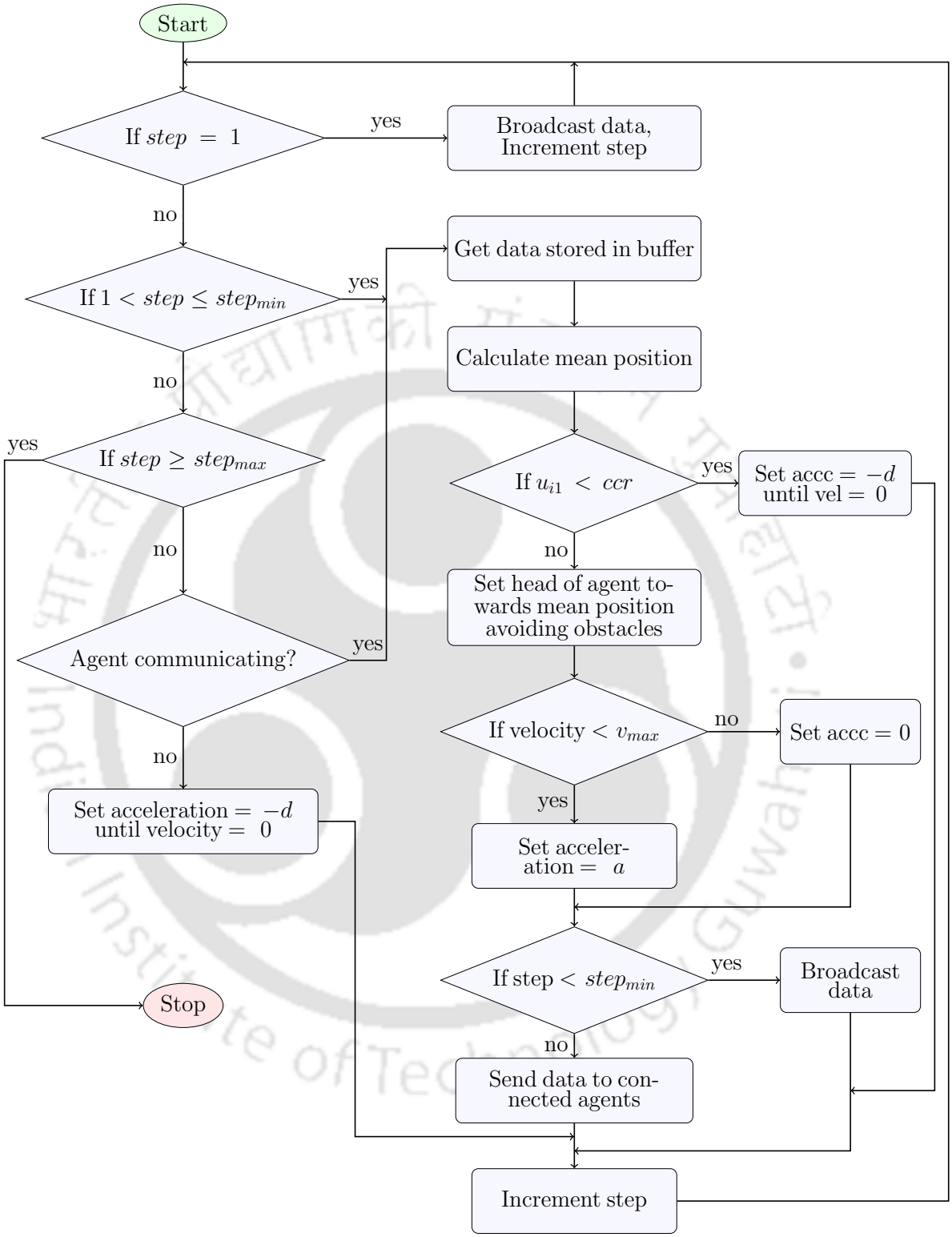


Figure 3.1: Flow chart for position based switching without back-tracking or history following

network need not be strongly connected to reach consensus. Consensus of a linear system can be ensured if the graph has a spanning tree for fixed topology or the union of all the topologies has a spanning tree for a periodic switching topologies [79]. However, in a spanning tree topology, non-root agents may not have information about other agents. This may lead to the agents not being able to decide on coming to halt or use back-tracking algorithm when the topology does not have a spanning tree. Moreover, the agents and the corresponding control laws are nonlinear. To overcome the issue, a strongly connected topology has been considered as a necessary condition. A multi-agent system with nonlinear control laws reach consensus if it has connected network topology [68].

If $x_i \to x_j, y_i \to y_j, \forall i \neq j$ and $\dot{x_i} \to 0, \dot{y_i} \to 0, \dot{\phi_i} \to 0$, $\forall i \in [1,n], \forall j \in [1,n]$, then consensus is said to be reached in position for n point masses. It can also be said that, consensus in position of n robots is reached if, $x_i \to {x_i}^*$, $y_i \to {y_i}^*$, $\dot{x_i} \to 0, \dot{y_i} \to 0$ and $\omega_i \to 0$, where substituting $({x_i}^*, {y_i}^*)$ in the place of (x_i, y_i) satisfy the condition $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} < ccr$, $\forall i \in [1,n]$.

3.3 Back-tracking Algorithm

In position based switching, incorporation of back-tracking allows an agent to back-track some predefined amount of distance when the communication topology is not strongly connected. The back-tracking occurs unless the agent reaches the position where communication is re-gained. At this point the agent takes another path around communication loss point, avoiding some area defined in the algorithm. Back-tracking, path change and the subsequent course correction are performed according to the algorithm. In a distributed control, since all the agents take decisions independently, every agent should have information about other agents to take decision on when the agent should back-track. Therefore, it is necessary for the network to be strongly connected. If an agent is not able to receive data from any of the other agents, it halts and backtracks the path to re-establish the communication link. Then the agent reroutes itself based on algorithmic steps, preserving connectedness of the network. This process is repeated until consensus is reached. The algorithm heuristically brings the agents into connected con-

figuration. If the agents are connected, they reach the consensus with the control inputs. The mathematical analysis of the same is provied in Section 5.4

The algorithmic steps are as given below.

1. If $step = 1$, the agents broadcast their positions without moving. If $1 < step \leq step_{min}$, the agents broadcast messages to incorporate time-delay and back-tracking in the system.

2. The heading of each agent is updated by the control law $f_2(u_{ix}, u_{iy}, u_{iv})$ based on feedback from other agents when the agent is not inside consensus circle radius ccr ($\sqrt{{u_{ix}}^2 + {u_{iy}}^2} > ccr$) with center as average position of other agents.

3. $f_1(u_{ix}, u_{iy}, u_{iv}) = \pm a$ if $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} > ccr$ and $u_{iv} < v_{max}$ i.e., if an agent is not inside the consensus circle radius ccr, set acceleration as $\pm a$ until velocity of the agent reaches v_{max}.

4. $f_2(u_{ix}, u_{iy}, u_{iv})$ continuously updates the heading using obstacle avoidance algorithm in section 3.5.

5. If an agent is not receiving data from other agents, then $f_1(u_{ix}, u_{iy}, u_{iv}) = -d$ if $u_{iv} > 0$ and $f_1(u_{ix}, u_{iy}, u_{iv}) = 0$ if $u_{iv} = 0$ are used to bring the agent to halt.

6. In the next step, make $f_1(u_{ix}, u_{iy}, u_{iv}) = -a$ to back-track and it is repeated for the next b_1 number of steps. Then use $f_1(u_{ix}, u_{iy}, u_{iv}) = d$ if $u_{iv} < 0$ and $f_1(u_{ix}, u_{iy}, u_{iv}) = 0$ if $u_{iv} = 0$ to bring the agent comes to halt.

7. Change the heading of agent by ϕ_b degrees using $f_2(u_{ix}, u_{iy}, u_{iv})$ and make $f_1(u_{ix}, u_{iy}, u_{iv}) = a$ if $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} > ccr$ and $u_{iv} < v_{max}$ for next b_2 number of steps.

8. Change heading towards mean position with the help of $f_2(u_{ix}, u_{iy}, u_{iv})$ and proceed further.

9. Iterate above steps until $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} > ccr$

10. When $\sqrt{{u_{ix}}^2 + {u_{iy}}^2} \leq ccr$, the agent is brought to halt by making $f_1(u_{ix}, u_{iy}, u_{iv}) = -d$ until $u_{iv} > 0$ and $f_1(u_{ix}, u_{iy}, u_{iv}) = 0$ if $u_{iv} = 0$.

11. The simulation is continued till $step_{max}$ number of steps.

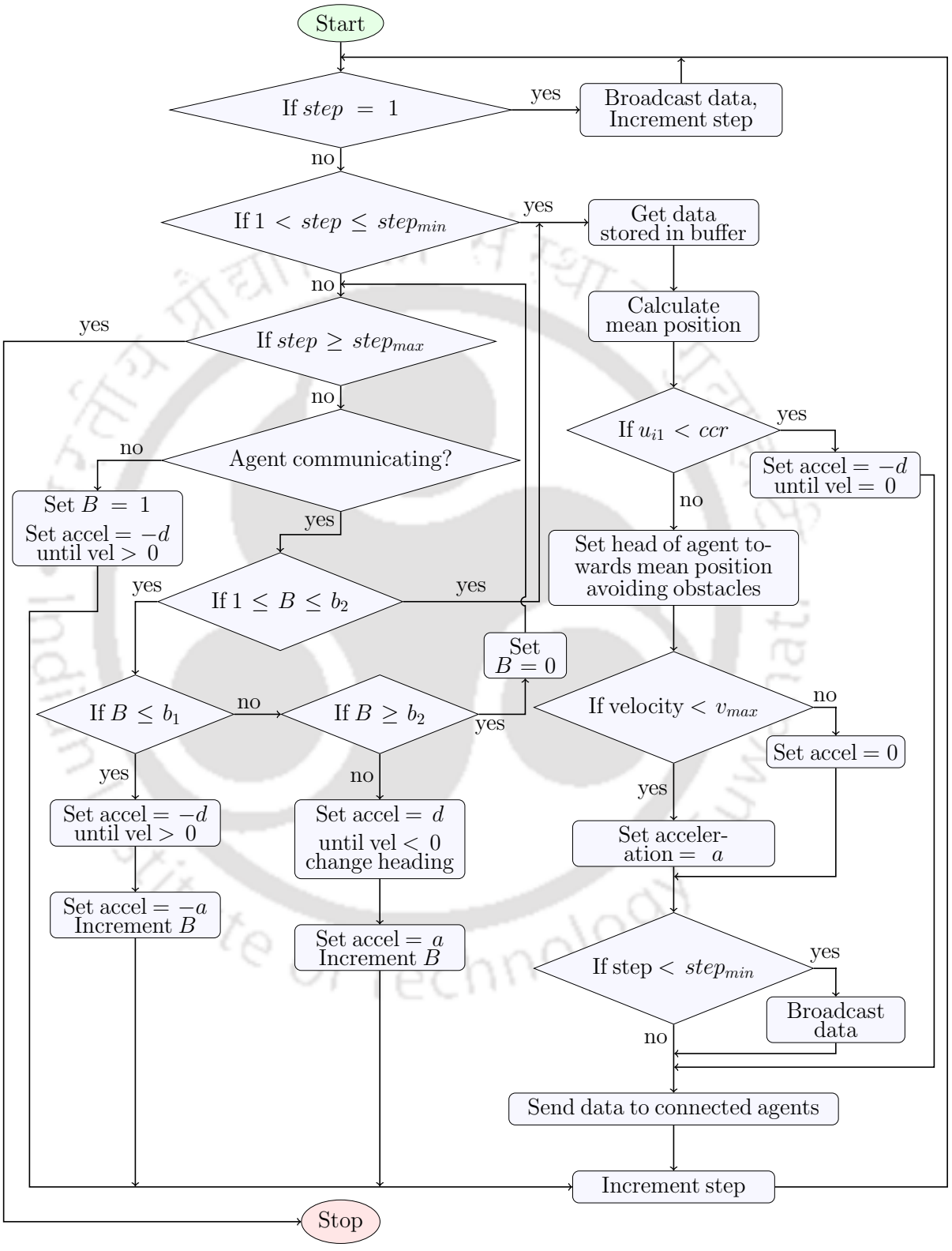


Figure 3.2: Flow chart for position based switching with back-tracking

Parameters $step_{min}$, $step_{max}$, ccr, a, v_{max}, b_1, b_2 and ϕ_b are to be judicially chosen for specific applications. The above algorithm is represented using a flow chart in Fig. 3.2.

3.4 History Following with Memory Enabled Agents

Here, position based switching is considered where agents can retain the history of average consensus point in the memory. Whenever an agent loses communication with other agents, it proceeds with data stored in memory till it regains communication. The algorithmic steps are very similar to the one without back-tracking except that an agent does not come to halt when it loses communication. The algorithm can be depicted in a flow chart as given in Fig. 3.3. Next section presents the obstacle avoidance algorithm to be incorporated with the proposed algorithms.

3.5 Sensor Based Obstacle Avoidance

A laser sensor that is available in MRSim application for Matlab is used in simulations for the purpose of obstacle avoidance. Beam-width of the sensor is defined to be 180° and a range of 300 *pixels* is assigned. The sensor returns an array of distance values with which, an algorithm enumerated below is implemented. Fig. 3.4 gives a pictorial representation of the robot shape and area of obstacle avoidance which is divided in a number of sectors. A detailed algorithm is provided below, that is used for agents of maximum size 5 *pixels*. For larger robots, the numerical values should be chosen appropriately.

1. Scan the 180° sector in the front using laser scanner to calculate cone of avoidance.
2. If an obstacle is found within 150*pixels* in the sector of ±2° of present heading, then turn the agent to nearest 4° sector with no obstacles.
3. If an obstacle is found within $5/\sin(\theta)$ distance in the 57°sectors of ±3° to ±60° of present heading, then turn the agent to avoid obstacle with minimum of 1° change and maximum 30° change.

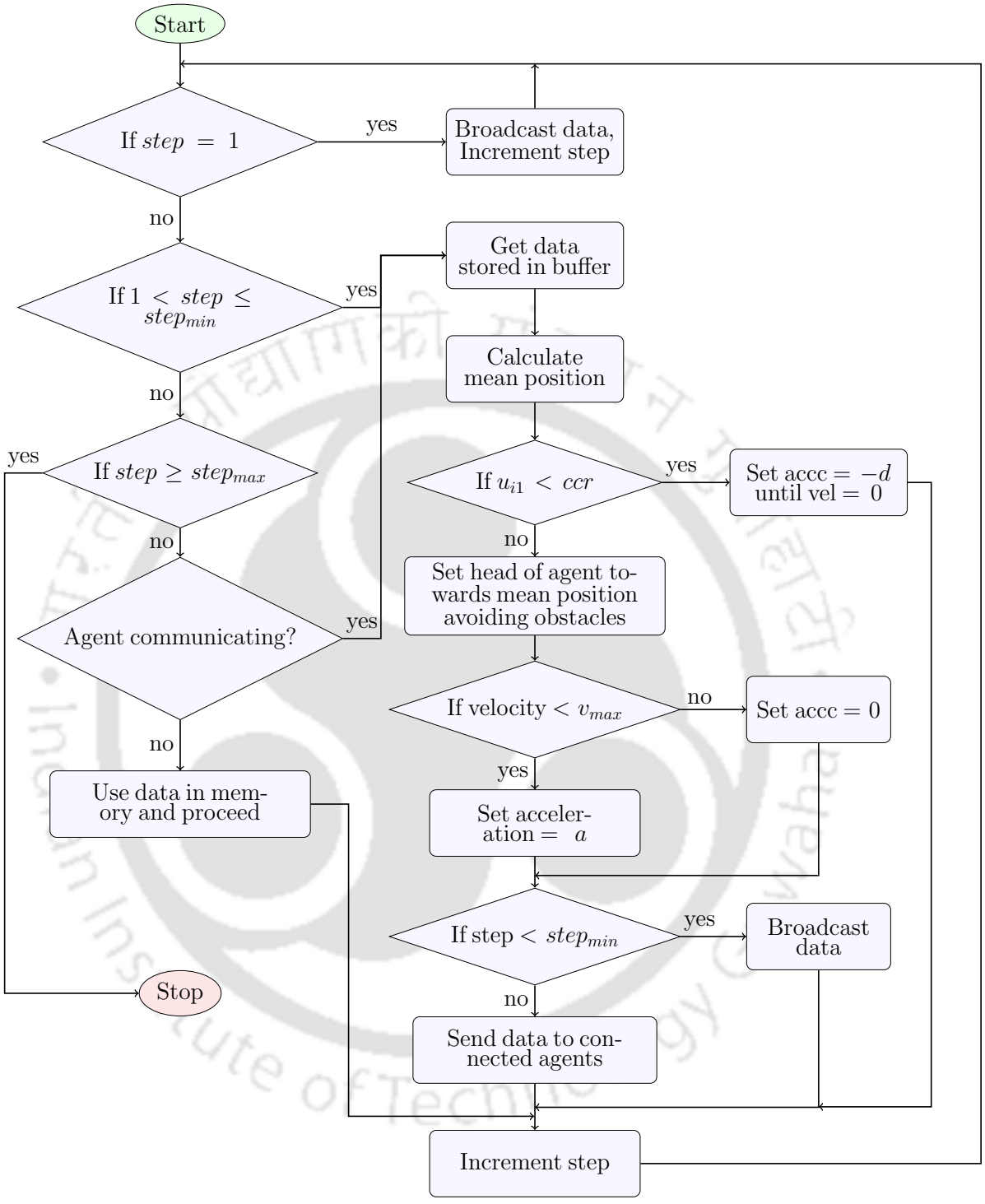


Figure 3.3: Flow chart for position based switching with memory enabled agents

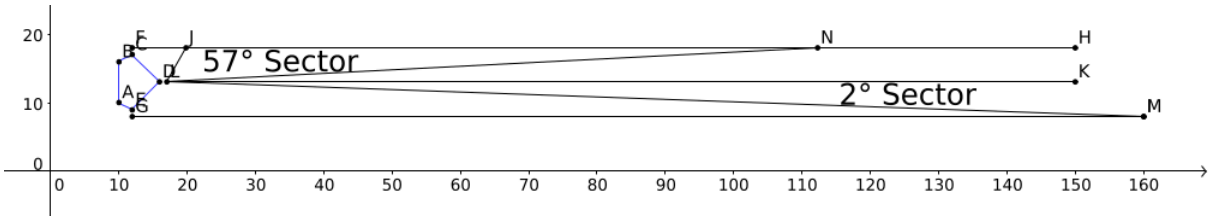


Figure 3.4: Pictorial Representation of Obstacle Avoidance

4. If an obstacle is found within $8 pixels$ in the 30 sectors of ± 61 to ± 90, then turn the agent by 30°.

5. If an obstacle is found within $5 pixels$ in the sector of $\pm 90^{o}$ of present heading, bring the agent to halt position.

3.6 Simulation Results

This section presents the simulation of multi-agent consensus for various non-switching and switching topologies. The simulations are done using MRSim - Multi-Robot Simulator(V1.0) compatible on MATLAB 7.12 (R2011a)+. The simulator allows a maximum of 255 robots to run at a time. The robots are allowed to have user-defined shape. The simulation environment is defined by a 2-dimensional 1-bit(black/white) bitmap image. Borders and obstacles are defined by zero valued pixels and rest of the pixels has value one. The obstacles in simulation results are depicted using darker shade gray boxes and communication loss areas with lighter shade gray boxes.

3.6.1 Arena and Initial Conditions

The arena is chosen to be of size $1412 \times 773 pixels$ whose origin is considered to be at south-west position. The arena has borders at $26 pixels$ in the West, at $35 pixels$ in the South, at $1384 pixels$ in the East and at $747 pixels$ in the North. Each agent is fit in a circle with $5 pixels$ radius.

Six agents are considered for simulation, so the consensus circle radius ccr is considered as $30 pixels$ to enable them to fit in the region even if they stand in straight line. The maximum

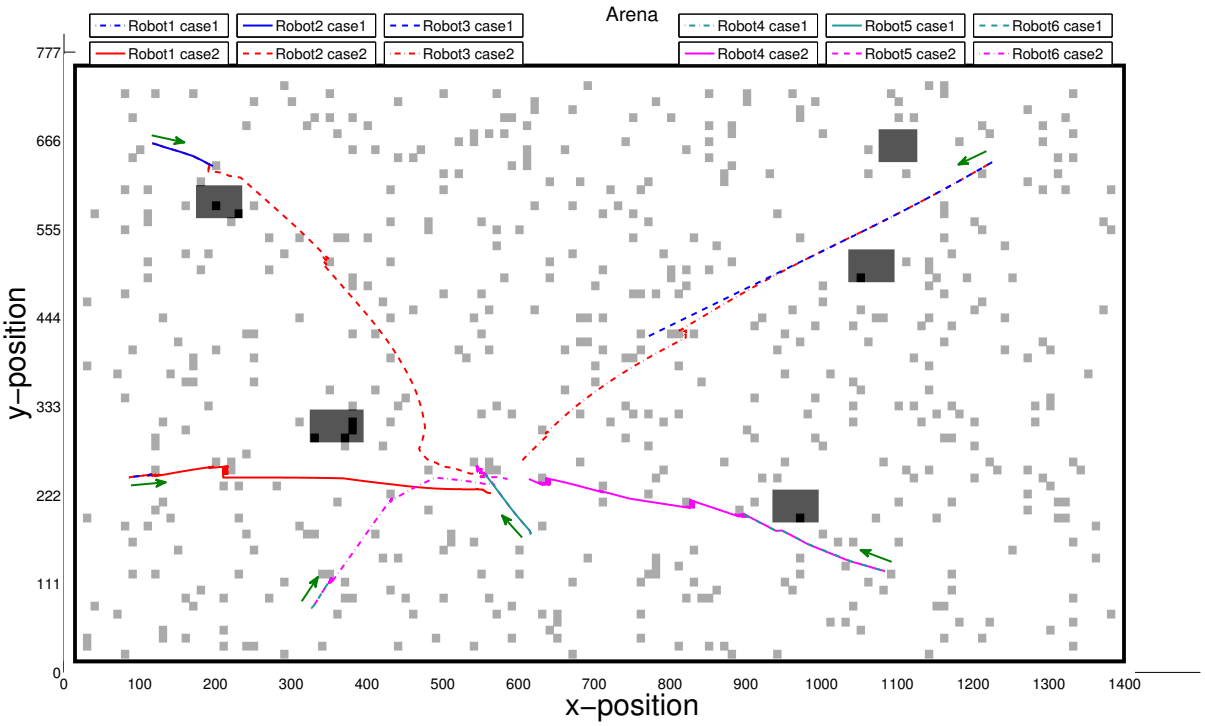


Figure 3.5: Consensus of switching topologies based on position without back-tracking algorithm(case-1) and with back-tracking algorithm(case-2): loss of communication occurs in 5% of the total area.

velocity of each agent is considered as $v_{max} = 2 pixels/step$, acceleration as $a = 0.1 pixels/step^2$ and deceleration as $d = 0.5 pixels/step^2$. It is considered that for the first $step_{min} = 10$ steps, all the agents broadcast data to enable back-tracking without failure. $step_{max}$ is considered as 500 for non-switching cases and 800 for position based switching cases. Number of steps for back-tracking is considered as $b_1 = 5$, number of steps for path change is considered as $b_2 = 10$ and change in the angle after back-tracking is considered to be $\phi_b = 45^\circ$.

3.6.2 Position Based Switching Topologies

In practical cases, the position of robot decides the communication topology switching. For the sake of simplicity, two communication topologies are assumed and the corresponding adjacency matrices are as given below.

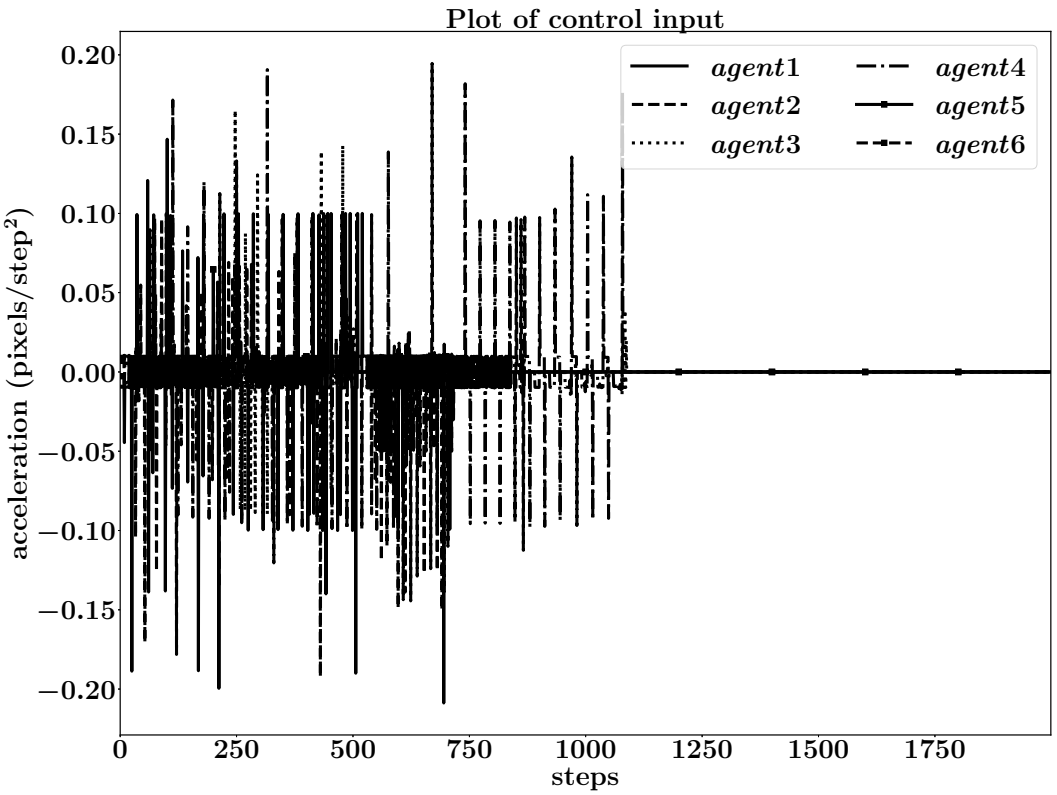


Figure 3.6: Plot of control inputs with back-tracking algorithm(case-2): loss of communication occurs in 5% of the total area.

$$A1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}, A2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The arena has five percent of area in which the agents lose communication and it is denoted with lighter shade gray boxes in the Fig. 3.5. Simulation is done using algorithm in Fig. 3.1, the agents come to halt when they lose communication, which is shown in Fig. 3.5 labelled as case-1. The direction robot motion is shown using a green arrow in all the simulation images. Initial positions of robots for result depicted in Fig. 3.5 are as given below,

Robot1: $(86, 244)$, Robot2: $(117, 663)$, Robot3: $(1225, 639)$, Robot4: $(1084, 127)$, Robot5: $(616, 173)$, Robot6: $(326, 80)$.

3.6.3 Position Based Switching Topologies with Back-tracking

Here, it has been considered that the switching of communication topology is based on position and agents have the capability of using back-tracking algorithm. The agent stops whenever it is not receiving data from any of the other agents, i.e., when the topology is not strongly connected. It then backtracks for b_1 number of steps to gain communication and reroutes itself to ensure a strongly connected topology. The simulations are performed using the communication topologies whose adjacency matrices are provided earlier.

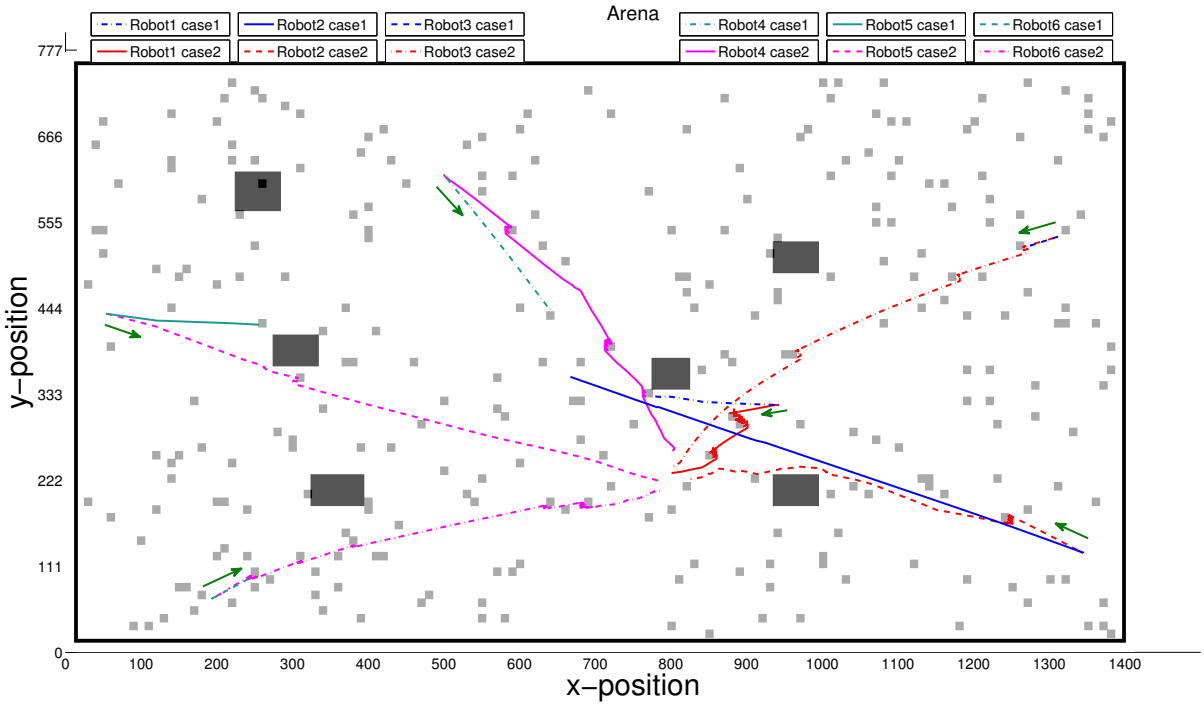


Figure 3.7: Consensus of switching topologies based on position without back-tracking [1](case-1) and with back-tracking algorithm(case-2): loss of communication occurs in 3% of the total area.

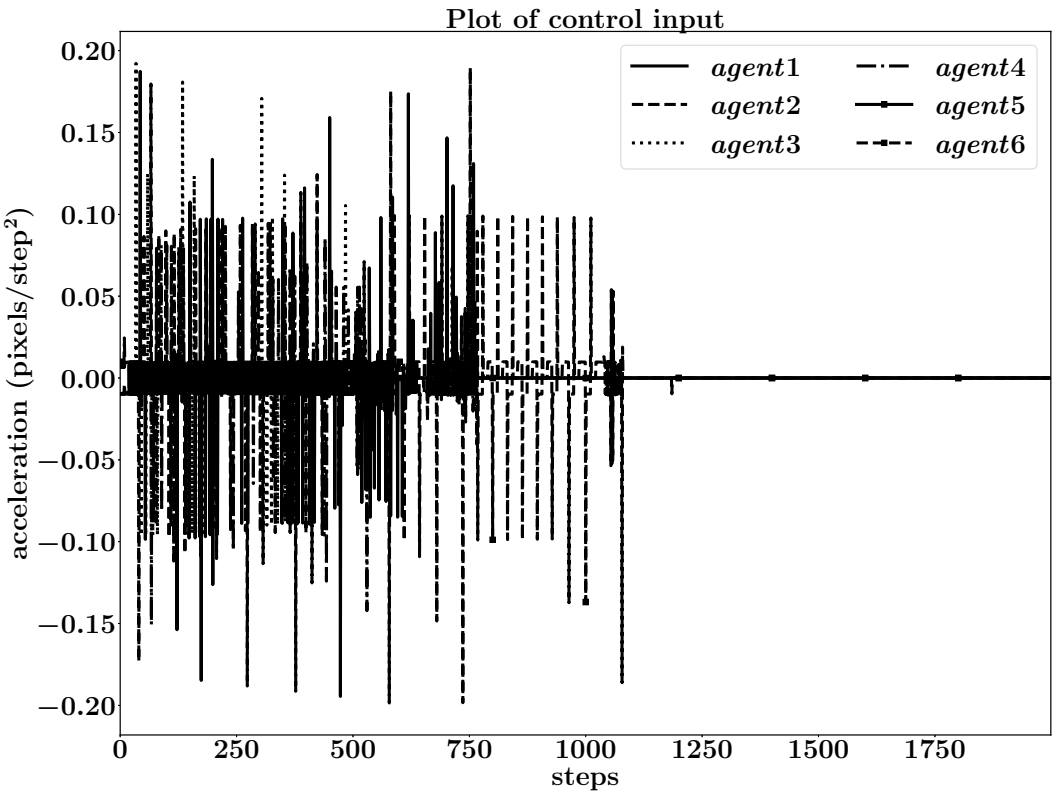


Figure 3.8: Plot of control inputs with back-tracking algorithm(case-2): loss of communication occurs in 3% of the total area.

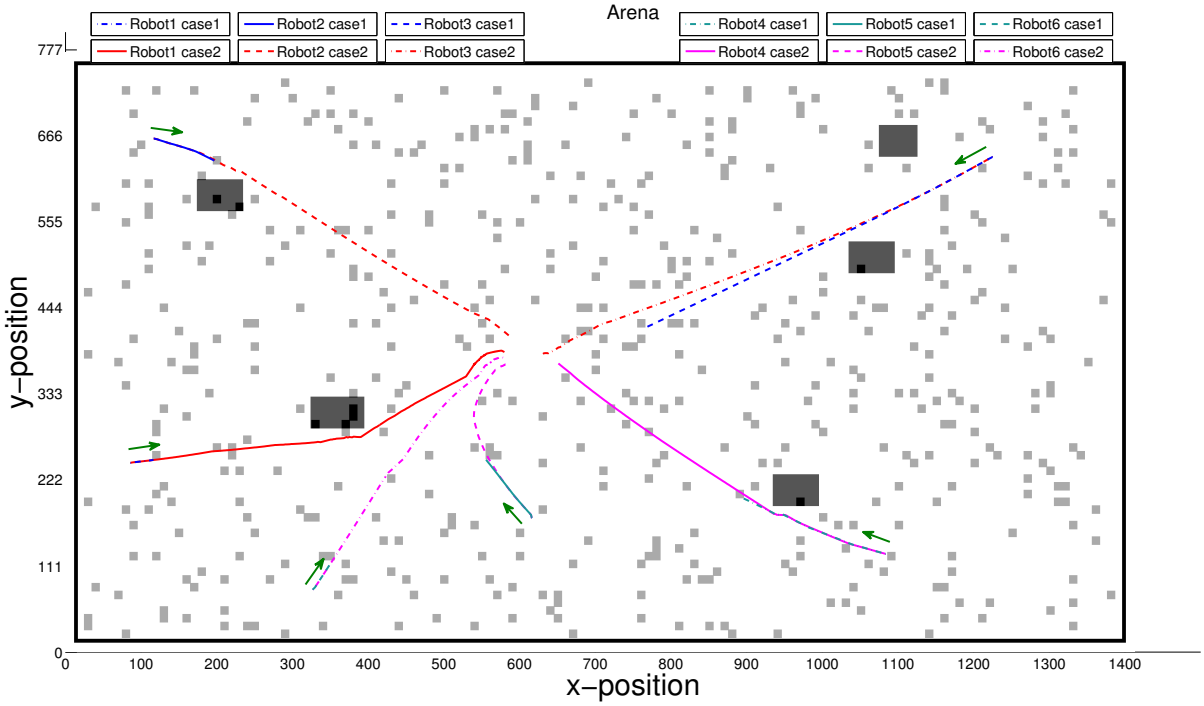


Figure 3.9: Consensus of switching topologies based on position without memory(case-1) [2] and with memory enabled agents(case-2): loss of communication occurs in 5% of the total area

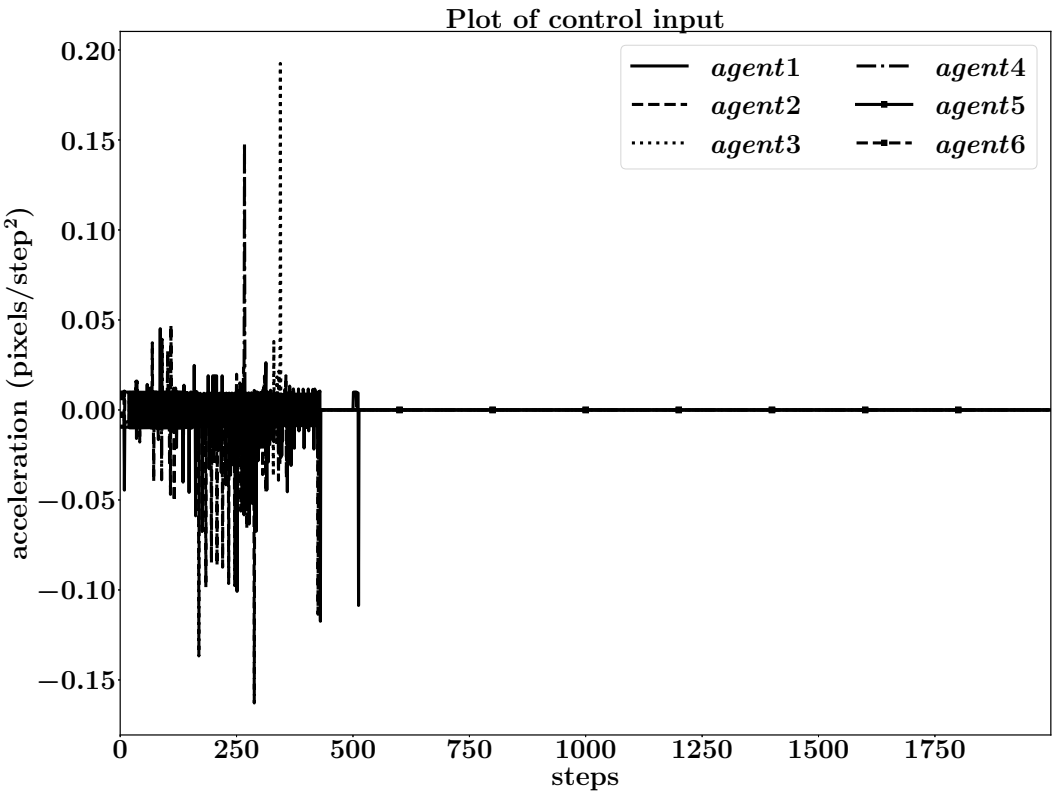


Figure 3.10: Plot of control inputs with memory enabled agents(case-2): loss of communication occurs in 5% of the total area.

Simulation results depict that whenever an agent is not receiving data, it reroutes using back-tracking algorithm. Due to usage of distributed control and back-tracking, strongly connected topology is maintained instead of graphs with spanning tree subgraph. From the simulation results, it is observed that agents reroute on need basis to ensure all the agents receive and send data to other agents. Simulations are done for two different scenarios as shown in Figs. 3.5 to 3.8. The first scenario in Fig. 3.5 has five percent of area in which the agents lose communication, the second scenario in Fig. 3.7 has three percent of area in which the agents lose communication and it is denoted with lighter shade gray boxes. Figs. 3.5 and 3.7 show the results where agents lose communication at various points. Agents try to reach consensus without using back-tracking algorithm is labeled as case-1 and with using back-tracking algorithm is labeled as case-2. It can be observed that, the agents finally reach consensus using back-tracking even with communication loss at multiple instances. Initial positions of robots for result depicted in Fig. 3.7 are as given below,

Robot1: $(943, 319)$, Robot2: $(1345, 128)$, Robot3: $(1312, 536)$, Robot4: $(500, 616)$, Robot5:

$(54,437)$, Robot6: $(194,69)$

3.6.4 Position Based Switching Network Topology with Memory Enabled Agents

In this case, the simulations are performed with the two adjacency matrices as given earlier. Simulations are done for two different scenarios similar to earlier case as shown in Figs. 3.9 and 3.10 case-2 and Figs. 3.11 and 3.13 case-1. From Fig. 3.9, it can be observed that the agents reach consensus but from Fig. 3.11 it can be observed that robot1 is not reaching consensus. The reason for this behavior is, robot1 comes to halt based on data in its memory after losing communication and rest of the agents change path to avoid collision, resulting in new mean position that is not available to robot1. This is a drawback which arises if an agent can not gain communication before rest of the agents reach consensus and come to halt. However, it can also be observed from Fig. 3.11 case-2, the agent reach consensus using back-tracking algorithm for similar situation. Initial positions of robots for result depicted in Fig. 3.9 are as given below, Robot1: $(86,244)$, Robot2: $(117,663)$, Robot3: $(1225,639)$, Robot4: $(1084,127)$, Robot5: $(616,173)$, Robot6: $(326,80)$.

The control inputs are depicted in Figs. 3.6, 3.8, 3.10 and 3.13. It can be observed that in all the cases, control input reaches zero after reaching the final destination. This implies that the aim of all the agents, *i. e.*, the consensus has achieved.

Initial positions of robots for result depicted in Fig. 3.11 are as given below, Robot1: $(943,319)$, Robot2: $(1345,128)$, Robot3: $(1312,536)$, Robot4: $(500,616)$, Robot5: $(54,437)$, Robot6: $(194,69)$.

3.6.5 Comparison of Results

In the comparison shown in Table 3.1, different cases have the following parameters

Case 1: Position based switching without back tracking using algorithm in Fig. 3.1

Case 2: Position based switching without back tracking and memory using control laws in [1] [2]

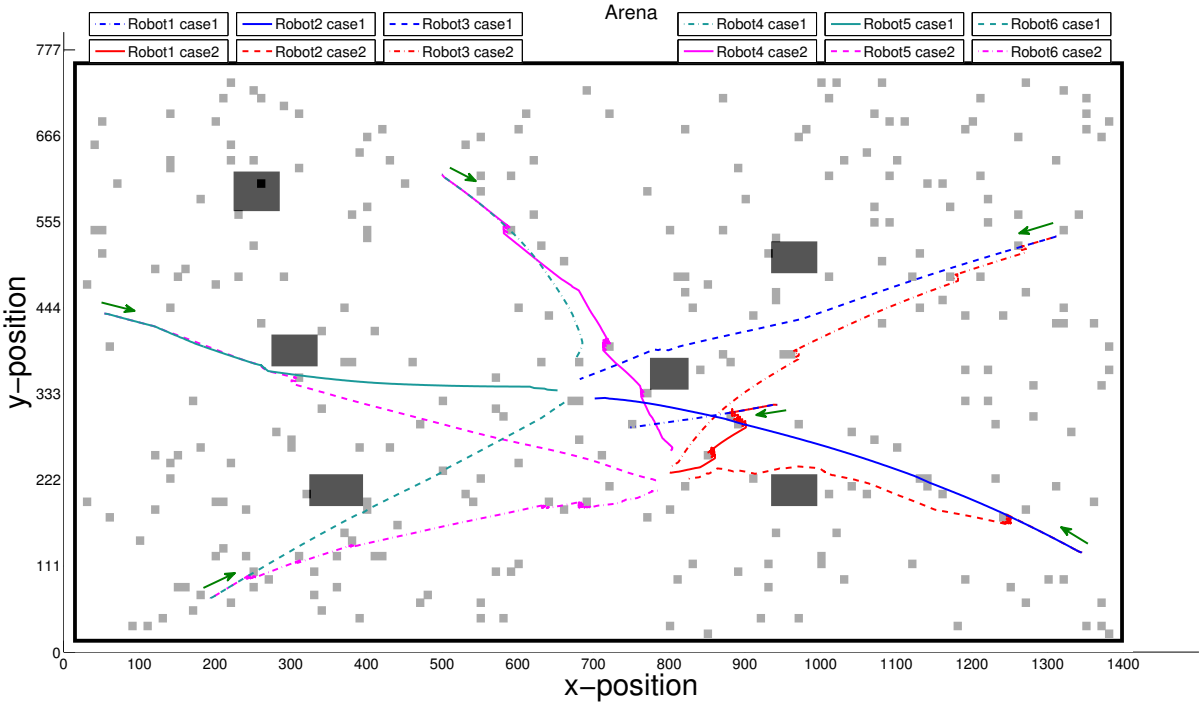


Figure 3.11: Consensus of switching topologies based on position with memory enabled agents(case-1) and with back-tracking(case-2): loss of communication occurs in 3% of the total area

Table. 3.1: A comparison on the time to reach consensus for various switching topologies

Switching topology	**Time to reach consensus**		
	First run	*Second run*	*Third run*
Case 1	-	-	-
Case 2	-	-	-
Case 3	187.6 sec and 248.4 sec for 3% and 5% loss of communication respectively	96.6 sec and 98.2 sec for 3% and 5% loss of communication respectively	60.4 sec and 73.2 sec for 3% and 5% loss of communication respectively
Case 4	133 sec for 3% and 5% loss of communication	68.4 sec for 3% and 5% loss of communication	41.8 sec for 3% and 5% loss of communication

Case 3: Position based switching with back tracking using algorithm in Fig. 3.2

Case 4: Position based switching with history following using algorithm in Fig. 3.3

First run: $a = 0.05$, $d = 0.25$, $V_{min} = -1$, $V_{max} = 1$, $ccr = 30$, $step_{min} = 20$, $b_1 = 20$, $b_2 = 20$ and arena as shown in Fig. 3.9

Second run: $a = 0.1$, $d = 0.5$, $V_{min} = -2$, $V_{max} = 2$, $ccr = 50$, $step_{min} = 10$, $b_1 = 10$, $b_2 = 20$ and arena is as shown in Fig. 3.11

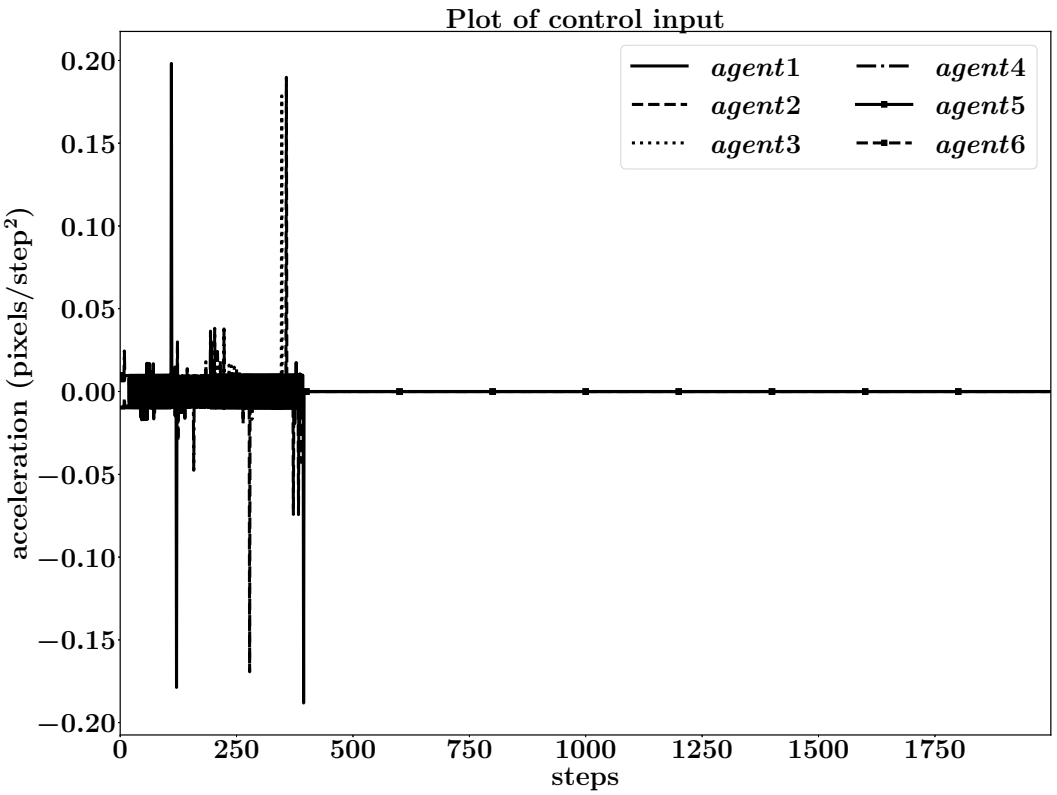


Figure 3.12: Plot of control inputs with with memory enabled agents(case-1): loss of communication occurs in 3% of the total area.

Third run: $a = 0.3$, $d = 1$, $V_{min} = -3$, $V_{max} = 3$, $ccr = 40$, $step_{min} = 5$, $b_1 = 5$, $b_2 = 15$ and arena as shown in Fig. 3.9

From the simulation results shown in Table 3.1, it can be observed that increase in acceleration and velocity with back tracking will not directly reduce the convergence time. The convergence time in a given arena is affected by initial conditions for different arenas and number of times back tracking take place for the last robot while reaching consensus. The number of times back tracked by last robot is different for non-identical parameters. *First run:* 8 and 13 for 3% and 5% loss of communication respectively. *Second run:* 5 and 9 for 3% and 5% loss of communication respectively. *Third run:* 6 and 7 for 3% and 5% loss of communication respectively. Values of $step_{min}$, b_1 and b_2 will affect the path of robot which will further affect convergence time. All the parameters should be chosen judiciously depending on size of arena, percentage of no communication zones and size of robots. Proposed algorithms are robust to work for different sets of parameters and arena conditions. It is observed that the convergence

time is increased when the percentage of area with no communication is increased from three percent to five percent. In the case of history following algorithm, agents are required to store the most recent update and the extra memory required is negligible. The average consensus reaching time is less in case of memory enabled agents. From different runs, it is evident that the changes in acceleration and velocity also reflect in convergence time. However, as shown in Fig. 3.11, in some situations, following previous history may result in some error in the final destination. An algorithm should be chosen by considering trade-offs between convergence time and guaranteed convergence.

Control laws from [1,2,92] have been considered for comparison and the result of simulation is similar as the case without back-tracking. The authors did not consider the complete loss of communication based on position of robots, the robots did not reach consensus once they lose communication in the arena. The simulation result for the control law proposed by Zhou. *et al.* [1] is shown in Fig. 3.7, which is labeled as case 1. It can be observed that the robots stop when they lose communication from other robots.

Mean square errors of the final position with respect to actual goal are calculated and given in Table 3.2. The values are calculated for the results shown in Figs. 3.5, 3.7, 3.9 and 3.11. Fig. 3.1 represents the algorithm without back-tracking and without history following. Fig. 3.2 represents the algorithm with back-tracking and Fig. 3.2 represents the algorithm with history following. As discussed earlier, it can observed that the mean square error for the algorithm with back-tracking is lowest for both 5% and 3% loss of communication.

Table. 3.2: Mean square errors of the final position

Loss of communication (%)	Algorithm reference	Mean square error
5%	Fig. 3.1	55878.61
5%	Fig. 3.2	485.05
5%	Fig. 3.3	507.19
3%	Fig. 3.1	115567.29
3%	Fig. 3.2	196.23
3%	Fig. 3.3	521.33

3.7 Implementation on Hardware

Hardware implementation is done on a network of three agents of which one is a Research PatrolBot manufactured by Adept MobileRobots and other two robots are developed in laboratory. Patrolbot is controlled using laptop and rest of the robots are controlled by BeagleBone-Black. The algorithm implemented on the system in Fig. 3.13 is very similar to simulation except the obstacle avoidance part.

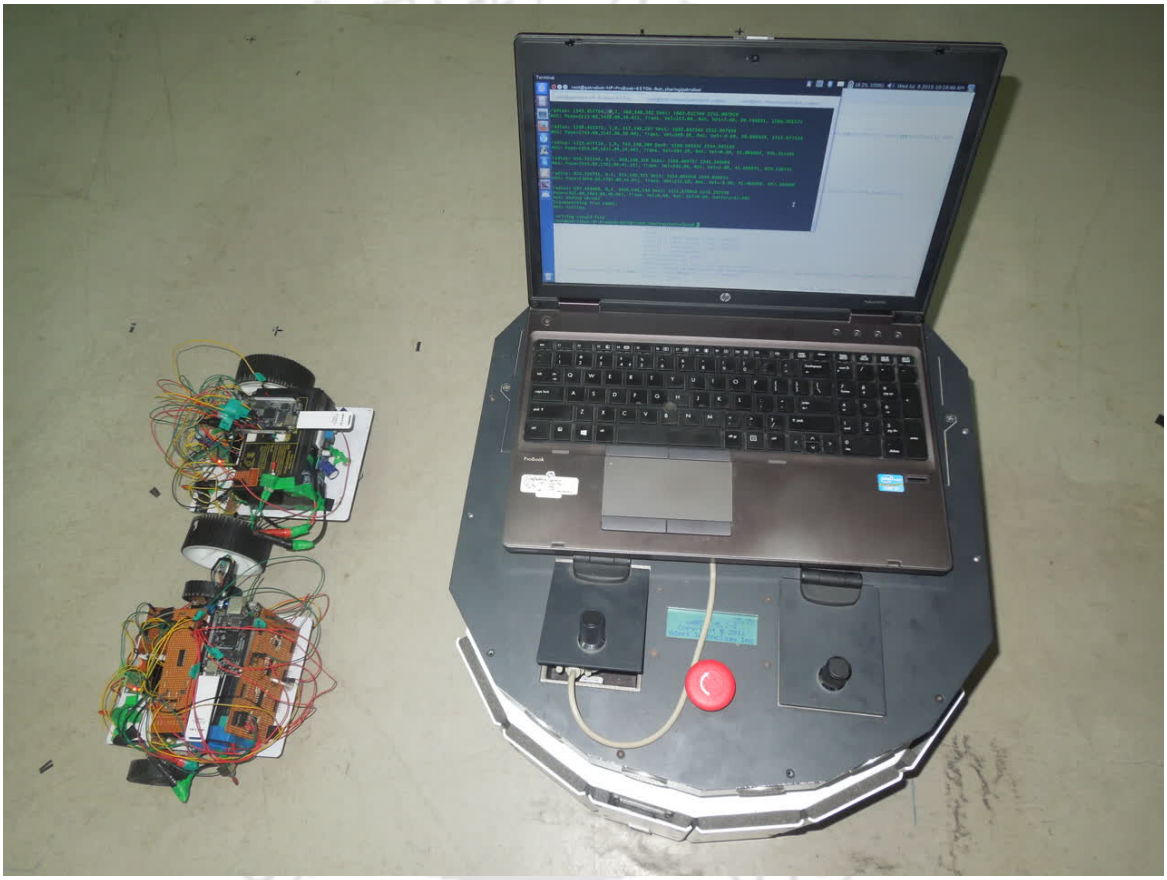


Figure 3.13: Multi-agent system hardware setup

3.7.1 Research Patrolbot

Patrolbot is an advanced robot equipped with a differential drive made of high torque, high speed reversible DC motors. It works on a client-server relation with a μARCS microcontroller as client and a computer as server. The microcontroller is responsible for handling the low-level control such as speed, direction and data acquisition from sensors like wheel encoders,

sonar. The computer handles the high-level control like path planning, communicating with other robots, localization and other intelligent controls. The computer and microcontroller are connected serially. All the control from user side is performed on the computer, there is no need for the normal user to access the microcontroller.

The patrolbot is controlled with the help of an open source development environment, Advanced Robotics Interface for Applications (ARIA) that communicates with microcontroller from the computer. It is a C++ based environment with vast number of functions for robot's motion control and provides a great client-side interface with elaborate documentation. The computer used here is a laptop mounted on top of the patrolbot. Technical specifications of the patrolbot are provided in Appendix A.4.

3.7.2 Robot Assembly

Two wheeled robots are built in the laboratory. They communicate among themselves and as well as with the patrolbot. All three of them act as a three-agent system. These two assembled robots are powered by Beaglebone Black (BBB). Each robot is driven by two bi-directional DC motors, which act as a differential drive. Centroid of the robots is assumed to be mid-point of the line connecting both the wheels. Both the motors are mounted under a plywood sheet of size $25cm \times 18cm$. Separation between the wheels is maintained as $24cm$ for one bot and $20cm$ for the other bot. Calculations of the robot motion are performed accordingly. Technical specifications of major hardware components are provided in Appendix A.5.

3.7.3 Robot motion

Motion control of Patrolbot is done using C++ script with ARIA library functions, installed on a laptop. The user has to define the velocity, heading etc., with the help of ARIA classes in the script. All the low-level functions like speed control, heading control etc., are taken care of by the μARCS microcontroller, which communicates with laptop through serial port. Patrolbot is equipped with high resolution quadrature encoders for the accurate measurement of position, μARCS and ARIA calculate the precise value of velocity. Localization is achieved

using dead-reckoning principle with initial conditions provided.

The other two robots are controlled using Beaglebone Black (BBB) loaded with Ubuntu-14.04. The BBB is equipped with main processor (1GHz ARM cortex A-8) and two axillary microcontrollers (Programmable real time units). Sending motion control signals to the drivers and communication with other robots are performed by using a C script, which runs in the OS on ARM processor. General Purpose Input Output (GPIO) pins, PWM are enabled using Device-trees during run time since, standard linux kernel doesn't define all the input-output pins. Device-trees can be very useful if we want to deploy pins for different purposes at different occasions. For example some pins on BBB can be defined as GPIOs or PWM outputs or PRU IOs based on mode definition given in system reference manual.

A Device-tree is defined as per pin configuration given in Table 3.3. Few pins can be used without explicitly defining in the Device-tree. The Device-tree files corresponding to those pins are provided with Ubuntu OS image tailored for Beaglebone Black. Some of them are given in Table 3.4.

Table. 3.3: Device tree pin description

Pin	Mode	Pin usage
P9_11	GPIO fast output pulldown	Right motor control signal B1
P9_12	GPIO fast output pulldown	Right motor control signal B2
P9_17	GPIO fast output pulldown	Left motor control signal A1
P9_18	GPIO fast output pulldown	Left motor control signal A2
P9_24	PRU0 input	Left wheel encoder sensor signal
P9_25	PRU0 input	Right wheel encoder sensor signal

Table. 3.4: Pins used but not defined in device tree

Pin	Name	Pin usage
P8_19	EHRPWM2A	Left motor pwm signal
P9_1	GND	Ground terminal
P9_3	DC_3.3V	3.3V Reference voltage for level shifters
P9_21	EHRPWM0B	Right motor pwm signal

Based on motor driver data sheet, appropriate frequency (5KHz) and duty ratio are assigned from host C-script to both the PWM pins. Direction of motor shaft rotation is controlled using control signals A1, A2, B1 and B2 from host C-script which follow the logic given in Tables 3.5

and 3.6.

Table. 3.5: Motor driver logic pins description for left motor

Right motor control signal A1	Right motor control signal A2	Function
0	0	Soft brake
0	1	Reverse rotation
1	0	Forward rotation
1	1	Hard brake

Table. 3.6: Motor driver logic pins description for right motor

Right motor control signal B1	Right motor control signal B2	Function
0	0	Soft brake
0	1	Reverse rotation
1	0	Forward rotation
1	1	Hard brake

Wheel encoder sensor signals can be read with polling from host C-script. In polling, a signal is checked at regular intervals of time. Pulse count is updated when there is a change in the signal, *high* to *low* or *low* to *high*. Sometimes, the pulses can be miscounted if some process is blocking the processor for longer duration. Missing a *high* between two *lows* or vice versa may happen. In practice, counter misses pulses during packet data transfer and file read-write operations. Moreover, velocity measurement from polling though main processor is erroneous owing to the fact that it is not real-time. To overcome these drawbacks, Programmable real time units are used for pulse counting and motor shaft angular velocity measurement. The measurements can be further used to estimate position (using dead-reckoning with initial conditions) and velocity of the robot. The main processor can become almost real-time (with few exceptions) by applying *rt patch* to the OS kernel, but PRUs are preferred due to their high accuracy and to distribute the load of computation.

The Programmable real time units PRU0 and PRU1 on BBB are independent of main processor as well as of each other. The clocks of the PRUs are set at $200MHz$, which implies each clock cycle takes 5ns. Normal arithmetic instructions take typical one clock cycle, data operations on data memory takes 2 clock cycles for reading and 3 clock cycles for writing. For data operations on external memory, it takes more clock cycles depending on bus congestion,

typically 10 to 15 cycles. The PRUs can be programmed using assembly level coding or C-script compiled on TI-CGT compiler provided by Texas Instruments. Assembly level coding is used in this work to have more control over number of instructions. Communication among main processor and PRUs can be done using *remoteproc* architecture or shared memory access. Shared memory access is chosen due to its simplicity in implementation and lack of proper documentation for *remoteproc* at the time of implementation.

A loop is defined in assembly script in such a manner that PRU0 polls each wheel sensors at every $80ns$ for pulse count. A loop counter stores the number of loop iterations executed between the pulses. The pulse count and loop counter values are stored in predefined memory location in the data memory of PRU0. The host C-script reads those memory locations to get pulse count and time gap between the pulses. The linear velocity of left and right wheels, separated by distance d, is calculated using number of pulses in a given interval of time. Using left and right wheel velocities (v_{iL}, v_{iR}), linear velocity of the robot can be calculated using Eqn. (3.7) and angular velocity is calculated using Eqn. (3.8). The pulse count is used to calculate position of robot with dead-reckoning principle. The calculated position, linear and angular velocities of the robot are used in Eqn. (3.1).

$$v_i = \frac{v_{iL} + v_{iR}}{2} \tag{3.7}$$

$$\omega_i = \frac{v_{iL} + v_{iR}}{d} \tag{3.8}$$

The host C-script will also take care of motor velocity control using PID control. Robot heading is set at the start and corrected at regular intervals. The back-tracking algorithm is implemented wherever the need arises. In case of history following algorithm, robots use the values stored in memory.

3.7.4 Robot communication

Robot communication is implemented using user datagram protocol (UDP). A local Ad-hoc network is defined by manually assigning IP addresses. Packets are sent after every loop itera-

tion and received using non-blocking functions. A $0s$ time-out is implemented to pop-out all the packets received and only the most recent one is considered. All the packets are time stamped to differentiate between recent and older packets.

3.7.5 Implementation Results

Before implementation, the robots have to be time synchronized with each other to start all the robots at same user-defined time. It is achieved by making the laptop of patrolbot as ntp-server and other two robots as clients. To overcome time drift over the course, the clients' time is updated before each run. Implementation is done for dynamically switching topologies based on position of agents.

A comparison on the time to reach consensus for simulation and experimental results is shown in table 3.7

Table. 3.7: A comparison on the time to reach consensus for simulation and experimental results

Switching topology	Simulation time to reach consensus	Experimental time to reach consensus
With no loss of communication	$8.3s$	$14.9s$
With loss of communication and no back-stepping	-	-
With loss of communication and back-stepping	$12.2s$	$34.7s$
With memory enabled robots	$8.3s$	$14.9s$

A three-agent system is also simulated on MRSim for comparison of results. The simulation and implementation results for back-tracking algorithm are depicted in Figs. 3.14 and 3.15. The arena is chosen to be of size $3000 \times 3000 pixels$ whose origin is considered to be at south-west position where pixels in simulation are analogous to millimetres in implementation. Fig. 3.14 shows a case with robots having initial positions $(250,250),(200,2800),(2800,2800)$ and initial angle 0°. For simulation, the values of acceleration and deceleration are chosen to be $50 pixels/s^2$ and reference velocity to be $200 pixels/s$. The parameter values are chosen as $b_1 = 5$, $b_2 = 10$ and $\phi_b = 45^\circ$. For implementation, acceleration and deceleration are dynamic in nature based on PID controller output. PID controller tires to maintain a reference velocity

magnitude of $200mm/s$ in both forward and backward directions.

Fig. 3.15 depicts robots' distances from origin $(0,0)$ versus time. The rectangular boxes show the area where robots lose communication. The robots use back-tracking algorithm when they enter into that region to change its path for reaching destination. It is observed that there is some deviation in path and convergence time between simulation and implementation. The path deviation is due to small errors in differential drive, low resolution of wheel encoders and corresponding PID controller trying to maintain reference velocity of $200mm/s$. These errors increased the number of robot heading corrections, which further increased convergence time.

The simulation and implementation results for memory enabled agents are shown in Figs. 3.16 and 3.17 with same initial conditions and arena. It can be observed that time taken to reach consensus in this scenario is less than that of back-tracking.

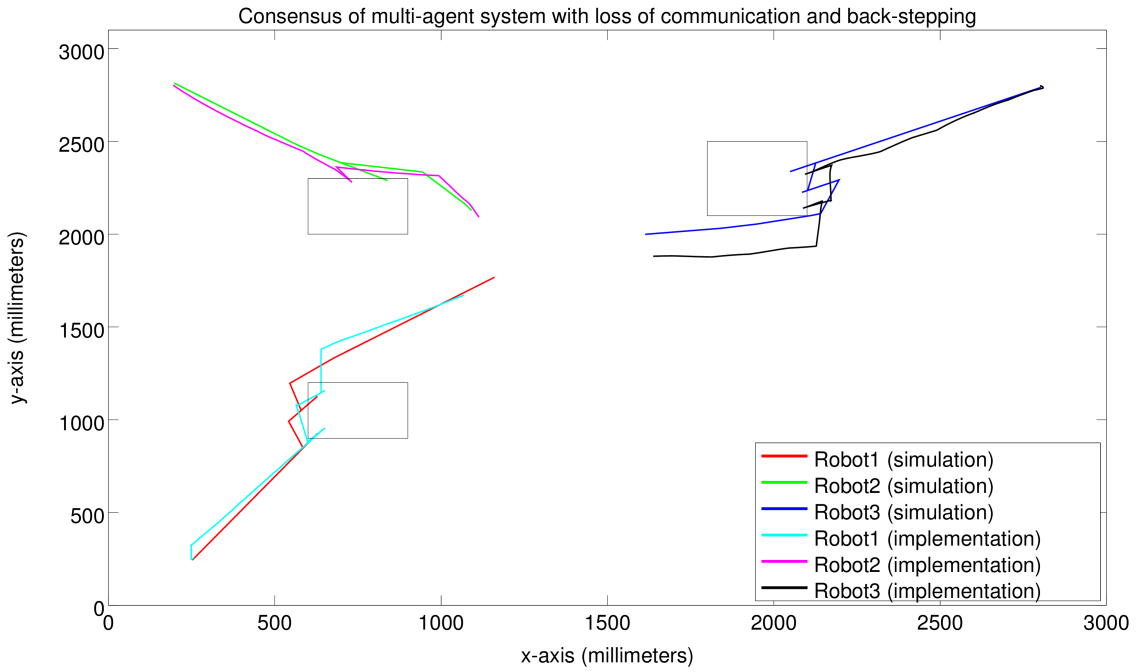


Figure 3.14: Consensus of switching topologies based on position with back-tracking

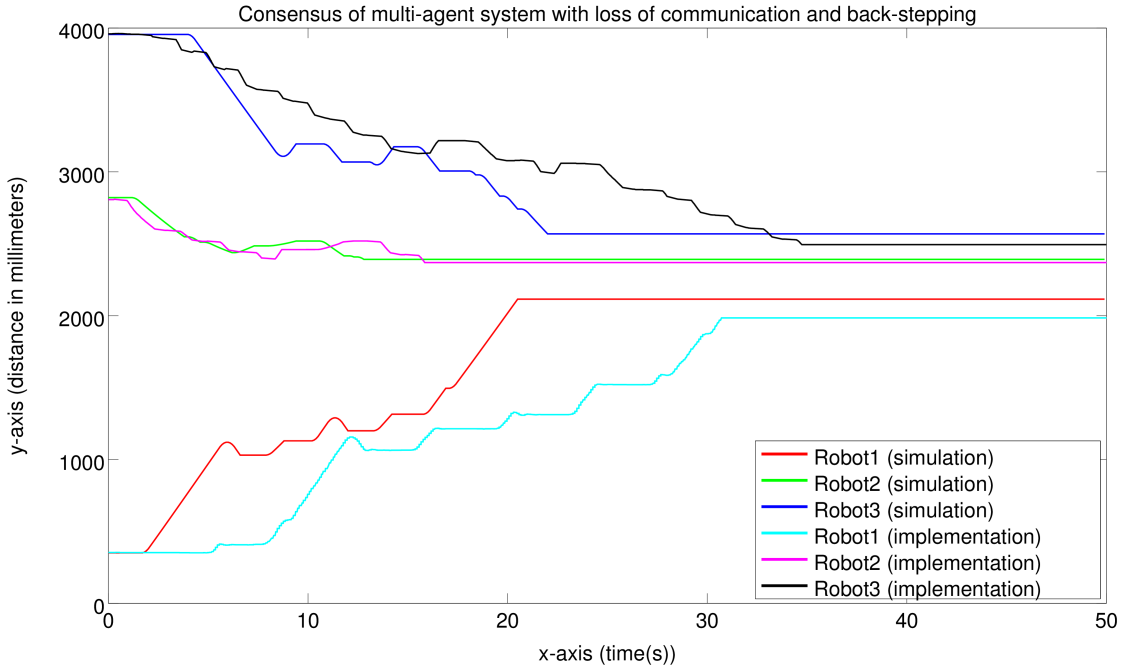


Figure 3.15: Consensus of switching topologies based on position with back-tracking

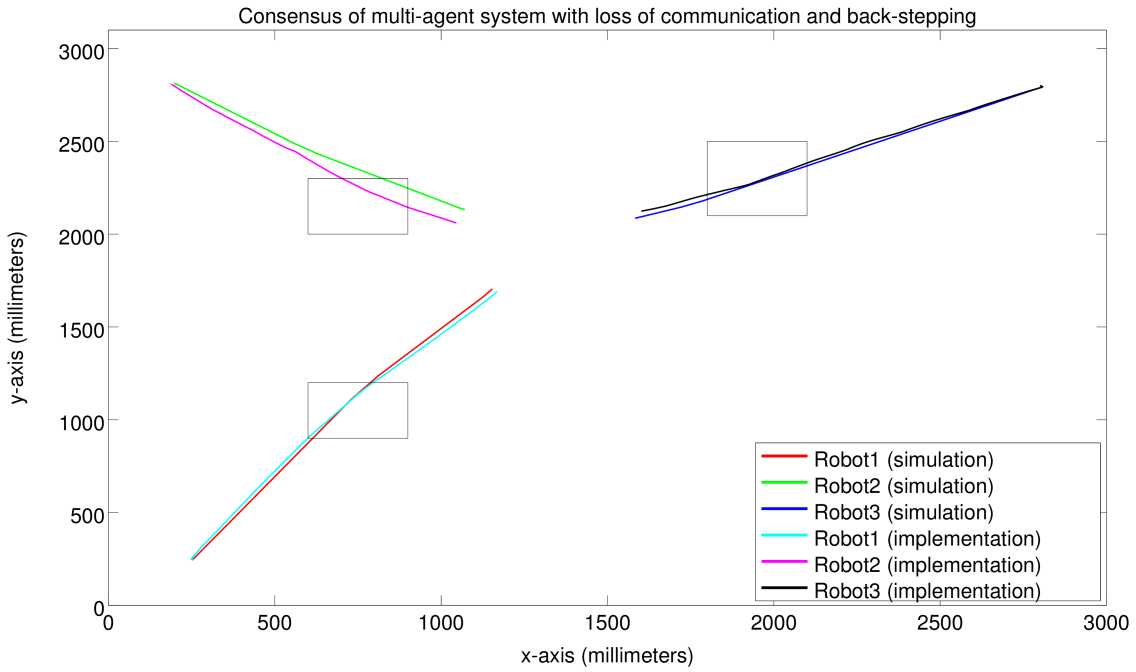


Figure 3.16: Consensus of switching topologies based on position with memory enabled agents

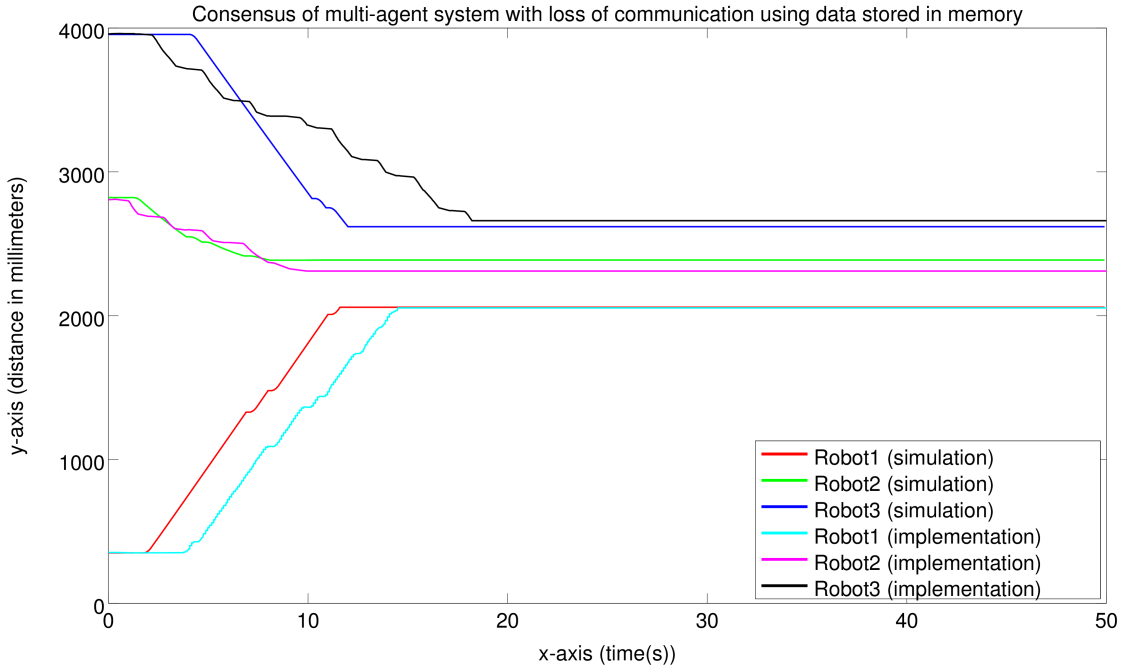


Figure 3.17: Consensus of switching topologies based on position with memory enabled agents

3.8 Conclusion

Two new algorithms, to reach consensus in a multi-agents network with switching topologies, are proposed in this work. Each agent is represented by a second order state space model where the control input is computed using the presented algorithms. A number of static obstacles are assumed to be present in the arena. A sensor based obstacle avoidance algorithm ensures no collision between the agents and the obstacles. Switching occurs depending on the positions of the agents. The algorithms guarantee consensus by letting the agents to either back-track and take a different path or follow the previous consensus point whenever a loss of communication takes place. Simulation results show that when the agents back-track and take another path whenever they lose contact with the network, consensus is reached all the time. However, when they follow the previous history, situations may arise where some of the agents stop at locations near the actual consensus point. Otherwise, the time to reach consensus is less in case of history following with memory enabled agents. Hardware implementation without obstacles is also presented which shows the usefulness of the algorithms for practical purposes. Stability analysis of non-linear systems with time-delays is highly complex. A multi-agent system has some time-delay margin for the system to be stable and reach consensus. An approximate analysis for estimating the time-delay margin of non-linear multi-agent system is performed in the subsequent chapters.

CHAPTER 4

MULTI-AGENT SYSTEM ANALYSIS: FREQUENCY DOMAIN APPROACH

4.1 Introduction

Time-delays in multi-agent systems are practically unavoidable. They can be categorized into two types; input time-delay and communication time-delay. Input time-delay is the amount of time taken by an agent to gather information of its own states, communication time-delay is the amount of time taken by an agent to receive the information about states from other agents. Time-delays in general are uneven, they affect performance and can make the system unstable. Initially, Olfati-Saber *et al.* [57] provided an upper bound for time-delay in first-order systems with equal input and communication time-delays. Subsequently, time-delay has been considered by many researchers [71, 79, 83, 107] and single delay, multiple delays in the systems are explored. Recently, some researchers are taking the time-delays into account while analysing the nonlinear multi-agent systems. A second order system with switching topologies and a single time-varying delay is considered by You *et al.* [96]. Communication and input time-delays are dissimilar in practice. Input time-delays are comparatively smaller than communication time-delays. A nonlinear multi-agent system with dissimilar time-delays has not been considered in the existing literature.

A second order nonlinear multi-agent system with different input and communication time-delays has been considered for the analysis in this chapter. The nonlinearity considered is a

single saturation. Describing function analysis [108] is used to approximate the system into linear and nonlinear blocks. To the best of our knowledge, frequency domain analysis is not used for the case of uneven input and communication time-delays. Frequency domain analysis, as studied in [58,78,80,83,107], has been extended to study the stability of multi-agent systems with uneven delays and saturation. The control laws that enable the use of frequency domain analysis with uneven delays are considered.

In this chapter, we discuss the consensus problem of a distributed multi-agent system with homogeneous nonlinear agents under a fixed directed graph using two control laws. The control laws have an averaging term $\frac{1}{\sum_{j=1}^{n} a_{ij}}$, which act as a scaling factor. The scaling factor makes the analysis simpler and its effect on convergence time, range of time-delay tolerance are discussed in this chapter. The effects are also discussed during comparative analysis in the subsequent chapter. The conditions for a stable range of time delays are derived using Nyquist stability criterion on the linear block and their effectiveness for the nonlinear system is tested. The major contributions of the chapter can be summarized as,

1. Consideration of uneven input and communication time-delays together.
2. Stability analysis of a nonlinear multi-agent system with the help of describing function analysis and nyquist stability criterion.
3. Obtaining necessary and sufficient conditions for stability of the multi-agent system with the proposed control laws.
4. Using the stability conditions to calculate the tolerance range of time-delays.
5. Testing the effectiveness of results using simulations and implementation on a network of four agents.

The rest of the chapter is organised with system model and results of mathematical analysis given in section 4.2. It is followed by simulations, hardware implementations depicted in section 4.3 to support the theoretical results. Finally, section 4.4 gives conclusion to the chapter.

4.2 System model and analysis

For a multi-agent system with adjacency matrix $\mathscr{A}$, another matrix $\tilde{\mathscr{A}}$ is defined whose elements are $\tilde{a}_{ij} = \dfrac{a_{ij}}{\sum_{j=1}^{n} a_{ij}}$ is defined. $[\tilde{\lambda}_1, \tilde{\lambda}_2, ..., \tilde{\lambda}_n]$ are the eigenvalues of matrix $\tilde{\mathscr{A}}$. Let x_i represents the position of i^{th} agent in time domain. Each agent is represented by differential equations in terms of x_i. Assume the notation $x = [x_1\ x_2\ ...\ x_n]^T$, $X = [X_1\ X_2\ ...\ X_n]^T$ to represent n agents in time domain and frequency domain.

Nonlinearities are very frequently encountered in multi-agent systems. Saturation is the most common nonlinearity observed in agents like robots, unmanned air vehicles, unmanned underwater vehicles and so on. Velocity of such systems is constrained due to various factors. Maximum RPM of engine, structural integrity to withstand high velocities etc., are some of the factors that limit the maximum velocity of mobile agents. The aim of the analysis presented in this chapter is to derive the conditions for consensus reachability when the agents are affected by velocity saturation and time-delays. Consider a network of n second order homogeneous agents, where i^{th} agent have the dynamics as given below:

$$\begin{aligned} \dot{x}_i(t) &= sat(\hat{v}_i(t)) \\ \dot{\hat{v}}_i(t) &= u_i(t) \end{aligned} \tag{4.1}$$

where $x_i(t) \in \Re$ represents position state, $sat(\hat{v}_i(t)) \in \Re$ represent velocity state and $u_i(t) \in \Re$ is the control protocol. It is assumed that $x_i(t) = x(0)$ and $\hat{v}_i(t) = 0, \forall t \in (-\infty, 0]$. The saturation function $sat(\hat{v}_i(t))$ is defined in Eqn. (4.2), where $-K$, and K represent lower and upper bounds of the function respectively. In an experimental setup, the value of K is dependent on limitations of the hardware.

$$sat(\hat{v}_i(t)) = \begin{cases} -K, & \text{if } \hat{v}_i(t) \leq -K \\ \hat{v}_i(t), & \text{if } -K < \hat{v}_i(t) < K \\ K, & \text{if } \hat{v}_i(t) \geq K \end{cases} \tag{4.2}$$

The saturation described in Eqn. (4.2) ensures that the system states do not explode with time and exhibit limit cycles when system is pushed into marginally stable region due to time delays. The dynamics of i^{th} agent can be represented in a block diagram as shown in Fig. 4.1.

By applying the concept of describing function analysis, we can approximate and rearrange the block diagram as depicted in Fig. 4.2. Since the nonlinearity is single-valued, it can be approximated with a describing function as given in [108] and always lie on real axis. The approximate gain $N(A)$ of the saturation nonlinearity is given by Eqn. (4.3) [108], where A represents the amplitude of limit cycles.

$$N(A) = \frac{2}{\pi}\left[\arcsin\left(\frac{K}{A}\right) + \frac{K}{A}\sqrt{1-\frac{K^2}{A^2}}\right] \tag{4.3}$$

Proceeding further, it can be found that the nonlinear system considered here give rise to limit cycles only if the linear system is either marginally stable or unstable [108]. Prediction of existence of limit cycles is done using stability analysis of approximate linear system as shown in sections 4.2.1 and 4.2.2 for different control laws.

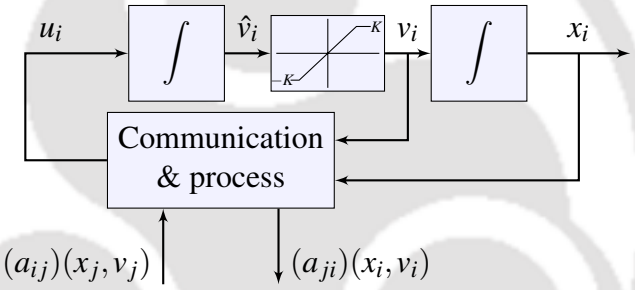


Figure 4.1: Block diagram of i^{th} agent.

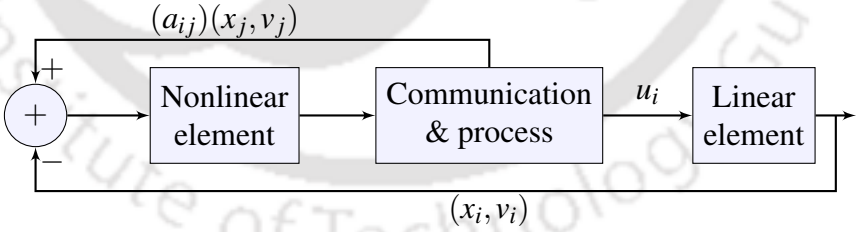


Figure 4.2: Rearranged block diagram of i^{th} agent.

The approximate linear system given in Fig. 4.2 is represented by Eqn. (4.4),

$$\begin{aligned} \dot{x}_i(t) &= v_i(t) \\ \dot{v}_i(t) &= u_i(t) \end{aligned} \tag{4.4}$$

The stability analysis of linear system for different inputs is performed using frequency domain analysis and Nyquist stability criterion which is discussed in sections 4.2.1 and 4.2.2.

4.2.1 Control law 1

A second order multi-agent system is said to reach consensus if, $\forall\,(i,j)\; x_i \to x_j$ and $v_i \to v_j$. In certain applications such as gathering agents to a common position, it is required that $v_i \to 0$ as $x_i \to x_j$. Consider a control input $u_i = u_{i1}$ as given in Eqn. (4.5), that makes $v_i \to 0$ as $x_i \to x_j$. Assume input time-delay τ_1 and communication delay τ_2 in seconds (sec) with $\tau_1 \leq \tau_2$.

$$u_{i1}(t) = -v_i(t-\tau_1) + \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} \left[a_{ij} \left(x_j(t-\tau_2) - x_i(t-\tau_1) \right) \right] \tag{4.5}$$

Theorem 1. *The system represented by Eqns.* (4.4) *and* (4.5) *is stable if and only if,*

$$\frac{|\tilde{\lambda}_k|}{\sqrt{\overline{\omega}^4 - 2\overline{\omega}^3 \sin(\overline{\omega}\tau_1) + \overline{\omega}^2 (1 - 2\cos(\overline{\omega}\tau_1)) + 1}} < 1$$

where $\overline{\omega}$ *satisfies*

$$-\pi = -\overline{\omega}\tau_2 + \arg\left(-\tilde{\lambda}_k\right) - \arctan\left(\frac{\overline{\omega}\cos(\overline{\omega}\tau_1) - \sin(\overline{\omega}\tau_1)}{-\overline{\omega}^2 + \cos(\overline{\omega}\tau_1) + \overline{\omega}\sin(\overline{\omega}\tau_1)} \right)$$

Proof. The system in Eqns. (4.4) and (4.5) can also be represented as,

$$\ddot{x}_i(t) = -\dot{x}_i(t-\tau_1) + \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij} \left(x_j(t-\tau_2) - x_i(t-\tau_1) \right) \tag{4.6}$$

Converting it into s domain for i^{th} agent,

$$\begin{aligned} s^2 X_i(s) - s x_i(0) - \dot{x}_i(0) = & -\left(s X_i(s) - x_i(0) \right) e^{-\tau_1 s} \\ & + \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij} \left(e^{-\tau_2 s} X_j(s) - e^{-\tau_1 s} X_i(s) \right) \end{aligned} \tag{4.7}$$

Consider $\tilde{\mathscr{A}}$ with elements $\tilde{a}_{ij} = \dfrac{a_{ij}}{\sum_{j=1}^{n} a_{ij}}$. For n agents, the above system can be represented as,

$$s^2 I X(s) - s I x(0) - I \dot{x}(0) = -\left(s I X(s) - I x(0) \right) e^{-\tau_1 s} - \left(I e^{-\tau_1 s} - \tilde{\mathscr{A}} e^{-\tau_2 s} \right) X(s) \tag{4.8}$$

Which can be rearranged as,

$$X(s) = \left(s^2 I + (s+1) I e^{-\tau_1 s} - \tilde{\mathscr{A}} e^{-\tau_2 s} \right)^{-1} I \left(s x(0) + e^{-\tau_1 s} x(0) + \dot{x}(0) \right) \tag{4.9}$$

The characteristic equation of the above system is given by,

$$\left| s^2 I + (s+1) I e^{-s\tau_1} - \tilde{\mathscr{A}} e^{-s\tau_2} \right| = \prod_{k=1}^{n} \left(s^2 I + (s+1) I e^{-s\tau_1} - \tilde{\lambda}_k e^{-s\tau_2} \right) \tag{4.10}$$

Eqn. (4.10) follows the same logic as given in Appendix A.3 with,

$$f(s) = \frac{s^2 + (s+1)\,e^{-s\tau_1}}{e^{-s\tau_2}} \tag{4.11}$$

Consider $\forall \tilde{\lambda}_k \neq 0$,

$$\begin{aligned} G_k(s) &= \frac{-\lambda_k}{f(s)} \\ &= \frac{-\tilde{\lambda}_k e^{-s\tau_2}}{s^2 + (s+1)\,e^{-s\tau_1}} \end{aligned} \tag{4.12}$$

Magnitude expression of Eqn. (4.12) is given by,

$$\left|G_k(j\omega)\right| = \frac{|\tilde{\lambda}_k|}{\sqrt{\omega^4 - 2\omega^3 \sin(\omega\tau_1) + \omega^2(1 - 2\cos(\omega\tau_1)) + 1}} \tag{4.13}$$

Phase expression of Eqn. (4.12) is given by,

$$\underline{/G_k(j\omega)} = -\omega\tau_2 + \arg\left(-\tilde{\lambda}_k\right) - \arctan\left(\frac{\omega\cos(\omega\tau_1) - \sin(\omega\tau_1)}{-\omega^2 + \cos(\omega\tau_1) + \omega\sin(\omega\tau_1)}\right) \tag{4.14}$$

Let us assume at $\omega = \overline{\omega}$, the Nyquist plot intersects with negative real axis. The phase at $\omega = \overline{\omega}$ is $-\pi$,

$$-\pi = -\overline{\omega}\tau_2 + \arg\left(-\tilde{\lambda}_k\right) - \arctan\left(\frac{\overline{\omega}\cos(\overline{\omega}\tau_1) - \sin(\overline{\omega}\tau_1)}{-\overline{\omega}^2 + \cos(\overline{\omega}\tau_1) + \overline{\omega}\sin(\overline{\omega}\tau_1)}\right) \tag{4.15}$$

Nyquist stability criterion demands that, magnitude Eqn. (4.13) should satisfy the condition as given in the inequality (4.16).

$$\frac{|\tilde{\lambda}_k|}{\sqrt{\overline{\omega}^4 - 2\overline{\omega}^3 \sin(\overline{\omega}\tau_1) + \overline{\omega}^2(1 - 2\cos(\overline{\omega}\tau_1)) + 1}} < 1 \tag{4.16}$$

4.2.2 Control law 2

In certain applications, $x_i \to x_j$ and $v_i \to v_j$ are sufficient for consensus. Consider control input $u_i = u_{i2}$ in Eqn. (4.17), with input delay τ_1, communication delay τ_2 and $\tau_1 \leq \tau_2$.

$$u_{i2} = \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} \left[a_{ij}\left(v_j(t-\tau_2) - v_i(t-\tau_1)\right) + a_{ij}\left(x_j(t-\tau_2) - x_i(t-\tau_1)\right)\right] \tag{4.17}$$

Theorem 2. *The system represented in Eqns.* (4.4) *and* (4.17) *is stable if and only if,*

$$\frac{|\tilde{\lambda}_k|\sqrt{1+\overline{\omega}^2}}{\sqrt{\overline{\omega}^4 - 2\overline{\omega}^3 \sin(\overline{\omega}\tau_1) + \overline{\omega}^2(1 - 2\cos(\overline{\omega}\tau_1)) + 1}} < 1$$

where $\overline{\omega}$ *satisfies,*

$$-\pi = -\overline{\omega}\tau_2 + \arg\left(-\tilde{\lambda}_k\right) + \arctan\left(\overline{\omega}\right) - \arctan\left(\frac{\overline{\omega}\cos\left(\overline{\omega}\tau_1\right) - \sin\left(\overline{\omega}\tau_1\right)}{-\overline{\omega}^2 + \cos\left(\overline{\omega}\tau_1\right) + \overline{\omega}\sin\left(\overline{\omega}\tau_1\right)}\right)$$

Proof. The above system in Eqns. (4.4) and (4.17) can also be represented as,

$$\ddot{x}_i(t) = \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij}\left[\left(\dot{x}_j(t-\tau_2) - \dot{x}_i(t-\tau_1)\right) + \left(x_j(t-\tau_2) - x_i(t-\tau_1)\right)\right] \tag{4.18}$$

Converting it into s domain for i^{th} agent,

$$\begin{aligned} s^2 X_i(s) - s x_i(0) - \dot{x}_i(0) = \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij}\Big(&\left(sX_j(s) - x_j(0)\right) e^{-\tau_2 s} \\ &- \left(sX_i(s) - x_i(0)\right) e^{-\tau_1 s}\Big) + \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij}\left(e^{-\tau_2 s} X_j(s) - e^{-\tau_1 s} X_i(s)\right) \end{aligned} \tag{4.19}$$

Consider $\tilde{\mathscr{A}}$ with elements $\tilde{a}_{ij} = \dfrac{a_{ij}}{\sum_{j=1}^{n} a_{ij}}$. For n agents, the above system can be represented as,

$$\begin{aligned} s^2 I X(s) - sIx(0) - I\dot{x}(0) = &\left(s\tilde{\mathscr{A}} e^{-\tau_2 s} - sIe^{-\tau_1 s}\right) X(s) - \tilde{\mathscr{A}} x(0) e^{-\tau_2 s} \\ &+ Ix(0) e^{-\tau_1 s} + \left(\tilde{\mathscr{A}} e^{-\tau_2 s} - Ie^{-\tau_1 s}\right) X(s) \end{aligned} \tag{4.20}$$

which can be rearranged as,

$$\begin{aligned} X(s) = &\left(s^2 I + (s+1) Ie^{-\tau_1 s} - (s+1)\tilde{\mathscr{A}} e^{-\tau_2 s}\right)^{-1} \\ &\left(I\left(sx(0) + e^{-\tau_1 s} x(0) + \dot{x}(0)\right) - \tilde{\mathscr{A}} x(0) e^{-\tau_2 s}\right) \end{aligned} \tag{4.21}$$

The characteristic equation of the above system is given by,

$$\left|s^2 I + (s+1) Ie^{-s\tau_1} - (s+1)\tilde{\mathscr{A}} e^{-s\tau_2}\right| = \prod_{i=1}^{n}\left(s^2 I + (s+1) Ie^{-s\tau_1} - (s+1)\tilde{\lambda}_i e^{-s\tau_2}\right) \tag{4.22}$$

Eqn. (4.22) follows the same logic as given in Appendix A.3 with,

$$f(s) = \frac{s^2 + (s+1) e^{-s\tau_1}}{(s+1) e^{-s\tau_2}} \tag{4.23}$$

Considering $\forall \tilde{\lambda}_k \neq 0$,

$$\begin{aligned} G_k(s) &= \frac{-\lambda_k}{f(s)} \\ &= \frac{-(1+s)\tilde{\lambda}_k e^{-s\tau_2}}{s^2 + (s+1) e^{-s\tau_1}} \end{aligned} \tag{4.24}$$

Magnitude expression of Eqn. (4.24) is given by,

$$\left|G_k(j\omega)\right| = \frac{|\tilde{\lambda}_k|\sqrt{1+\omega^2}}{\sqrt{\omega^4 - 2\omega^3\sin(\omega\tau_1) + \omega^2(1-2\cos(\omega\tau_1)) + 1}} \tag{4.25}$$

Phase expression of Eqn. (4.24) is given by,

$$\begin{aligned}\underline{/G_k(j\omega)} = &-\omega\tau_2 + \arg\left(-\tilde{\lambda}_k\right) + \arctan(\omega) - \\ &\arctan\left(\frac{\omega\cos(\omega\tau_1) - \sin(\omega\tau_1)}{-\omega^2 + \cos(\omega\tau_1) + \omega\sin(\omega\tau_1)}\right)\end{aligned} \tag{4.26}$$

Let us assume at $\omega = \overline{\omega}$, the Nyquist plot intersects with negative real axis. The phase at $\omega = \overline{\omega}$ is $-\pi$,

$$-\pi = -\overline{\omega}\tau_2 + \arg\left(-\tilde{\lambda}_k\right) + \arctan(\overline{\omega}) - \arctan\left(\frac{\overline{\omega}\cos(\overline{\omega}\tau_1) - \sin(\overline{\omega}\tau_1)}{-\overline{\omega}^2 + \cos(\overline{\omega}\tau_1) + \overline{\omega}\sin(\overline{\omega}\tau_1)}\right) \tag{4.27}$$

Using magnitude Eqn. (4.25) and Nyquist stability criterion, we get inequality (4.28).

$$\frac{|\tilde{\lambda}_k|\sqrt{1+\overline{\omega}^2}}{\sqrt{\overline{\omega}^4 - 2\overline{\omega}^3\sin(\overline{\omega}\tau_1) + \overline{\omega}^2(1-2\cos(\overline{\omega}\tau_1)) + 1}} < 1 \tag{4.28}$$

Eqn. (4.15) and inequality (4.16) are used to calculate the stable ranges of τ_1, τ_2 with input u_{i1}. Eqn. (4.27) and inequality (4.28) are used to calculate the stable ranges of τ_1, τ_2 with input u_{i2}.

(a) Connected cyclic graph. (b) Directed graph.

Figure 4.3: Graphs of communication topologies.

In section 4.2.1, Eqns. (4.13) and (4.14) have three unknowns (ω, τ_1, τ_2). Similarly in section 4.2.2, Eqns. (4.25) and (4.26) also have three unknowns. For uniform delay $\tau_1 = \tau_2$, a unique solution can be obtained when the magnitude is equated to unity and phase is equated to $-\pi$. In case of non-uniform time-delay, e.g., $\tau_1 < \tau_2$, τ_1 is fixed at some predetermined value and the stable range of τ_2 is found.

For each of the sections 4.2.1 and 4.2.2, the communication topologies considered for simulation and implementation are shown in Fig. 4.3. The arrows in the figure represent the di-

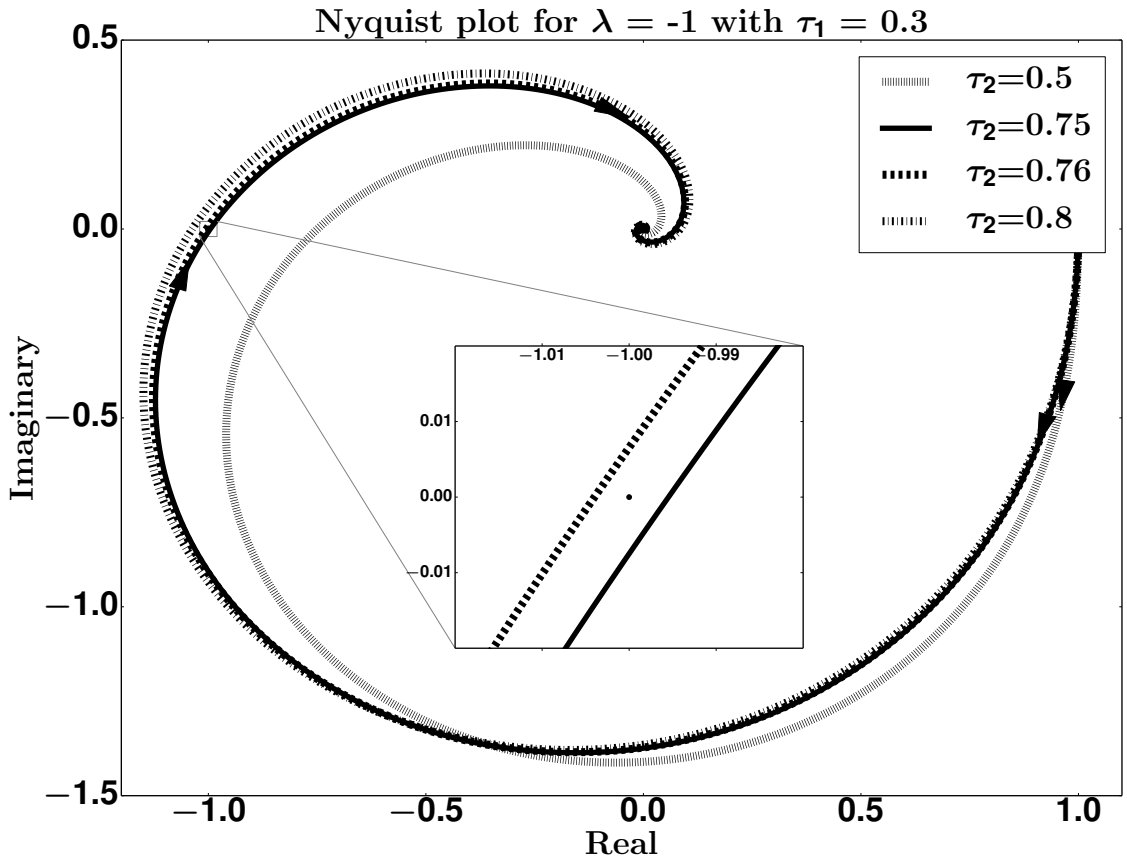


Figure 4.4: nyquist plots with u_{i1} and different time-delays.

rection of information flow and the weights are considered to be unity for all the branches in graph. The graph is directed, strongly connected and balanced with each node receiving states' information from two neighbours and sending states' information to the same neighbours. An unbalanced and/or undirected graph can also be considered. The stability analysis discussed in sections 4.2.1 and 4.2.2 is applicable to suck networks . When $\tau_1 = \tau_2$, the linear system is stable for $\tau_1 < 0.422$ with control input in Eqn. (4.5). Similarly, the linear system is stable for $\tau_1 = \tau_2 < 0.521$ with control input in Eqn. (4.17).

The stability can also be studied graphically with the help of Nyquist plots. Various Nyquist plots are obtained for different values of τ_1 and τ_2 using Eqn. (4.12) in section 4.2.1 for input Eqn. (4.5). Fig. 4.4 depicts Nyquist plots for $\tau_1 = 0.3$ and for different values of τ_2. It is observed that for $\tau_2 > 0.75$, the Nyquist plot encircles -1 (shown by a dot in the figure) and the system becomes unstable. From Fig. 4.6, it can be observed that the system with u_{i1} considered in Eqn. (4.5) is stable for $\tau_1 = \tau_2$ only when $\tau_1 < 0.422$. Using Eqns. (4.13) and (4.14), the

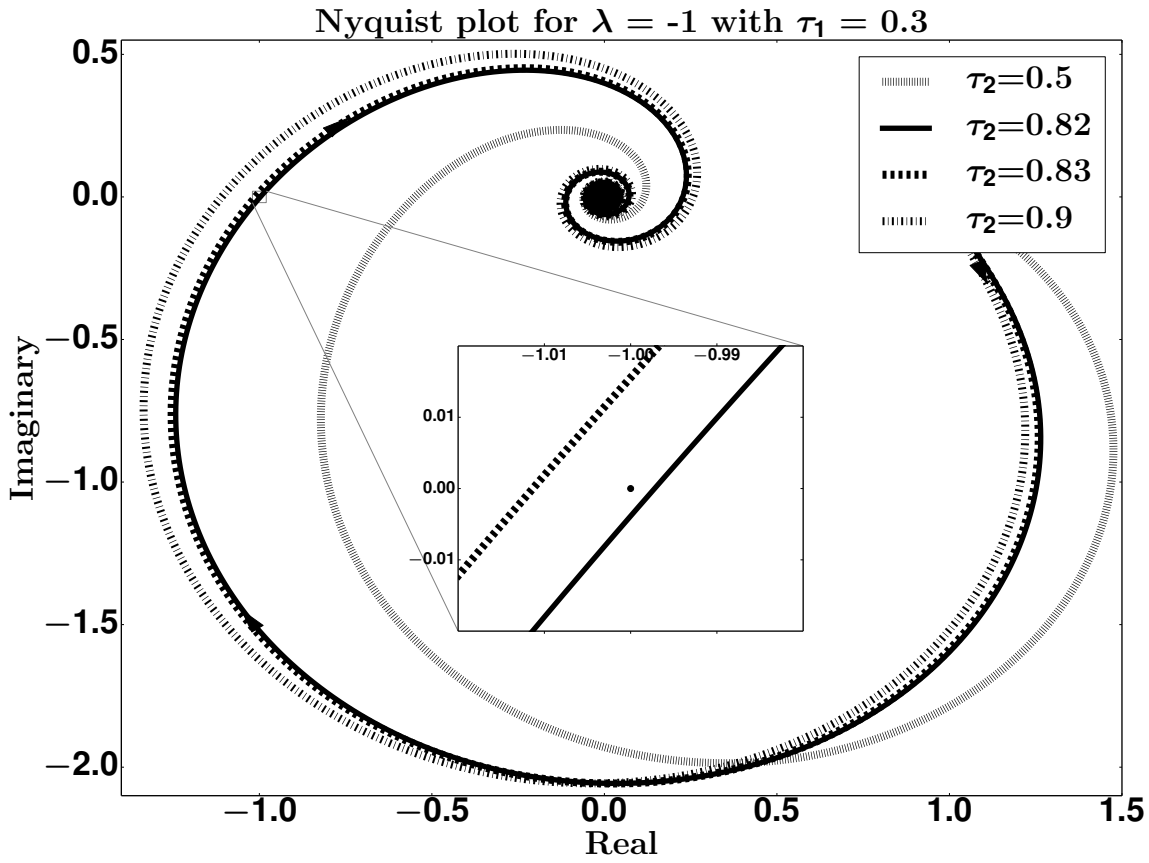


Figure 4.5: nyquist plots with u_{i2} and different time-delays.

stable range of τ_2 will be determined for a given $\tau_1 < 0.422$.

A similar analysis is used with Eqn. (4.24) for input Eqn. (4.17). Figure 4.5 depicts Nyquist plots for $\tau_1 = 0.3$s and for different values of τ_2. The Nyquist plot encircles -1 for $\tau_2 > 0.82$. From Fig. 4.6, it can be observed that the system with u_{i2} considered in Eqn. (4.17) is stable for $\tau_1 = \tau_2$ only when $\tau_1 < 0.521$. The stable ranges are found by assuming some $\tau_1 < 0.521$ and calculating the maximum value of τ_2 for which Nyquist stability criterion is satisfied.

4.3 Simulation and Hardware implementation results

Simulations and corresponding implementations are performed using scripts coded in C language. The initial values are considered as $x_1(0) = 0$, $x_2(0) = 230$, $x_3(0) = 110$, $x_4(0) = 40$ and $\dot{x}_i(0) = 0$, $i \in [1,4]$. The value of K is assumed to be 2 in saturation function $(sat(\,))$. Simulations are performed with communication topology as shown in Fig. 4.3a. The corresponding

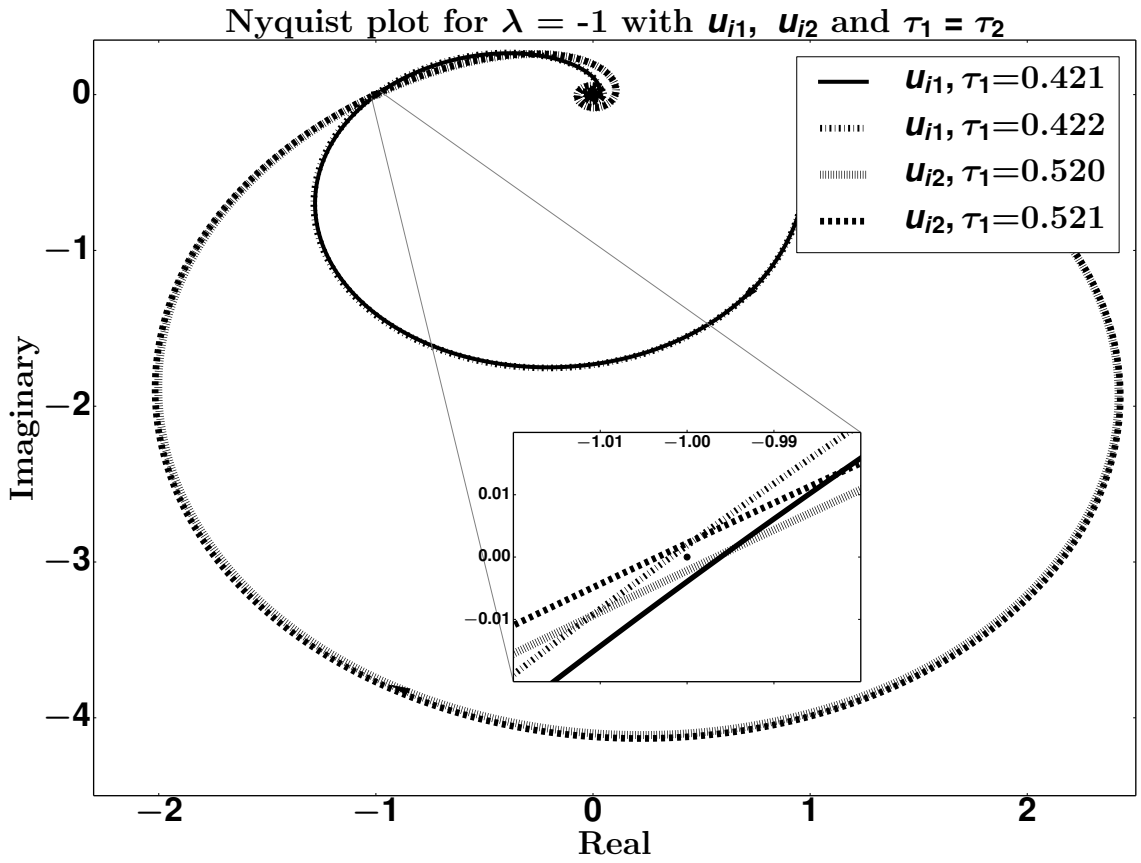


Figure 4.6: nyquist plots with u_{i2}, u_{i2} and different time-delays.

adjacency matrix $\mathscr{A}$ and $\tilde{\mathscr{A}}$ are as given below,

$$\mathscr{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \tilde{\mathscr{A}} = \begin{bmatrix} 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \end{bmatrix}$$

Agent can be a physical system like robot or a virtual system defined with some parameters. The virtual system can be a node in sensor network. To demonstrate the theoretical results, a sensor network is implemented with microcontrollers as nodes. The aim is to have communication among the microcontrollers and bring some parameters of each microcontroller to a common value (reach consensus). Pulse width of PWM wave generated on each microcontroller is assumed as one parameter. Each microcontroller generates PWM wave with certain initial pulse width. The microcontrollers have the capability of controlling the rate of change in pulse width, which serves the purpose of demonstrating the theoretical results. The aim is to bring

consensus of the pulse widths on all microcontrollers by using the control laws in Eqns. (4.5) and (4.17).

In the field of robotics, duty-ratio of PWM wave is most commonly used to control the wheel velocity. PWM pulse width, which affects the duty-ratio, is chosen as the first state variable $x_i(t)$, where as the second state variable $\dot{x}_i(t)$ represents the rate of change of the pulse width. Figs. 4.7, 4.8, 4.11, 4.12, 4.14, 4.15, 4.17 and 4.18 represent plots of $x_i(t)$ and $\dot{x}_i(t)$ for $\tau_1 = 0.3$s and different values of τ_2 using the control law u_i discussed in section 4.2.1. In the same line, Figs. 4.20, 4.21 and 4.23 to 4.26 represent for the control law u_i discussed in section 4.2.2.

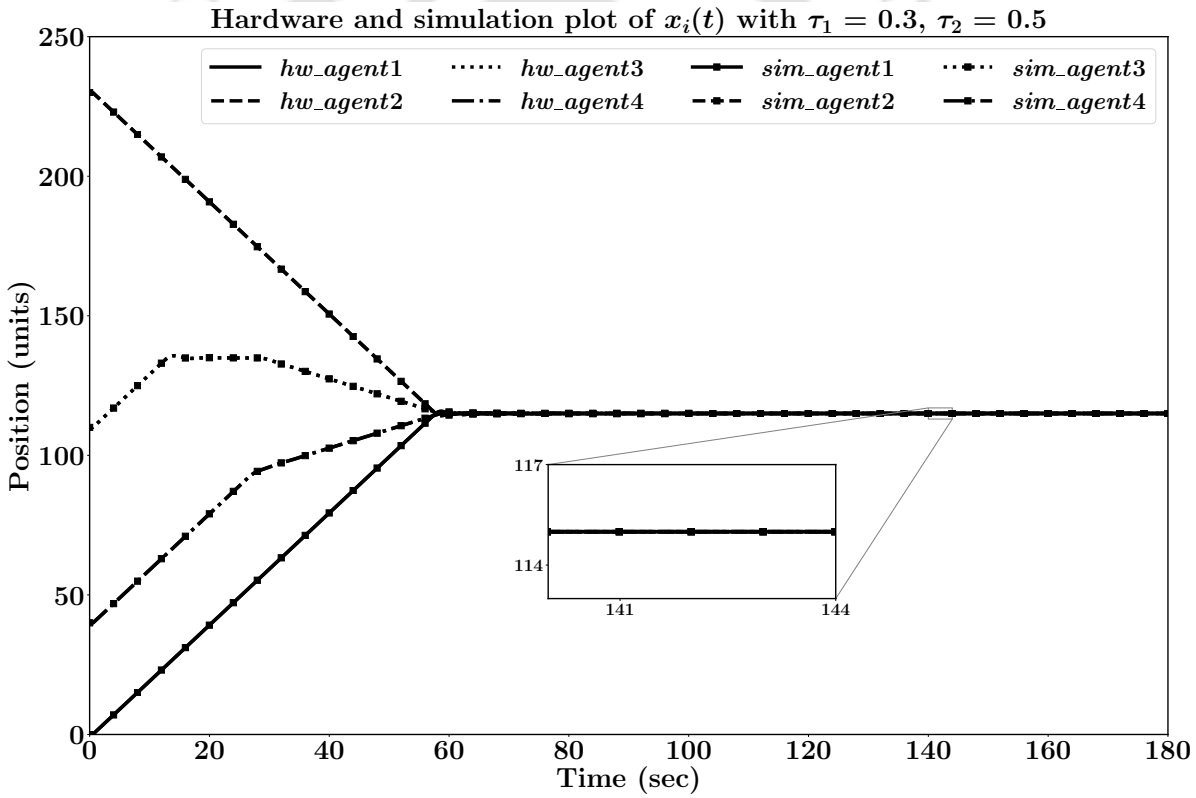


Figure 4.7: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

Hardware implementation is performed using four and five Arduino-Uno boards with each one connected to a computer. Pulse width generated from one of the PWM pins, available on Arduino-Uno is considered as $x_i(t)$. The initial states and step sizes in simulation and implementation are considered to be same for better comparative analysis.

The Arduino-Uno and host computer communicate serially and all the four host comput-

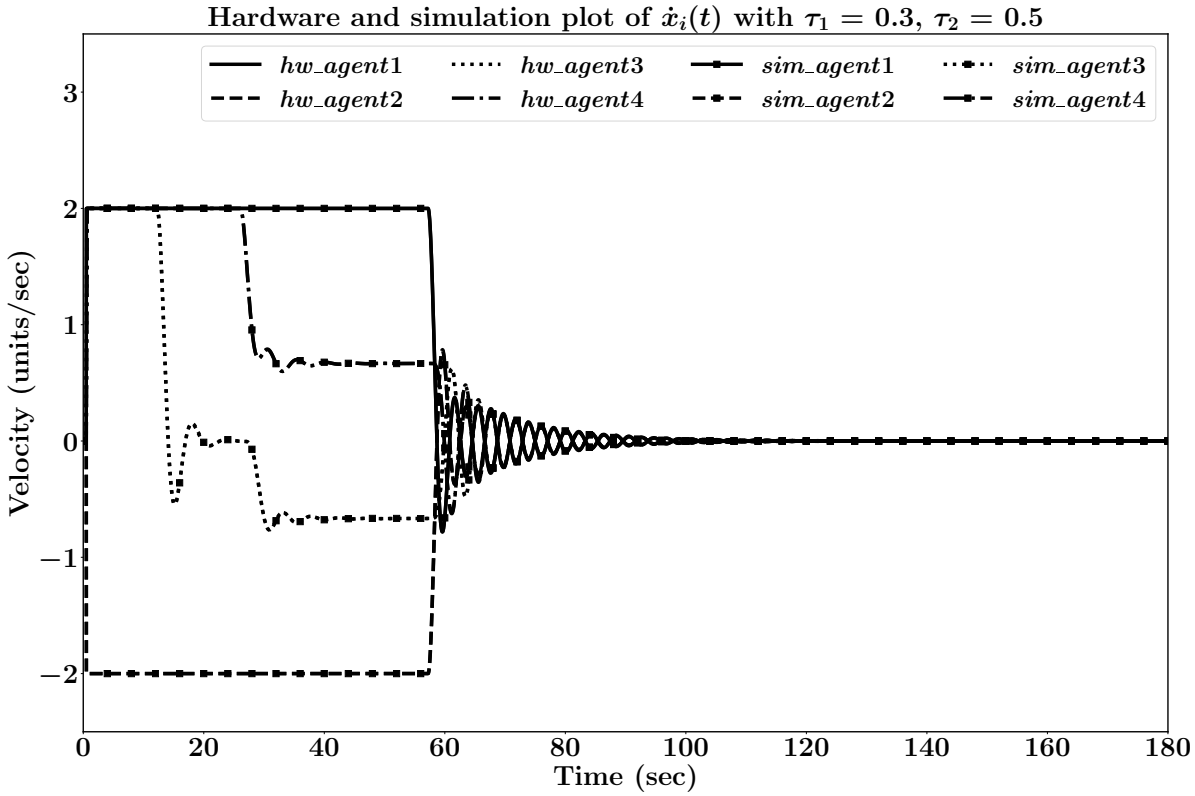


Figure 4.8: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

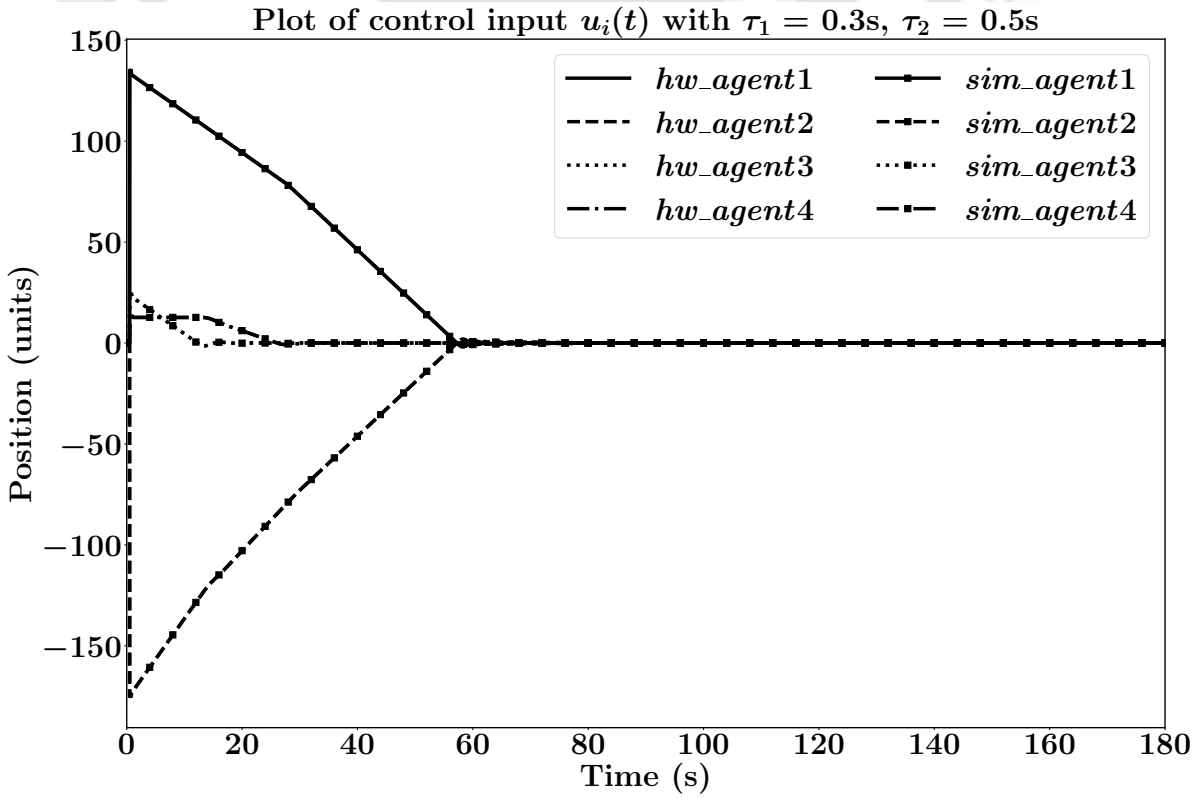


Figure 4.9: Plot of u_{i1} in Eqn. (4.5) for non-uniform time delays.

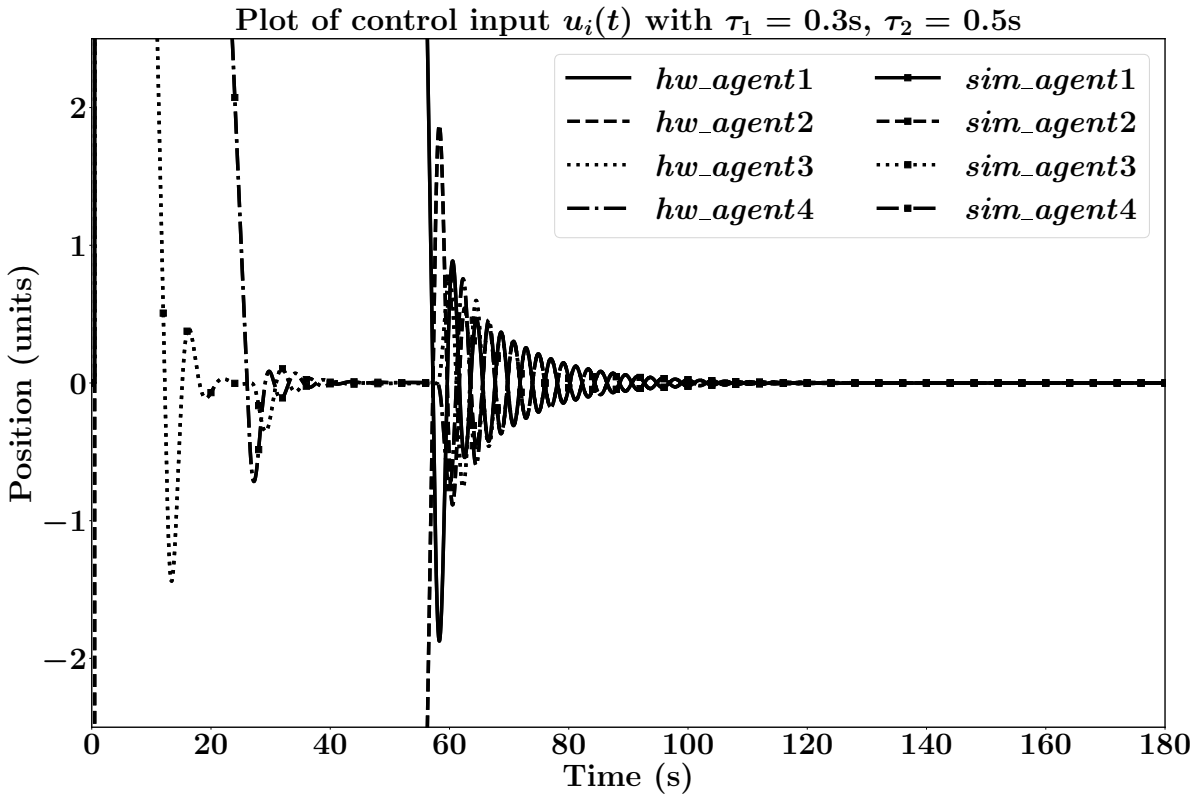


Figure 4.10: Plot of u_{i1} in Eqn. (4.5) for non-uniform time delays.

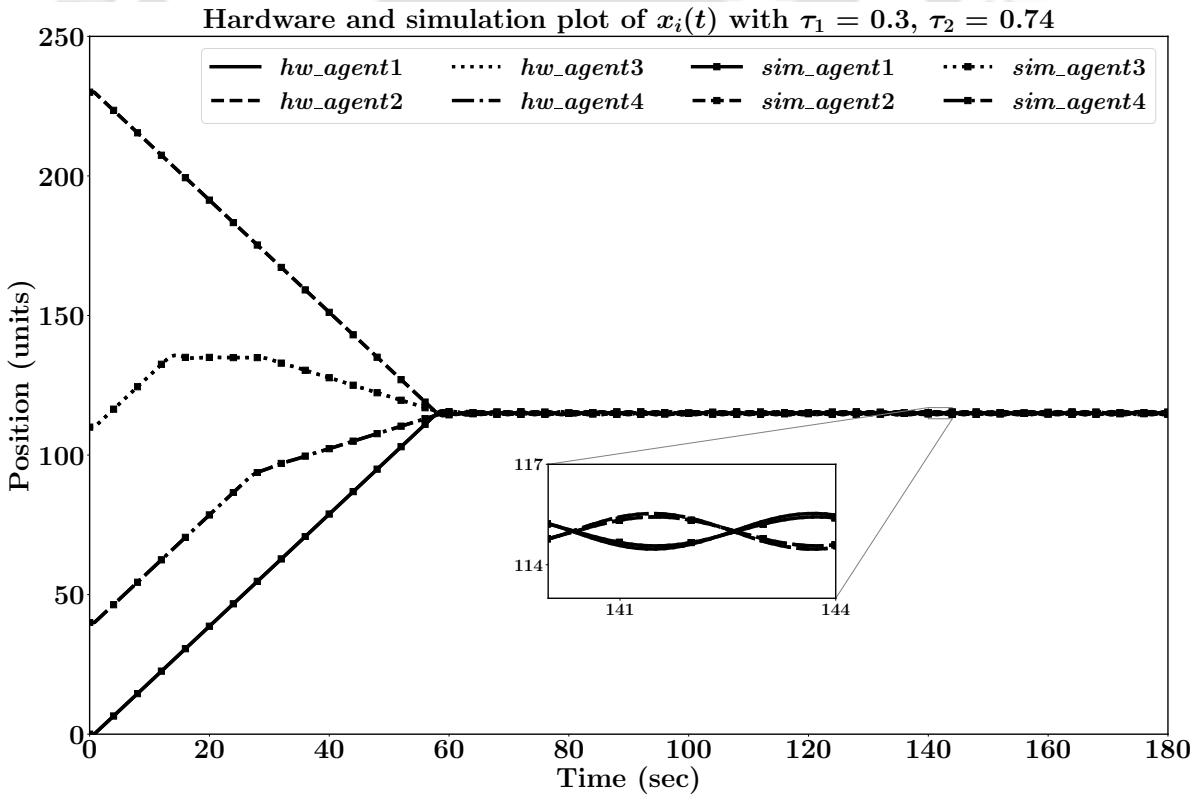


Figure 4.11: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

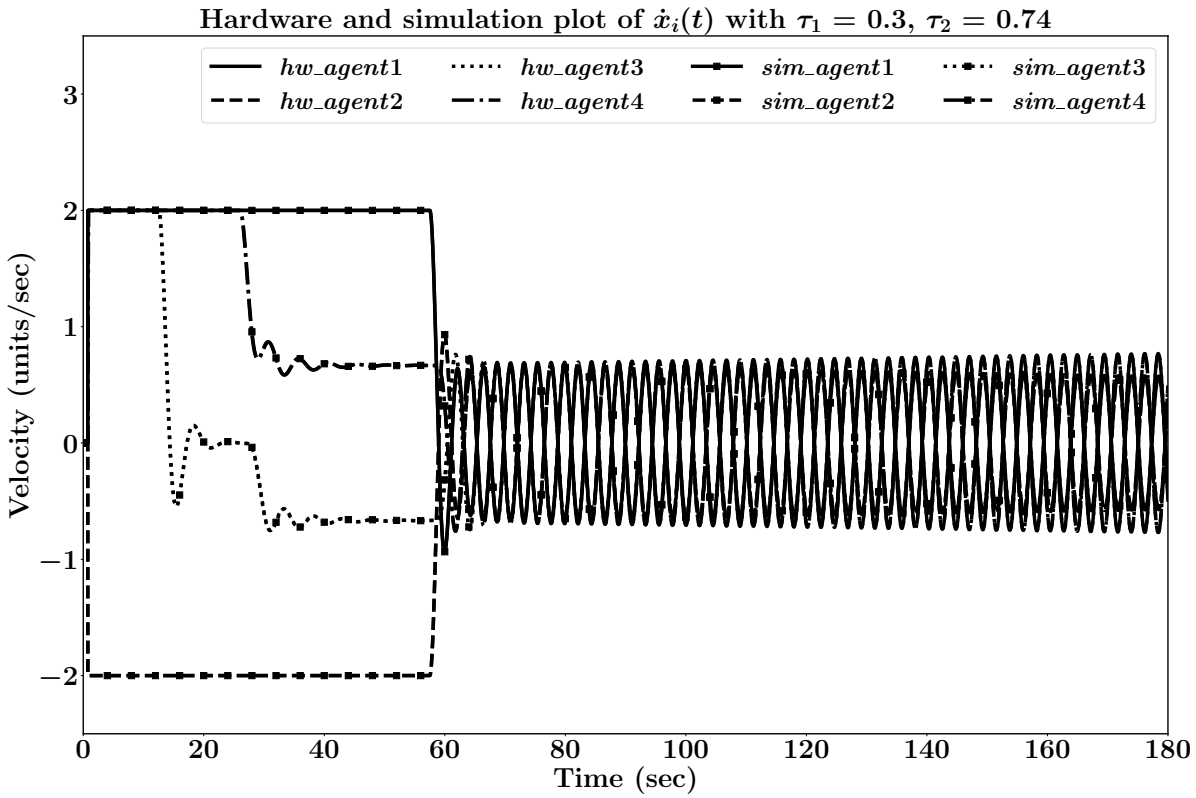


Figure 4.12: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

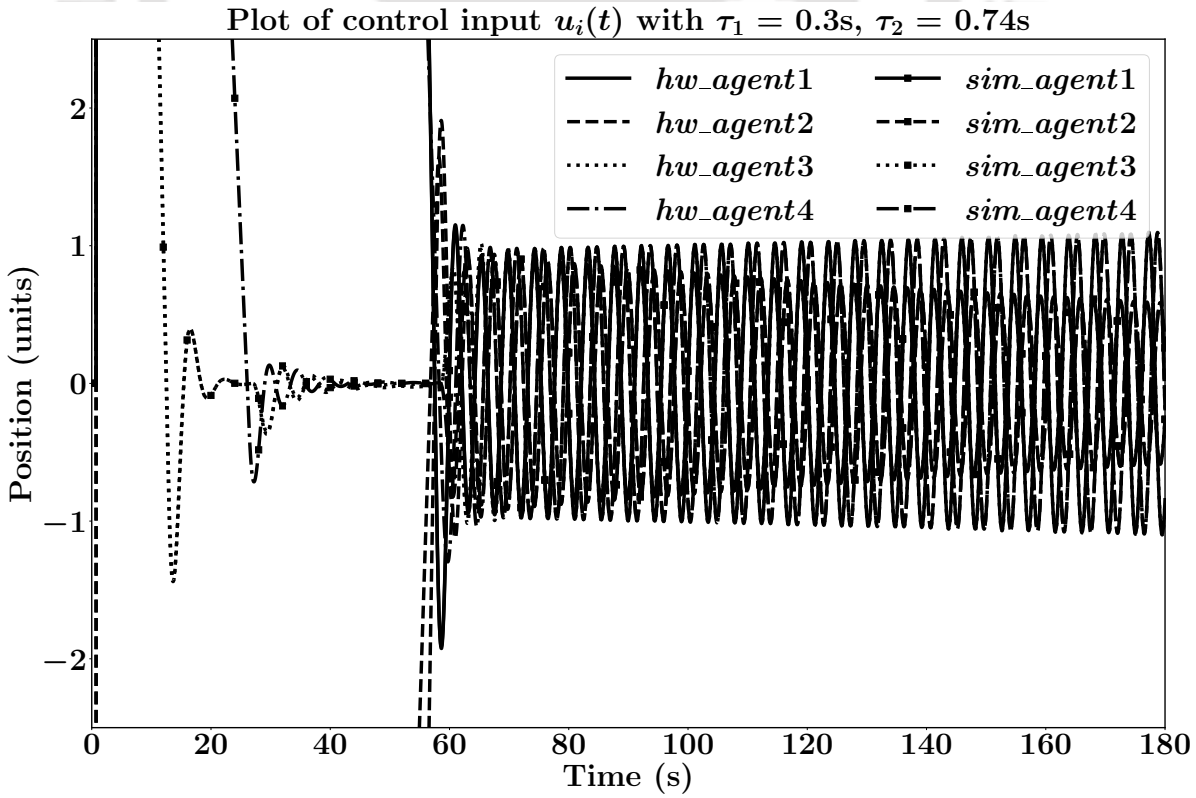


Figure 4.13: Plot of u_{i1} in Eqn. (4.5) for non-uniform time delays.

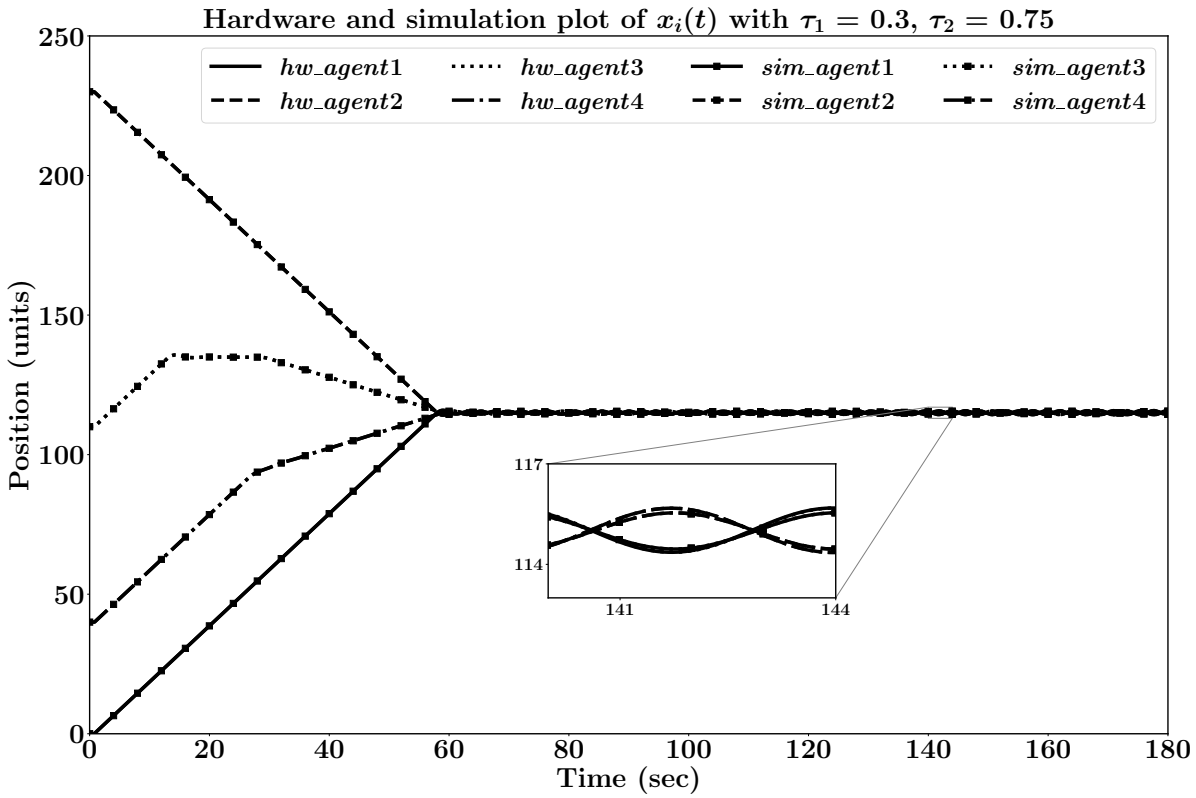


Figure 4.14: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

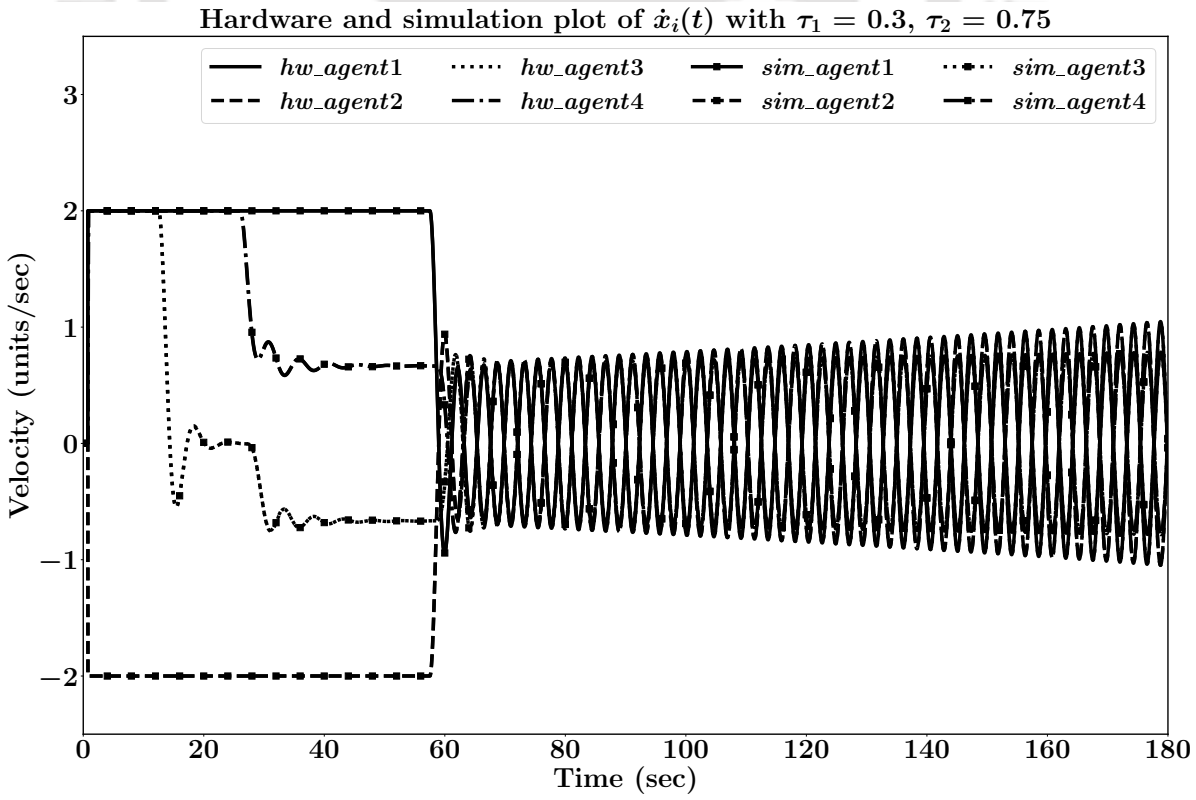


Figure 4.15: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

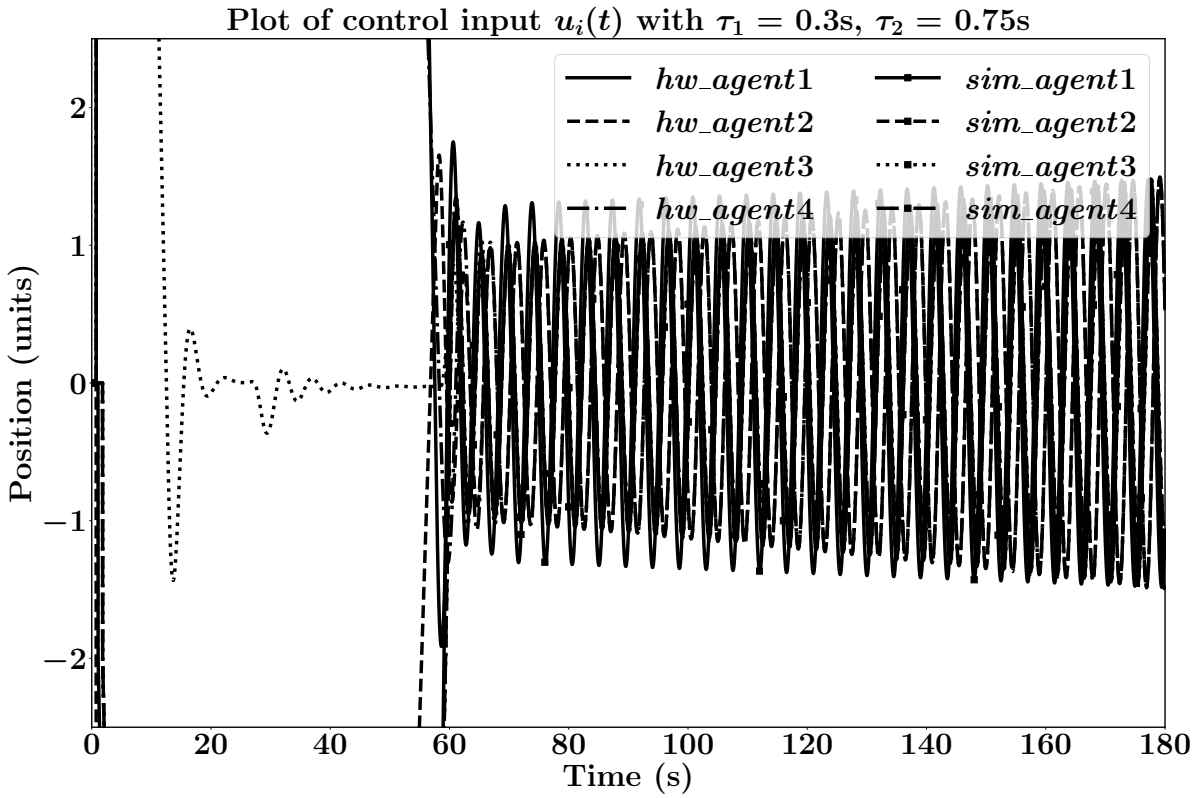


Figure 4.16: Plot of u_{i1} in Eqn. (4.5) for non-uniform time delays.

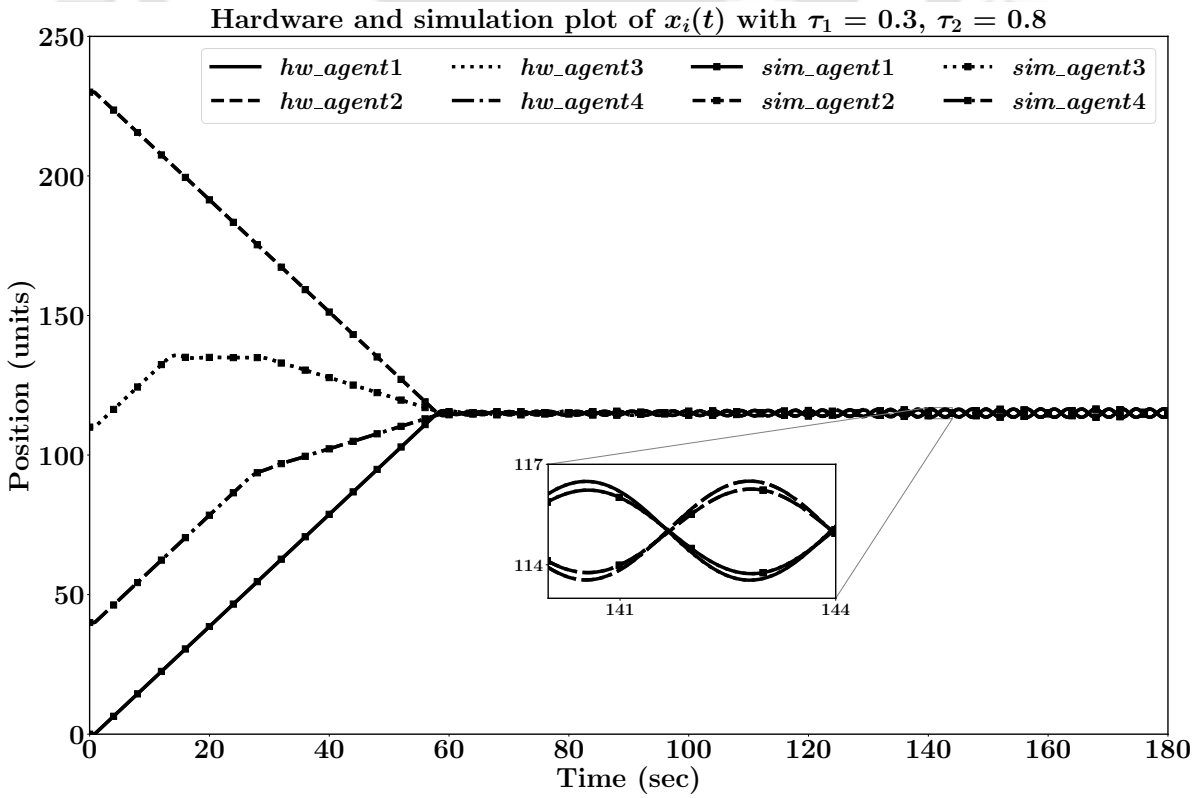


Figure 4.17: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

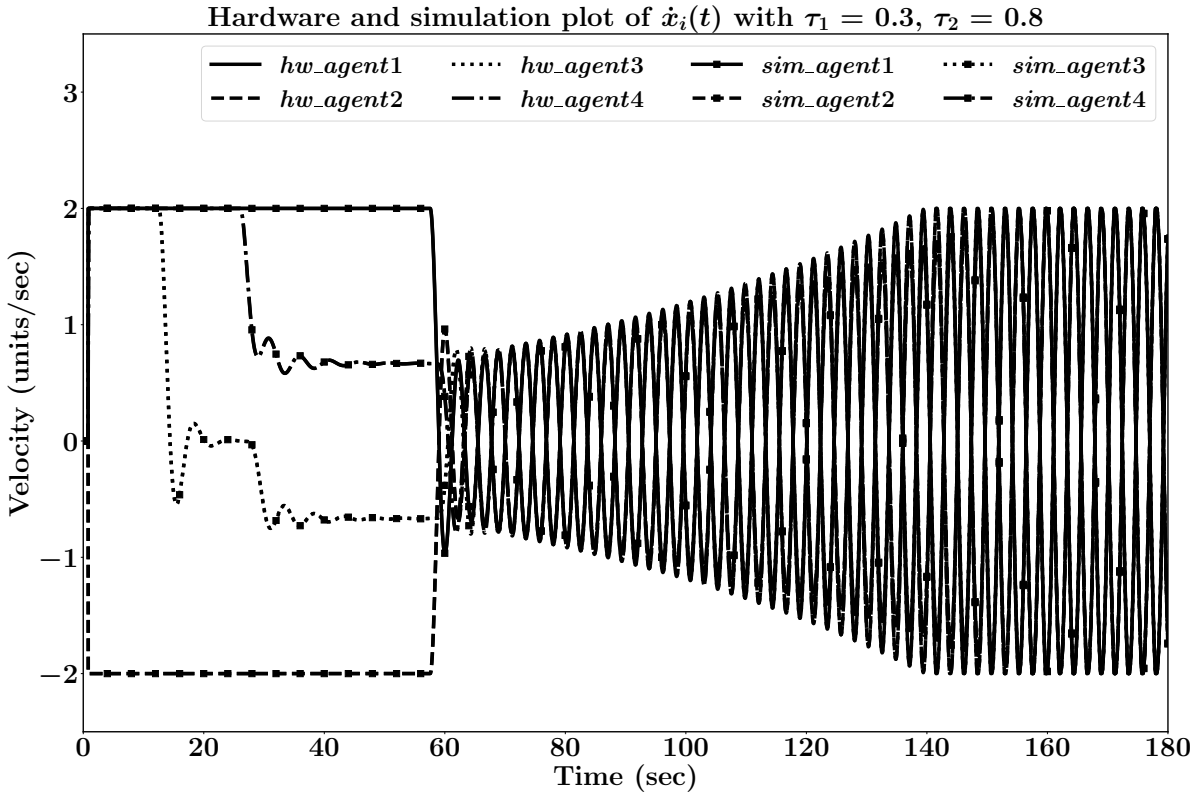


Figure 4.18: Plot with u_{i1} in Eqn. (4.5) for non-uniform time delays.

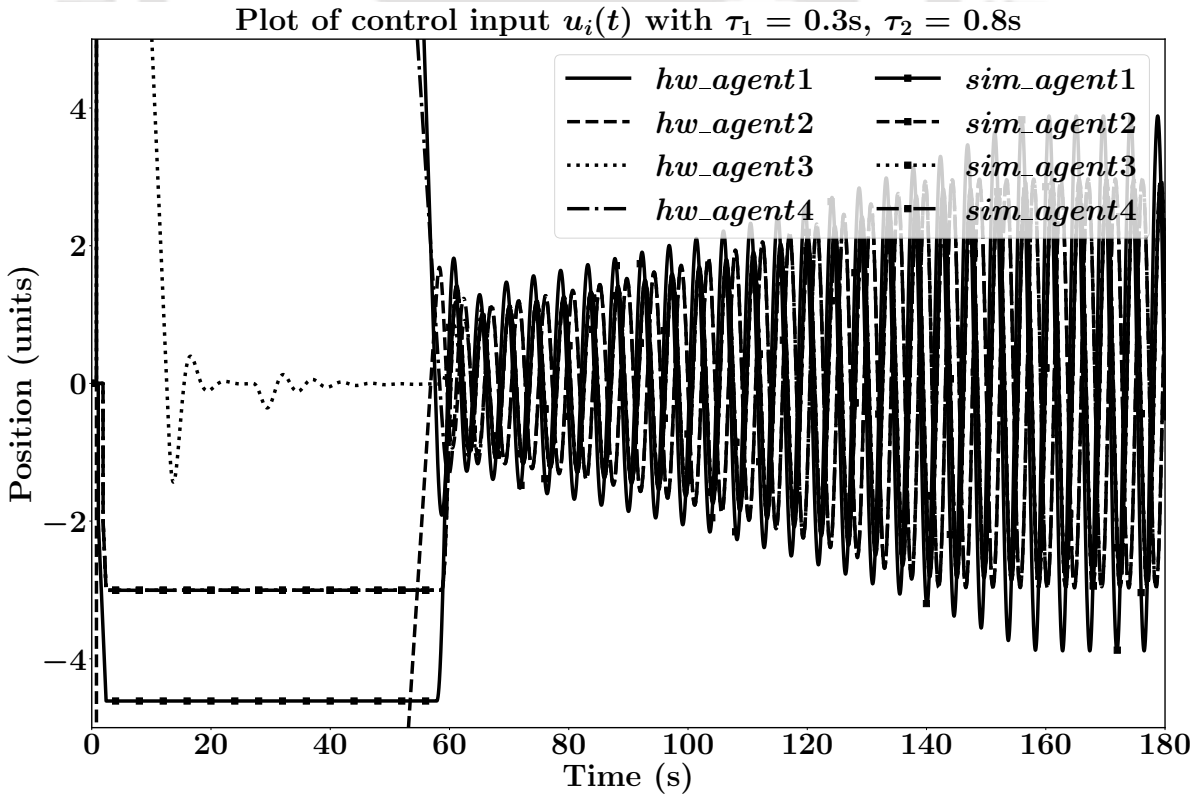


Figure 4.19: Plot of u_{i1} in Eqn. (4.5) for non-uniform time delays.

ers are connected through a LAN switch to exchange information based on the communication graph as shown in Fig. 4.3. Time synchronization is achieved at the start of each experimental run using *ntp* protocol and a local server. Communication among the host computers is performed using *UDP* protocol with precautions being taken to ensure the ordered processing of packets received. Due to practical limitations in hardware implementation, the step size is chosen to be 10ms. While implementing time-delays in experiments, there is some additional time-delay($\approx 4ms$) from the communication links beyond the user defined time-delays.

From Figs. 4.7 and 4.8, it can observed that the multi-agent system states converge at a considerably fast rate with input time-delay $\tau_1 = 0.3$ and communication time-delay $\tau_2 = 0.5$. Figs. 4.9 and 4.10 depict the control inputs. Since input saturation is not considered, from Fig. 4.9 it can be observed that the initial control inputs have very large magnitude compared to the magnitude of control inputs for rest of the duration. To have a better insight of the control inputs, the range on *y-axis* is reduced in Fig. 4.10. It can be observed that the control input reaches zero as the agent reach consensus and hence the goal is achieved. A similar trend can be observed in Fig. 4.13, the control input is non-zero as long as the agents do not reach consensus. The control inputs of rest of the cases also follow a similar trend as shown in Figs. 4.16, 4.19 and 4.22. Whenever the system is stable, the control inputs reach zero and whenever the system is unstable, the control inputs do not reach zero.

Figs. 4.12, 4.15 and 4.24 show a deviation of the hardware results from the simulation results. It is attributed to the unavoidable additional time-delay from communication links. However, for $\tau_1 = 0.3$ and $\tau_2 = 0.75$, the convergence rate is very slow as depicted in Figs. 4.11 and 4.12 due to the fact that $\left|G_i(\overline{\omega})\right| < 1$ but near to unity. It can be observed that there is slight deviation in implementation results compared to simulation results as discussed earlier. In Figs. 4.14 and 4.15, the states don't converge and diverge at a slower pace towards limit cycle behaviour for $\tau_1 = 0.3$ and $\tau_2 = 0.76$ since $\left|G_i(\overline{\omega})\right| > 1$ and near to unity. When $\tau_1 = 0.3$ and $\tau_2 = 0.9$, the states diverge faster towards the limit cycle behaviour and can be observed in Figs. 4.17 and 4.18. The numerical values of $\overline{\omega}$ and $\left|G_i(\overline{\omega})\right|$ for different values of τ_1 and τ_2 are provided in Table 4.1.

τ_1 (sec)	τ_2 (sec)	$\overline{\omega}$	$\lvert G_i(\overline{\omega})\rvert$
0.3	0.5	1.599	0.749
0.3	0.75	1.408	0.996
0.3	0.76	1.403	1.004
0.3	0.8	1.38	1.037

Table. 4.1: Values of $\overline{\omega}$ & $\left|G_i(\overline{\omega})\right|$ for different values of τ_1 and τ_2 with u_{i1} in Eqn. (4.5).

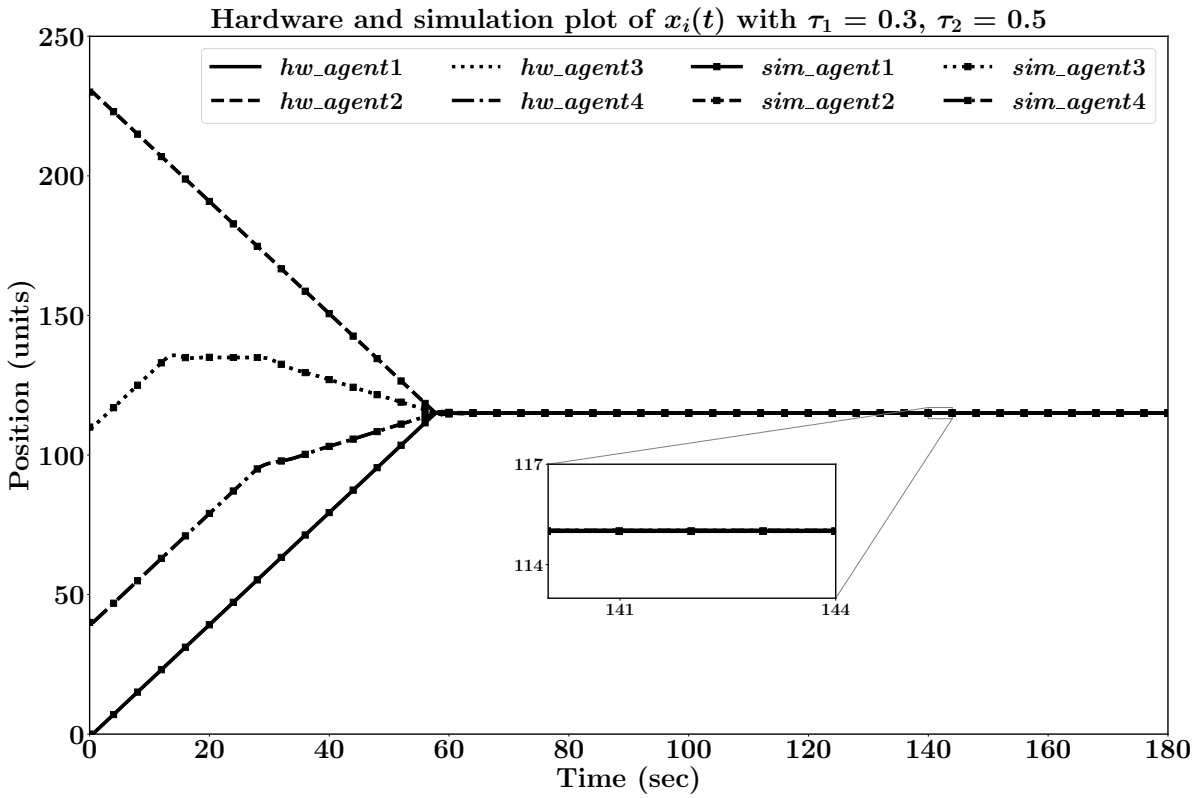


Figure 4.20: Plot with u_{i2} in Eqn. (4.17) for non-uniform time delays.

A similar trend is observed in the results for input u_{i2} in Eqn. (4.17). Some of the results are shown in Figs. 4.20, 4.21 and 4.23 to 4.26 and the numerical values are given in Table 4.2. The states converge for $\tau_1 = 0.3$ and $\tau_2 = 0.5$. Corresponding simulation and implementation results are depicted in Figs. 4.20 and 4.21. For $\tau_1 = 0.3$ and $\tau_2 = 0.83$, divergence rate is slow for simulation results as compared with the hardware implementation. The states diverge slowly towards the exhibition of limit cycles and the corresponding results are shown in Figs. 4.23 and 4.24. States in case of both simulation and implementation diverge at a fast rate towards the exhibition of limit cycles for $\tau_1 = 0.3$ and $\tau_2 = 0.9$ as depicted in Figs. 4.25 and 4.26.

In a similar fashion, few simulations and hardware validations are performed on a five-agent

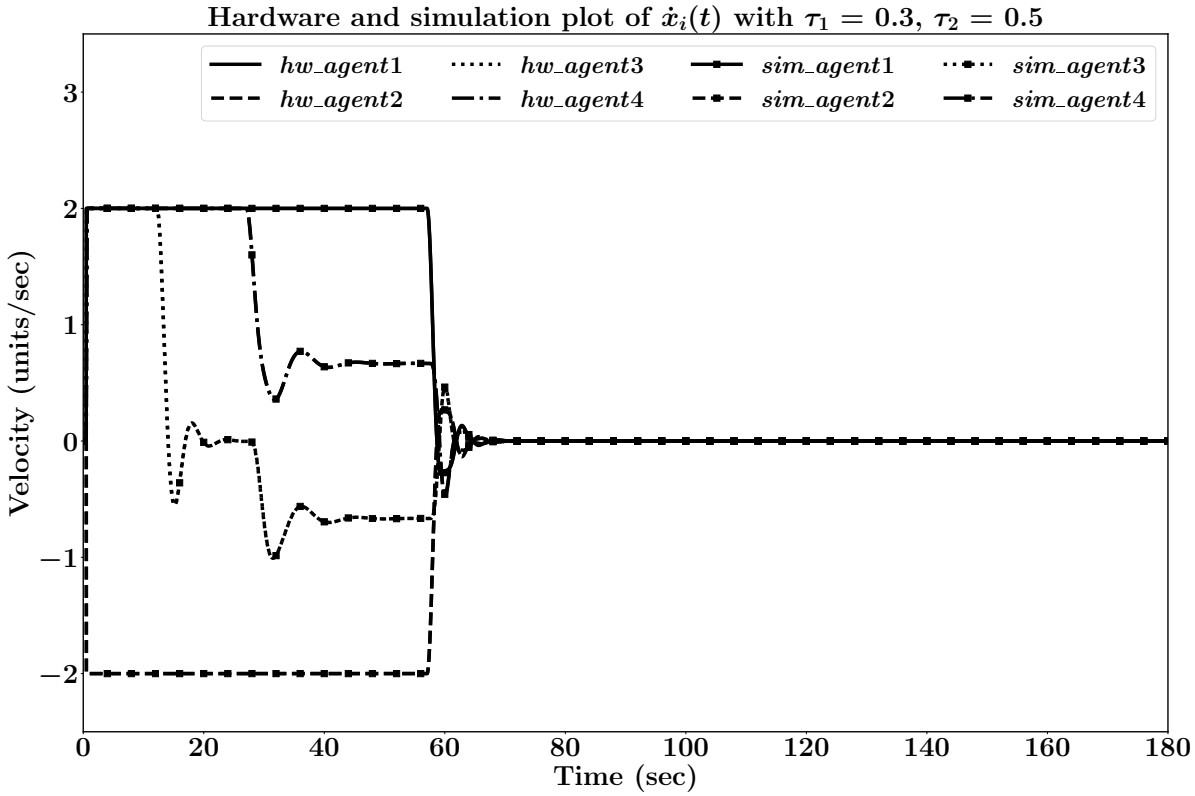


Figure 4.21: Plot with u_{i2} in Eqn. (4.17) for non-uniform time delays.

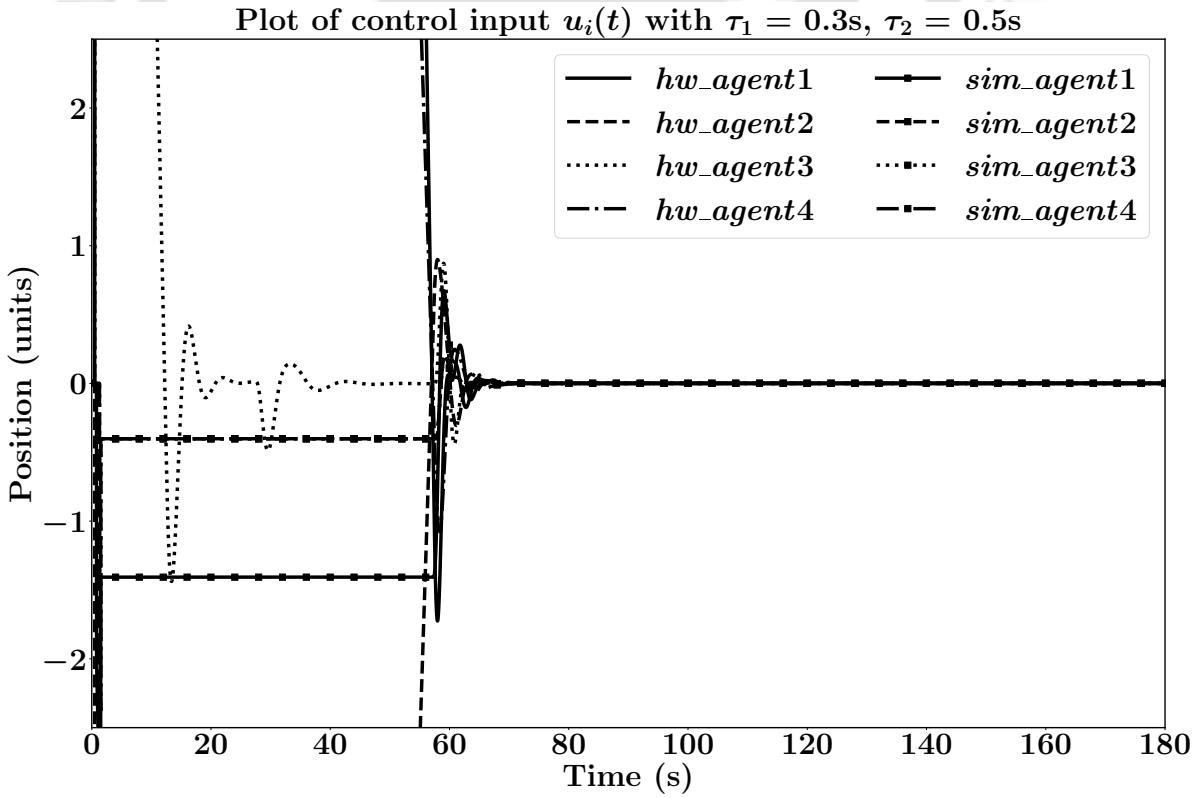


Figure 4.22: Plot of u_{i1} in Eqn. (4.5) for non-uniform time delays.

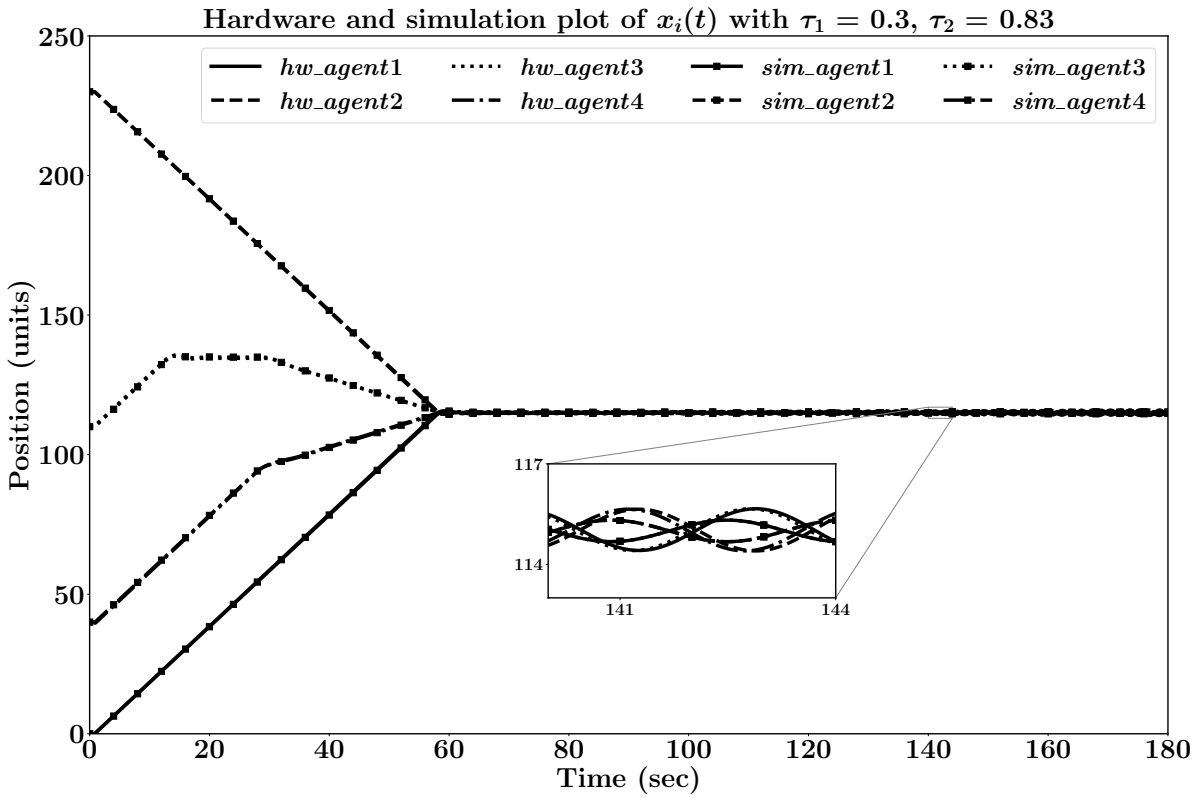


Figure 4.23: Plot with u_{i2} in Eqn. (4.17) for non-uniform time delays.

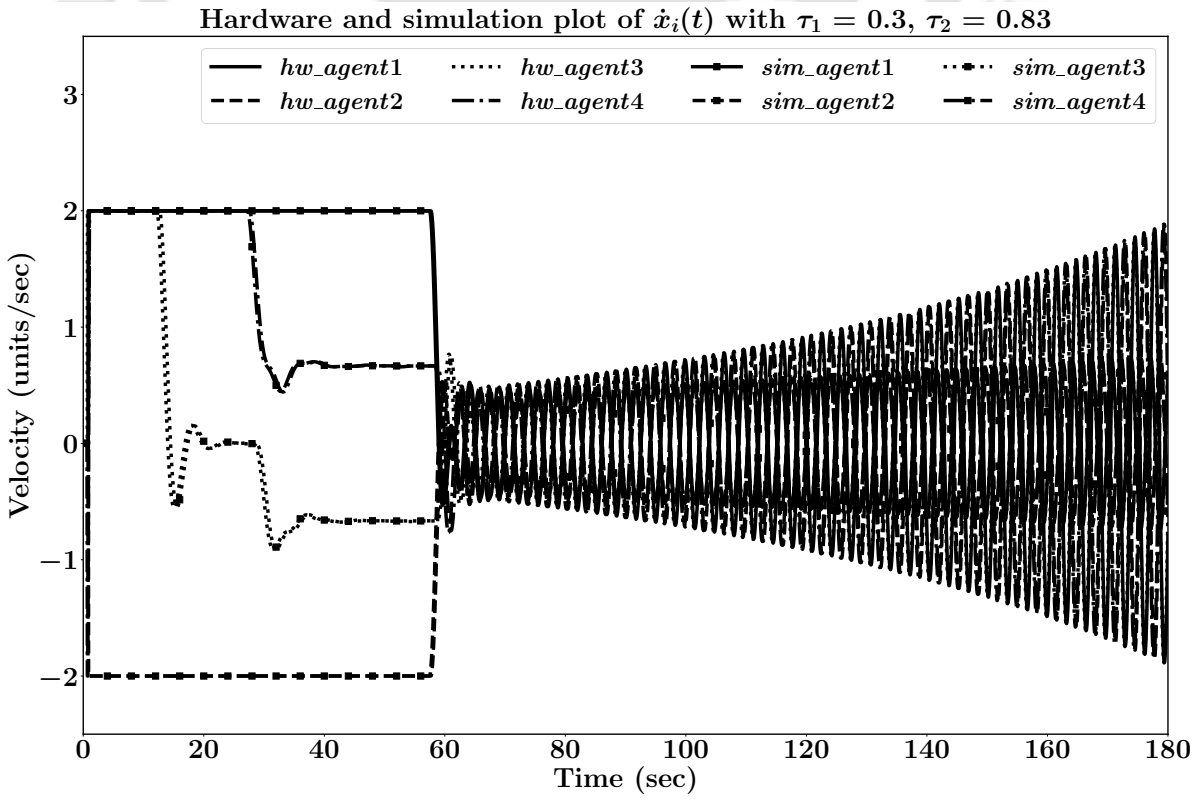


Figure 4.24: Plot with u_{i2} in Eqn. (4.17) for non-uniform time delays.

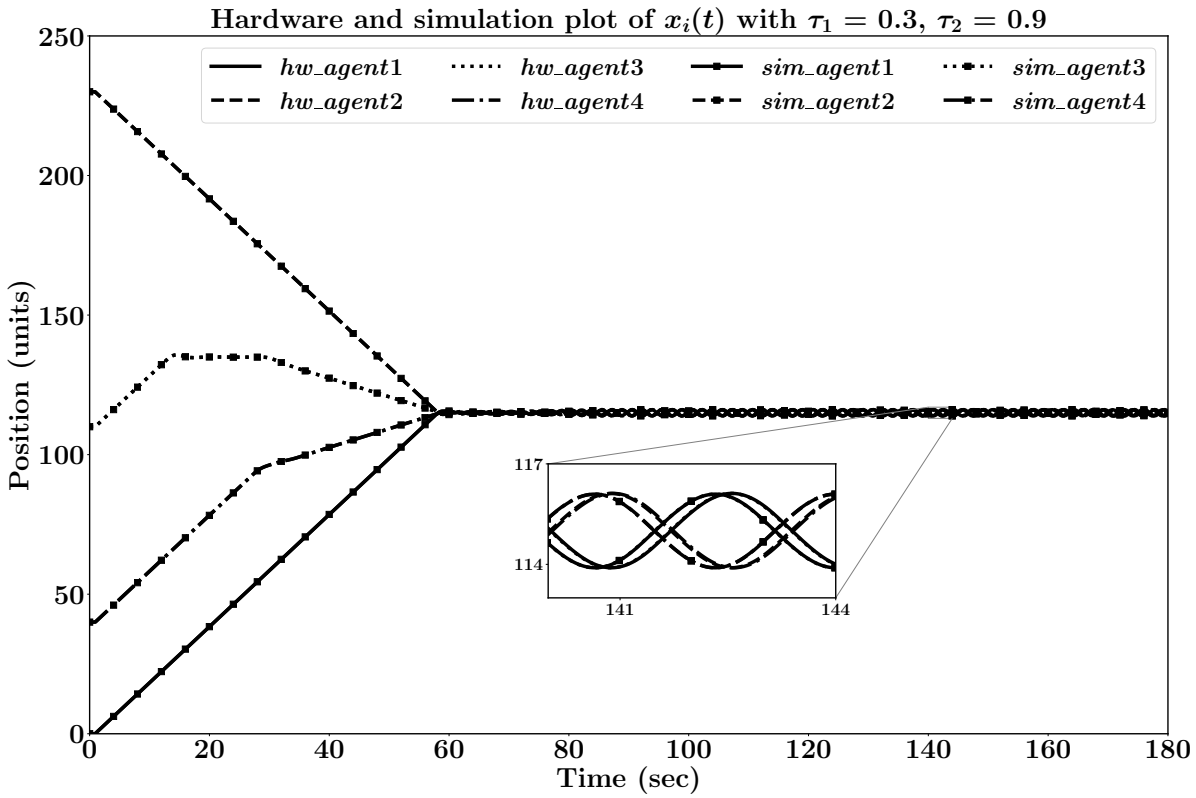


Figure 4.25: Plot with u_{i2} in Eqn. (4.17) for non-uniform time delays.

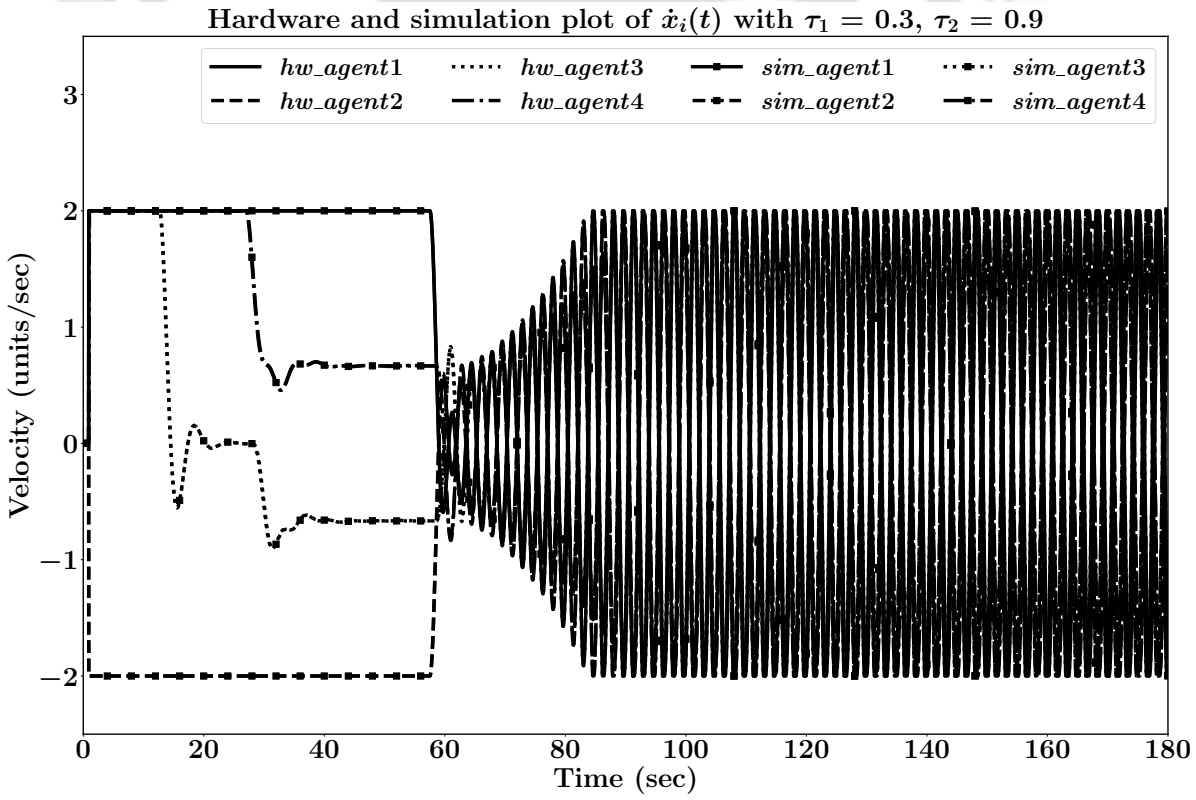


Figure 4.26: Plot with u_{i2} in Eqn. (4.17) for non-uniform time delays.

τ_1 (sec)	τ_2 (sec)	$\overline{\omega}$	$\lvert G_i(\overline{\omega})\rvert$
0.3	0.5	2.879	0.549
0.3	0.82	1.962	0.997
0.3	0.83	1.946	1.011
0.3	0.9	1.842	1.109

Table. 4.2: Values of $\overline{\omega}$ & $\left|G_i(\overline{\omega})\right|$ for different values of τ_1 and τ_2 with u_{i2} in Eqn. (4.17).

system with communication topology given in Fig. 4.3b. The corresponding adjacency matrix $\mathscr{A}$ and $\tilde{\mathscr{A}}$ are as given below,

$$\mathscr{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tilde{\mathscr{A}} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The initial values for the five-agent system are considered as $x_1(0) = 0$, $x_2(0) = 230$, $x_3(0) = 110$, $x_4(0) = 40$, $x_5(0) = 170$ and $\dot{x}_i(0) = 0$, $i \in [1,5]$. From Figs. 4.27 and 4.28, it can be observed that the system converges at a fast rate with $\tau_1 = 0.2$ and $\tau_2 = 0.8$. With $\tau_1 = 0.2$ and $\tau_2 = 1.03$, the system converges at a very slow rate in simulation and further slower in implementation due to added delay from communication links as discussed earlier. The corresponding results are depicted in Figs. 4.29 and 4.30. Limit cycles are exhibited with $\tau_1 = 0.2$ and $\tau_2 \geq 1.04$ as shown in Figs. 4.31 and 4.32. The corresponding numerical values are provided in Table 4.3.

τ_1 (sec)	τ_2 (sec)	$\overline{\omega}$	$\lvert G_i(\overline{\omega})\rvert$
0.2	0.8	1.368	0.849
0.2	1.03	1.249	0.997
0.2	1.04	1.244	1.002
0.2	1.07	1.231	1.019

Table. 4.3: Values of $\overline{\omega}$ & $\left|G_i(\overline{\omega})\right|$ for different values of τ_1 and τ_2 with u_{i1} in Eqn. (4.5).

Using results in Theorems 1 and 2, stable regions with respect to time-delays for both the

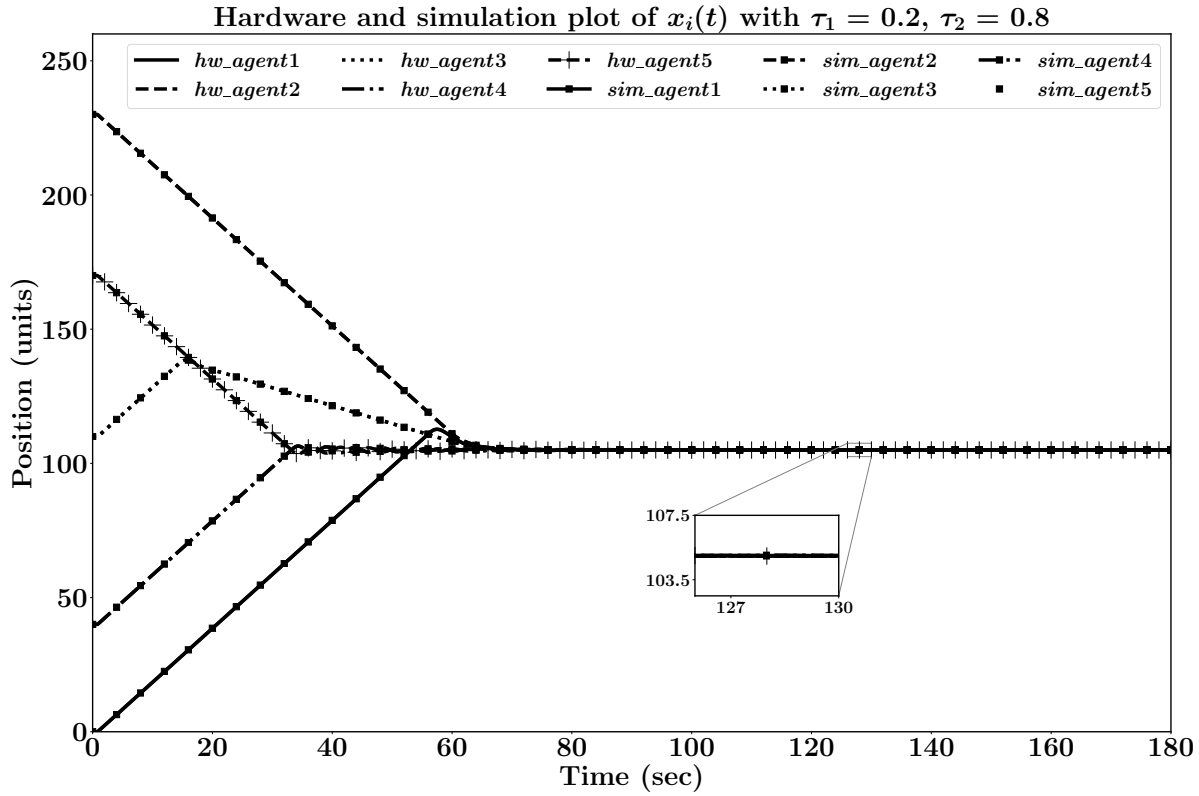


Figure 4.27: Plot with u_{i1} in Eqn. (4.5) for uneven time delays.

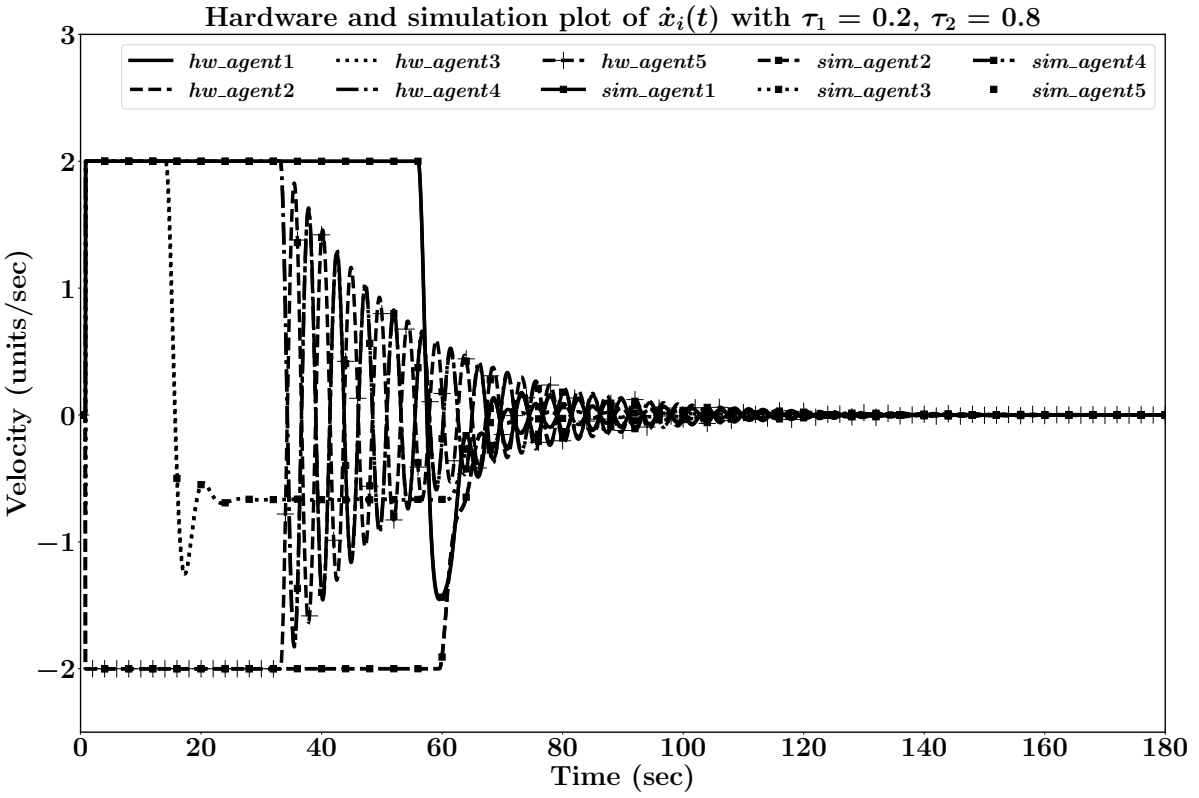


Figure 4.28: Plot with u_{i1} in Eqn. (4.5) for uneven time delays.

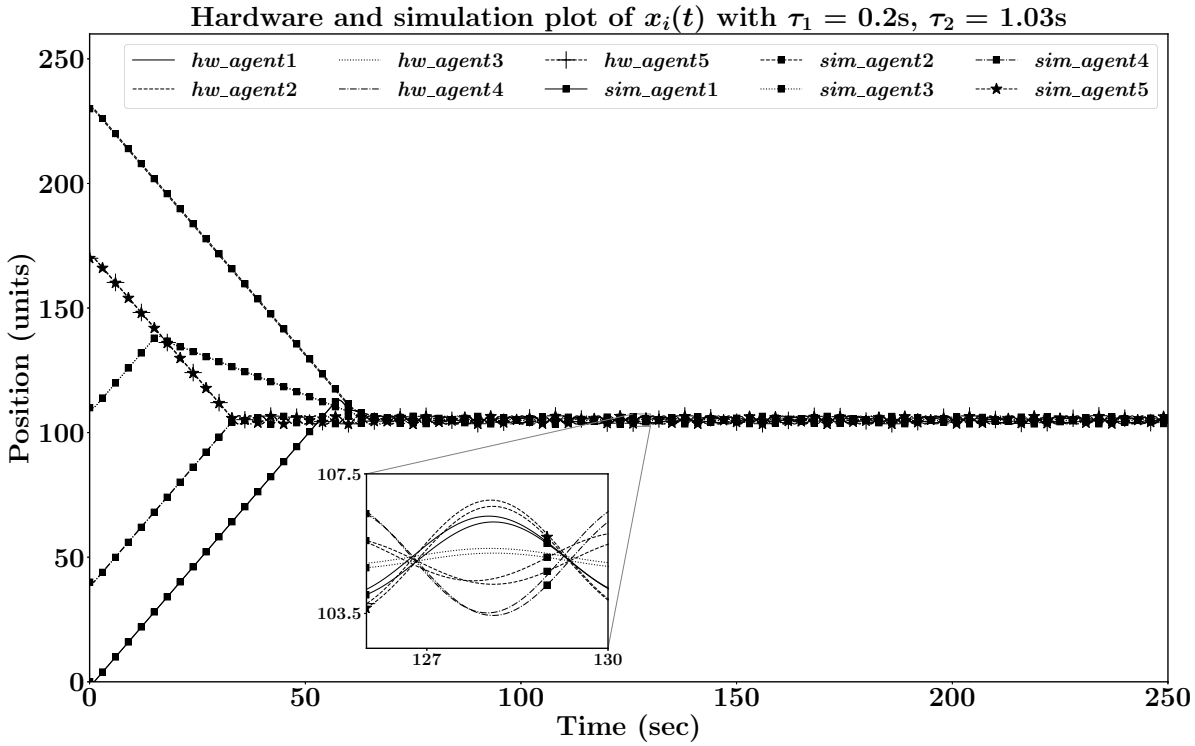


Figure 4.29: Plot with u_{i1} in Eqn. (4.5) for uneven time delays.

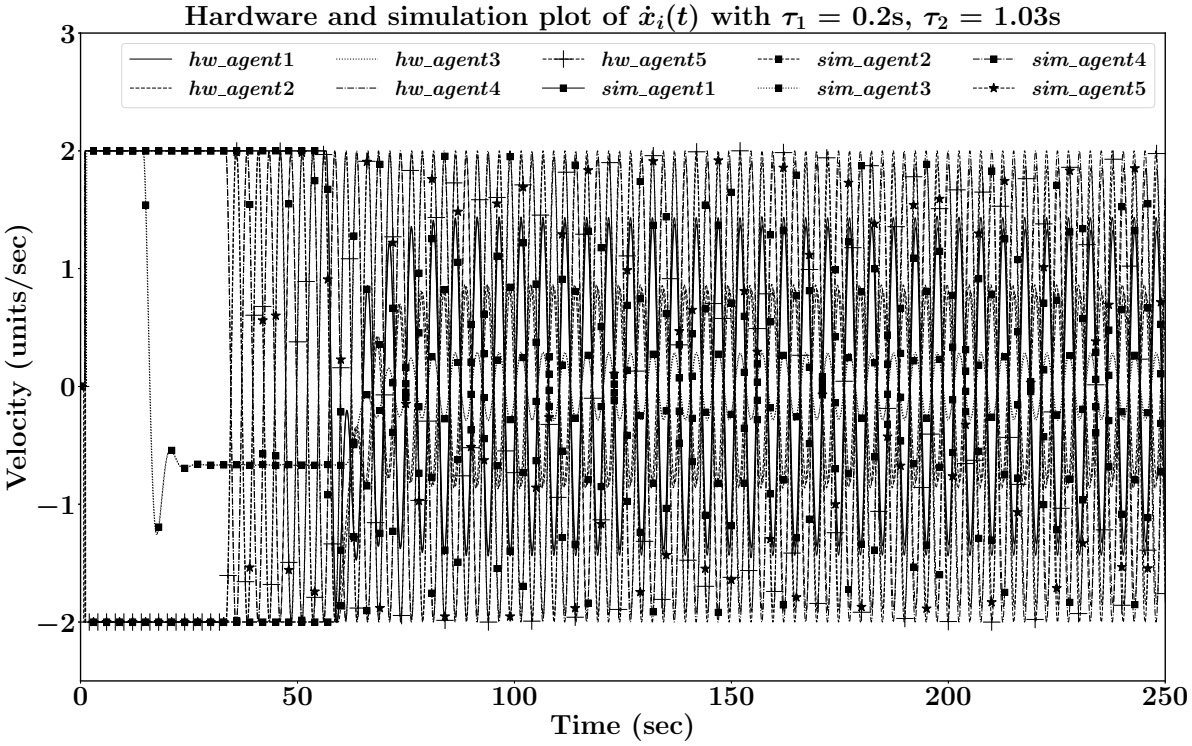


Figure 4.30: Plot with u_{i1} in Eqn. (4.5) for uneven time delays.

topologies as given in Figs. 4.3a and 4.3b are calculated. Figure 4.33 depicts the stable regions calculated theoretically. Few simulations and corresponding implementations are performed to

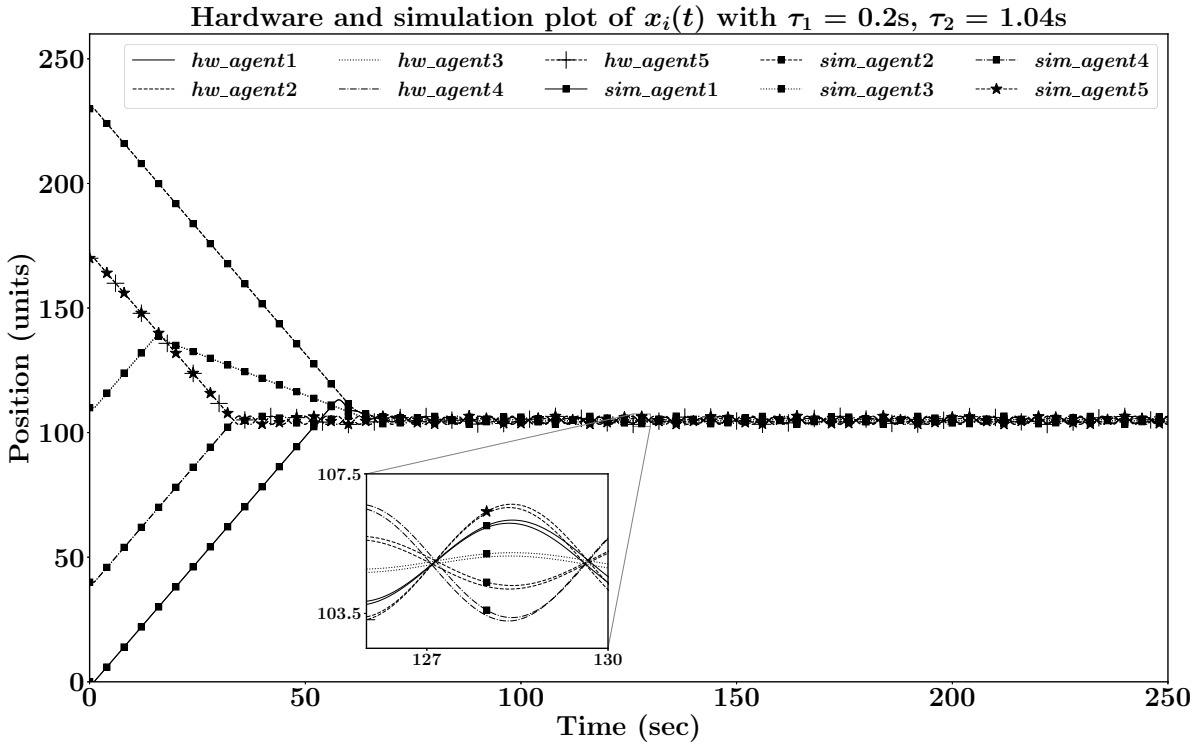


Figure 4.31: Plot with u_{i1} in Eqn. (4.5) for uneven time delays.

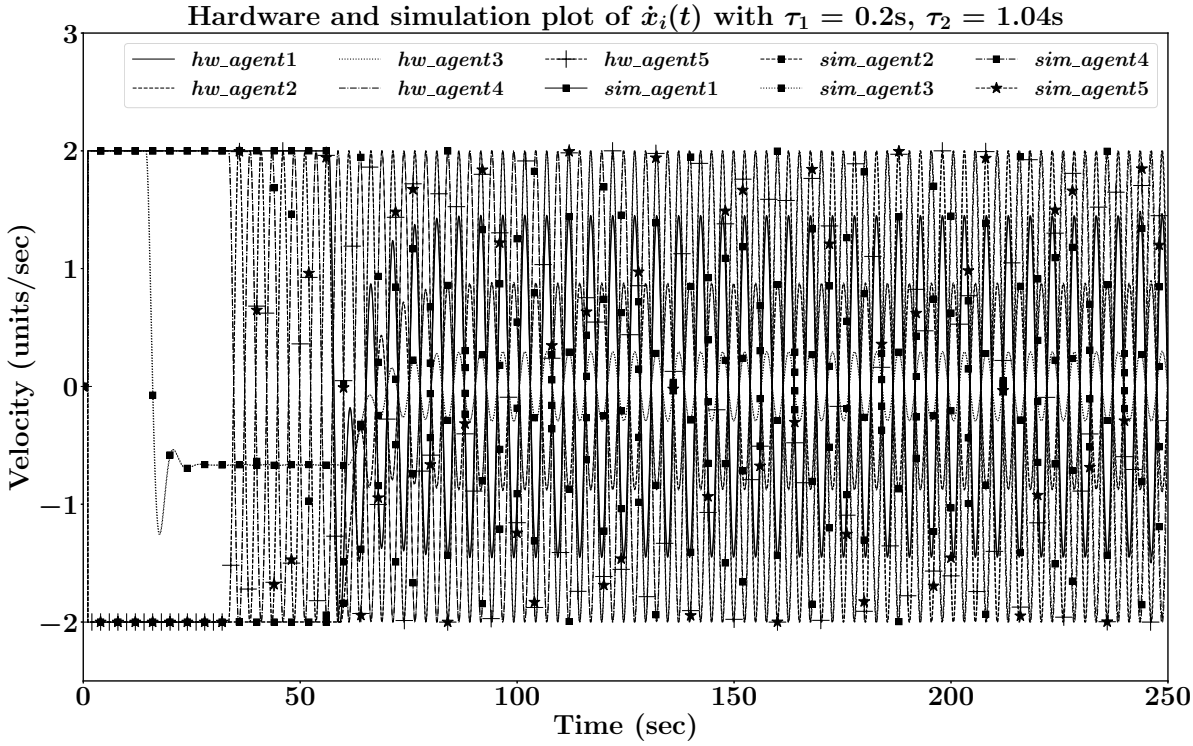


Figure 4.32: Plot with u_{i1} in Eqn. (4.5) for uneven time delays.

validate the theoretical results as discussed earlier.

Form Fig. 4.33, it can be observed that the multi-agent system has better communication

time-delay (τ_2) tolerance with u_{i1} in Eqn. (4.5) for a given input time-delay $\tau_1 < 0.237$. Similarly, the multi-agent system has better communication time-delay (τ_2) tolerance with u_{i2} in Eqn. (4.17) for a given input time-delay $\tau_1 > 0.237$. At $\tau_1 = 0.237$, both u_{i1} and u_{i2} have the communication time-delay tolerance of 0.932, within which the multi-agent system converges.

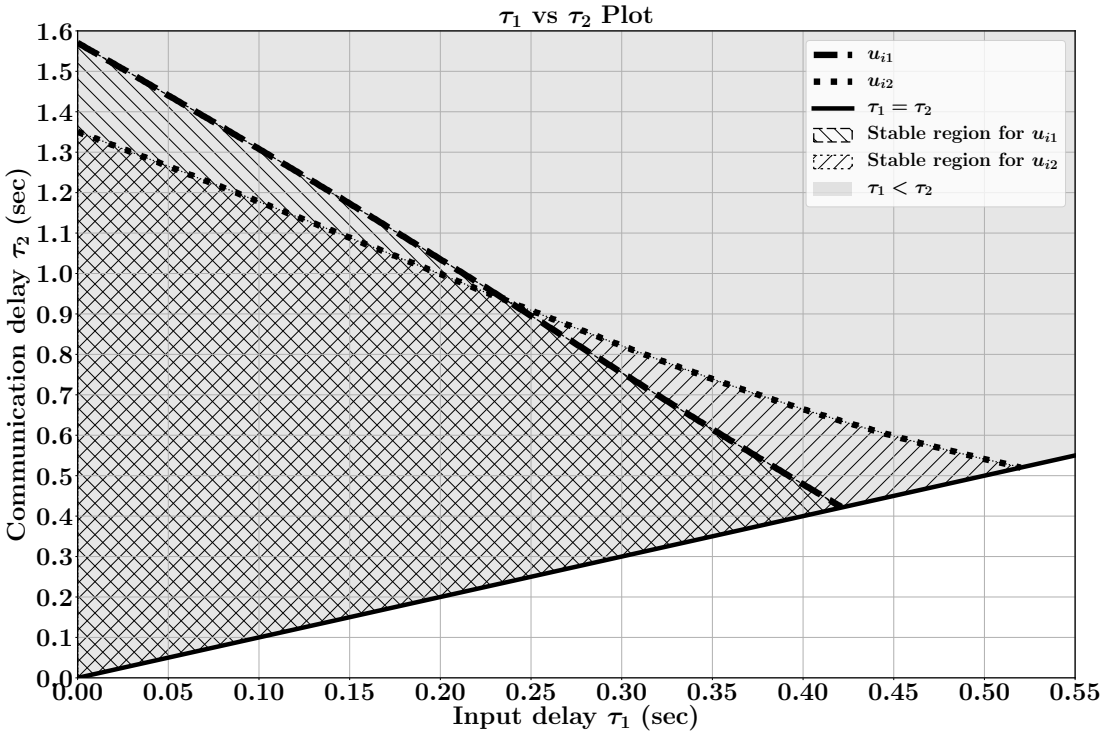


Figure 4.33: Plot of τ_1 vs τ_2 representing stable regions under the condition $\tau_1 \leq \tau_2$.

4.4 Conclusion

The consensus problem of a saturated second order system with uneven input and communication time-delays is studied in this chapter. Stability analysis of the saturated system is performed on the rearranged system with a cascade of nonlinear and linear systems. Explicit expressions are derived to obtain the stable range of time-delays using Nyquist stability criterion. Finally, simulations and hardware implementations are performed to demonstrate the effectiveness of the proposed method. The multiplication factor $\frac{1}{\sum_{j=1}^{n} a_{ij}}$ in the proposed control inputs make the analysis easier, but increases the convergence time. In the next chapter, two more control laws without the multiplication factor are analysed using Lyapunov-Krasovskii approach.

CHAPTER 5

MULTI-AGENT SYSTEM ANALYSIS: LYAPUNOV APPROACH

5.1 Introduction

In the preceding chapter, explicit expressions are obtained using frequency domain analysis. The derived expressions are applicable to control inputs with scaling factor $\frac{1}{\sum_{j=1}^{n} a_{ij}}$. If an agent is receiving information from more than one agent, the multiplication factor scales down the control input. The affect of such reduction in the control input is reflected in the increase of convergence time. In certain applications, convergence time is a crucial factor. To overcome the issue, two control laws without the scaling factor are used to study the consensus problem of multi-agent system.

A similar approach as discussed in the preceding chapter is followed in rearranging the system into linear and nonlinear blocks. Stability of the linear system determines the existence of limit cycles in the output. Stability analysis of the linear system with all the four control laws is performed with the help of Lyapunov-Krasovskii functions [109–112]. Lyapunov approach is more conservative in obtaining the time-delay tolerance range when compared to frequency domain approach. The time-delay tolerances vary among all the control laws. A comparative analysis is also presented to highlight the advantages and disadvantages of both frequency domain approach and Lyapunov approach. Simulations and corresponding hardware implementations are performed to demonstrate the effectiveness of theoretical results.

A multi-agent system with general nonlinear input has received very little attention in the

existing literature [24, 31, 106]. A multi-agent system with double integrator dynamics and general nonlinear input without time-delay is studied. Few conditions are derived to prove that a general nonlinear odd function $g(x)$ symmetric to origin and $g(x) > 0$, $\forall x > 0$ is able to solve the consensus problem. The theoretical results are validated through simulations and hardware implementations.

5.2 System model and analysis

Consider a multi-agent network of homogeneous second order agents with i^{th} agent dynamics given in Eqn. (5.1),

$$\begin{aligned} \dot{x}_i(t) &= sat(\hat{v}_i(t)) \\ \dot{\hat{v}}_i(t) &= u_i(t) \end{aligned} \tag{5.1}$$

The system is similar to the one considered in the previous chapter. The position of an agent is represented by $x_i(t)$ and the velocity by $\dot{x}_i(t)$. Various control protocols used in the analysis are given in Eqns. (5.5) to (5.8). It is assumed that $\forall t \in (-\infty, 0]$, $x_i(t) = x(0)$ and $\hat{v}_i(t) = 0$. Saturation nonlinearity used in the system is defined in Eqn. (5.2) with $\pm K$ as bounds.

$$sat(\alpha) = \begin{cases} -K, & \text{if } \alpha \leq -K \\ \alpha, & \text{if } -K < \alpha < K \\ K, & \text{if } \alpha \geq K \end{cases} \tag{5.2}$$

The i^{th} agent dynamics are depicted using a block diagram given in Fig. 5.1. As discussed

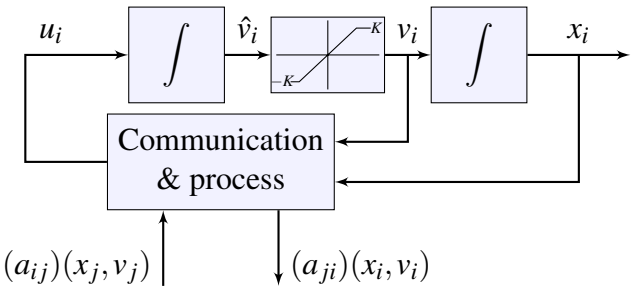


Figure 5.1: Block diagram of i^{th} agent.

earlier in Chapter-4, the system can be approximately transformed as shown in Fig. 5.2. Since a single-valued nonlinearity is considered, its approximate describing function for the saturation is given in Eqn. (5.3) [108],

$$N(A) = \frac{2}{\pi}\left[\arcsin\left(\frac{K}{A}\right) + \frac{K}{A}\sqrt{1-\frac{K^2}{A^2}}\right] \tag{5.3}$$

where, the limit cycles' amplitude is represented by A.

The describing function is real valued and $-1/N(A) \in [-1,\infty)$, it can be estimated that the limit cycles are stable when the transfer function of linear element in Fig. 5.2 encircles $(-1,0)$ in a complex plane. In other words, limit cycles are exhibited when the linear element is unstable in the multi-agent system. Stability analysis of the linear element is performed using Lyapunov-Krasovskii approach in section 5.2.1.

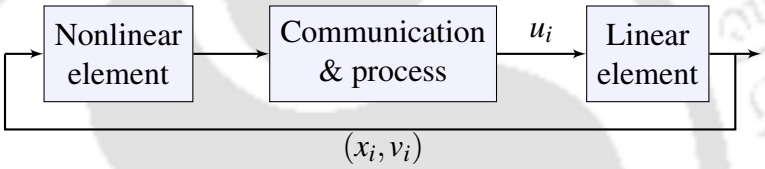


Figure 5.2: Rearranged block diagram of i^{th} agent.

The approximate linear element given in Fig. 5.2 is represented by Eqn. (5.4),

$$\begin{aligned} \dot{x}_i(t) &= v_i(t) \\ \dot{v}_i(t) &= u_i(t) \end{aligned} \tag{5.4}$$

Various control laws considered from the literature [77,83,113] for analysis are given in Eqns. (5.5) to (5.8),

$$u_{i1}(t) = -v_i(t-\tau_1) + \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} \left[a_{ij}\left(x_j(t-\tau_2) - x_i(t-\tau_1)\right)\right] \tag{5.5}$$

$$u_{i2} = \frac{1}{\sum_{j=1}^{n} a_{ij}} \sum_{j=1}^{n} \left[a_{ij}\left(v_j(t-\tau_2) - v_i(t-\tau_1)\right) + a_{ij}\left(x_j(t-\tau_2) - x_i(t-\tau_1)\right)\right] \tag{5.6}$$

$$u_{i3}(t) = -v_i(t-\tau_1) + \sum_{j=1}^{n} \left[a_{ij}\left(x_j(t-\tau_2) - x_i(t-\tau_1)\right)\right] \tag{5.7}$$

$$u_{i4} = \sum_{j=1}^{n} \left[a_{ij}\left(v_j(t-\tau_2) - v_i(t-\tau_1)\right) + a_{ij}\left(x_j(t-\tau_2) - x_i(t-\tau_1)\right)\right] \tag{5.8}$$

where τ_1 and τ_2 represent input and communication time-delays respectively. With any of the

control laws in Eqns. (5.5) to (5.8), consensus is said to be reached if $(x_i(t) - x_j(t)) \to 0$ and $(\dot{x}_i(t) - \dot{x}_j(t)) \to 0 \; \forall \{i, j\} \in [1, \, n]$. Control laws in Eqns. (5.5) and (5.6) generate lesser magnitude of control input u_i which result in slightly larger convergence time compared to the ones in Eqns. (5.7) and (5.8). The averaging term $\frac{1}{\sum_{j=1}^{n} a_{ij}}$ in control laws given by Eqns. (5.5) and (5.6) gives better time-delay tolerance due to smaller Fiedler eigenvalue compared to control laws in Eqns. (5.7) and (5.8) at the expense of convergence time. With control laws in Eqns. (5.5) and (5.7), the state $\dot{x}_i(t) \to 0$ when the consensus is achieved since they do not consider difference in velocity. State $\dot{x}_i(t) \to 0$ is not guaranteed with control laws in Eqns. (5.6) and (5.8).

5.2.1 Lyapunov-Krasovskii approach

Consider the linear element represented in Eqns. (5.4) to (5.8), which can be represented as given in Eqn. (5.9).

$$\dot{X}(t) = \mathscr{A}_0 X(t) + \mathscr{A}_1 X(t - \tau_1) + \mathscr{A}_2 X(t - \tau_2) \tag{5.9}$$

Where $\mathscr{A}_0$, $\mathscr{A}_1$ and $\mathscr{A}_2$ are as given in Eqns. (5.10) to (5.13).

For u_{i1} given in Eqn. (5.5),

$$\mathscr{A}_0 = \begin{bmatrix} 0_{n\times n} & I_n \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}; \mathscr{A}_1 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ -I_n & -I_n \end{bmatrix}; \mathscr{A}_2 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \widetilde{\mathscr{A}} & 0_{n\times n} \end{bmatrix} \tag{5.10}$$

For u_{i2} given in Eqn. (5.6),

$$\mathscr{A}_0 = \begin{bmatrix} 0_{n\times n} & I_n \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}; \mathscr{A}_1 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ -I_n & -I_n \end{bmatrix}; \mathscr{A}_2 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \widetilde{\mathscr{A}} & \widetilde{\mathscr{A}} \end{bmatrix} \tag{5.11}$$

For u_{i3} given in Eqn. (5.7),

$$\mathscr{A}_0 = \begin{bmatrix} 0_{n\times n} & I_n \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}; \mathscr{A}_1 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ -D & -I_n \end{bmatrix}; \mathscr{A}_2 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \mathscr{A} & 0_{n\times n} \end{bmatrix} \tag{5.12}$$

For u_{i4} given in Eqn. (5.8),

$$\mathscr{A}_0 = \begin{bmatrix} 0_{n\times n} & I_n \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix}; \mathscr{A}_1 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ -D & -D \end{bmatrix}; \mathscr{A}_2 = \begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \mathscr{A} & \mathscr{A} \end{bmatrix} \tag{5.13}$$

Some definitions and lemmas analogous to the ones in [114] are given below, these definitions are useful in the further analysis.

Definition 5. *Balanced graph*: A graph is said to be balanced if in-degree equals to out-degree for all vertices in the graph, $d_{in}(v_i) = d_{out}(v_i), \forall i \in [1,n]$ [103].

Definition 6. *k-regular graph*: It is a balanced graph with all the vertices having in-degree and out-degree equal to k, $d_{in}(v_i) = d_{out}(v_i) = k, \forall i \in [1,n]$ [103].

Control laws given in Eqns. (5.5) and (5.6) make the multi-agent system behave like a system connected by 1-regular graph.

Lemma 3. *Consider* $\Phi_{01} = \frac{1}{n}\begin{bmatrix} 1_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 1_{n\times n} \end{bmatrix}$ *and* $\mathscr{E} = I_{2n} - \Phi_{01}$, *then the following statements hold true:*

1. *A multi-agent system with k-regular graph communication topology and with inputs in Eqns.* (5.5) *to* (5.8) *produces balanced matrices* $\mathscr{E}(\mathscr{A}_0 + \mathscr{A}_1 + \mathscr{A}_2)$, $\mathscr{E}\mathscr{A}_0$, $\mathscr{E}\mathscr{A}_1$, $\mathscr{E}\mathscr{A}_2$ *and* $\mathscr{E}(\mathscr{A}_1 + \mathscr{A}_2)$ *with maximum rank* $2n-2$ *and eigenvalues* 0 *of multiplicity atleast two.*

2. *A multi-agent system with spanning tree communication topology and with inputs in Eqns.* (5.5) *and* (5.6) *produces balanced matrices* $\mathscr{E}(\mathscr{A}_0 + \mathscr{A}_1 + \mathscr{A}_2)$, $\mathscr{E}\mathscr{A}_0$, $\mathscr{E}\mathscr{A}_1$, $\mathscr{E}\mathscr{A}_2$ *and* $\mathscr{E}(\mathscr{A}_1 + \mathscr{A}_2)$ *with maximum rank* $2n-2$ *and eigenvalues* 0 *of multiplicity atleast two.*

Definition 7. *Balanced matrix*: A square matrix $\mathscr{M} \in R^{n\times n}$ is said to be balanced iff $\mathscr{M}\mathbf{1}_n^T = 0$ and $\mathbf{1}_n\mathscr{M} = 0$.

Lemma 4. *Consider* $\Phi_{01} = \frac{1}{n}\begin{bmatrix} 1_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 1_{n\times n} \end{bmatrix}$ *and* $\mathscr{E} = I_{2n} - \Phi_{01}$, *then the following statements hold true for k-regular graph with inputs in Eqns.* (5.5) *to* (5.8) *and for spanning tree graph with inputs in Eqns.* (5.5) *and* (5.6)*:*

1. $\mathscr{E}(\mathscr{A}_0 + \mathscr{A}_1 + \mathscr{A}_2)$ *is a balanced matrix with rank* $2n-2$ *and eigenvalues* 0 *of multiplicity* 2.

2. *Matrices* $\mathscr{E}\mathscr{A}_0$, $\mathscr{E}\mathscr{A}_1$, $\mathscr{E}\mathscr{A}_2$ *and* $\mathscr{E}(\mathscr{A}_1+\mathscr{A}_2)$ *are all balanced with eigenvalues* 0 *of multiplicity atleast* 2.

3. *There is a matrix* U, *an orthogonal matrix of eigenvectors of* $\mathscr{E}$ *which satisfies,*
$$U^T\mathscr{E}U = \begin{bmatrix} \widetilde{\mathscr{E}}_{(2n-2)\times 2} & 0_{(2n-2)\times 2} \\ 0_{2\times(2n-2)} & 0_{2\times 2} \end{bmatrix}$$

4. *Let* $\mathscr{E}\mathscr{A}_0 = \mathscr{F}_0$, $\mathscr{E}\mathscr{A}_1 = \mathscr{F}_1$ *and* $\mathscr{E}\mathscr{A}_2 = \mathscr{F}_2$. $\mathscr{E}$, $\mathscr{F}_0$, $\mathscr{F}_1$ *and* $\mathscr{F}_2$ *have maximum rank* $2n-2$ *and with zero row sums, then,*
$$U^T\mathscr{F}_iU = \begin{bmatrix} \widetilde{\mathscr{F}_i}_{(2n-2)\times(2n-2)} & 0_{(2n-2)\times 2} \\ 0_{2\times(2n-2)} & 0_{2\times 2} \end{bmatrix},\ i\in[0,2].$$

5. *Also, for cases of* $\tau_1=\tau_2>0$ *and* $\tau_1=\tau_2=0$,
$$U^T\mathscr{E}(\mathscr{A}_1+\mathscr{A}_2)U = \begin{bmatrix} (\widetilde{\mathscr{F}_1}+\widetilde{\mathscr{F}_2})_{(2n-2)\times(2n-2)} & 0_{(2n-2)\times 2} \\ 0_{2\times(2n-2)} & 0_{2\times 2} \end{bmatrix}$$
$$U^T\mathscr{E}(\mathscr{A}_0+\mathscr{A}_1+\mathscr{A}_2)U = \begin{bmatrix} (\widetilde{\mathscr{F}_0}+\widetilde{\mathscr{F}_1}+\widetilde{\mathscr{F}_2}) & 0_{(2n-2)\times 2} \\ 0_{2\times(2n-2)} & 0_{2\times 2} \end{bmatrix}$$

Let the difference in position and velocity among the agents be assumed as error Ψ, each element of Ψ is given by Eqn. (5.14)

$$\Psi_i = \begin{cases} \frac{1}{n}\sum_{j=1}^{n}(X_i - X_j) & \forall i\in[1,\,n] \\ \frac{1}{n}\sum_{j=n+1}^{2n}(X_i - X_j) & \forall i\in[n+1,\,2n] \end{cases} \tag{5.14}$$

From the assumption in lemma 3,

$$\Psi = \mathscr{E}X \tag{5.15}$$

Lemma 5. *When error* $\Psi\to 0$, *then* $x_i\to x_j$ *and* $v_i\to v_j$. *Similarly, when* $x_i\to x_j$ *and* $v_i\to v_j$, *then* $\Psi\to 0$.

Proof. Consider a matrices,

$$\gamma_{n\times n} = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & -1 \\ -1 & 0 & \dots & 0 & 1 \end{bmatrix} \tag{5.16}$$

$$\Gamma_{2n\times 2n} = \begin{bmatrix} \gamma_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & \gamma_{n\times n} \end{bmatrix} \tag{5.17}$$

Multiplying with Γ on both sides of Eqn. (5.15),

$$\begin{bmatrix} \Psi_1 - \Psi_2 \\ \Psi_2 - \Psi_3 \\ . \\ . \\ \Psi_n - \Psi_1 \\ \Psi_{n+1} - \Psi_{n+2} \\ \Psi_{n+2} - \Psi_{n+3} \\ . \\ . \\ \Psi_{2n} - \Psi_{n+1} \end{bmatrix} = \begin{bmatrix} X_1 - X_2 \\ X_2 - X_3 \\ . \\ . \\ X_n - X_1 \\ X_{n+1} - X_{n+2} \\ X_{n+2} - X_{n+3} \\ . \\ . \\ X_{2n} - X_{n+1} \end{bmatrix} \tag{5.18}$$

When $\Psi \to 0$, left side of Eqn. (5.18) becomes $0_{2n\times 1}$. Which implies, $X_i \to X_j$, $\forall\{i,j\} \in [1,n]$ and $X_i \to X_j$, $\forall\{i,j\} \in [n+1,2n]$.

From Eqn. (5.14), when $x_i \to x_j$ and $v_i \to v_j$, then $\Psi \to 0$. □

Let us consider a 3-agent$(n=3)$ system, then $\mathscr{E} = I_{2n} - \frac{1}{n}\begin{bmatrix} 1_{n\times n} & 0_{n\times n} \\ 0_{n\times n} & 1_{n\times n} \end{bmatrix}$

$$\Psi = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ 0 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix} X, \text{ give rise to following equations,}$$

$$\Psi_1 = \frac{2}{3}X_1 - \frac{1}{3}X_2 - \frac{1}{3}X_3 \tag{5.19}$$

$$\Psi_2 = -\frac{1}{3}X_1 + \frac{2}{3}X_2 - \frac{1}{3}X_3 \tag{5.20}$$

$$\Psi_3 = -\frac{1}{3}X_1 - \frac{1}{3}X_2 + \frac{2}{3}X_3 \tag{5.21}$$

$$\Psi_4 = \frac{2}{3}X_4 - \frac{1}{3}X_5 - \frac{1}{3}X_6 \tag{5.22}$$

$$\Psi_2 = -\frac{1}{3}X_4 + \frac{2}{3}X_5 - \frac{1}{3}X_6 \tag{5.23}$$

$$\Psi_3 = -\frac{1}{3}X_4 - \frac{1}{3}X_5 + \frac{2}{3}X_6 \tag{5.24}$$

By taking difference of appropriate equations,

$$\Psi_1 - \Psi_2 = X_1 - X_2 \tag{5.25}$$

$$\Psi_2 - \Psi_3 = X_2 - X_3 \tag{5.26}$$

$$\Psi_3 - \Psi_1 = X_3 - X_1 \tag{5.27}$$

$$\Psi_4 - \Psi_5 = X_4 - X_5 \tag{5.28}$$

$$\Psi_5 - \Psi_6 = X_5 - X_6 \tag{5.29}$$

$$\Psi_6 - \Psi_4 = X_6 - X_4 \tag{5.30}$$

From above equations, when $\Psi \to 0$, $X_i \to X_j$, $\forall\{i,j\} \in [1,3]$ and $\forall\{i,j\} \in [4,6]$. The procedure can be extrapolated to any number of agents.

A control input is said to have solved the consensus problem in a globally asymptotic manner when $x_i \to x_j$ and $v_i \to v_j$, in other words, $\Psi \to 0$. Stability of linear element with the control inputs estimates the existence of limit cycles in the system. When the linear element is stable, the system reaches consensus and limit cycles are not exhibited. Similarly, when the linear

element is unstable, the system exhibit limit cycles.

Theorem 3. *Consider the linear element in Eqn.* (5.4) *with time-delays* $(\tau_1, \tau_2) \geq 0, \tau_1 \leq \tau_2$. *The control inputs for a k-regular graph given in Eqns.* (5.5) *to* (5.8) *and the control inputs for a spanning tree graph given in Eqns.* (5.5) *and* (5.6) *globally asymptotically solve consensus problem, if there exist matrices* $\widetilde{P} > 0$, $\widetilde{Q}_1 > 0$, $\widetilde{Q}_2 > 0$, $\widetilde{Z}_i > 0$, $\forall i \in [1,3]$, $\widetilde{\mathscr{F}_i}$ *from lemma 4* $\forall i \in [1,3]$ *and arbitrary matrices* $\{\widetilde{H}_{ij}, \widetilde{H}_{ij}^T, \widetilde{I}_{ij}, \widetilde{I}_{ij}^T, \widetilde{J}_{ij}, \widetilde{J}_{ij}^T\}$ $\forall i \in [1,3]$ $\forall j \in [1,4]$ *of size* $(2n-2) \times (2n-2)$ *such that,*

$$\begin{bmatrix} \widetilde{G}_{11} & \widetilde{G}_{12} & \widetilde{G}_{13} \\ \widetilde{G}_{12}^T & \widetilde{G}_{22} & \widetilde{G}_{23} \\ \widetilde{G}_{13}^T & \widetilde{G}_{23}^T & \widetilde{G}_{33} \end{bmatrix} \prec 0 \tag{5.31}$$

$$\begin{bmatrix} \widetilde{H}_{11} & \widetilde{H}_{12} & \widetilde{H}_{13} & \widetilde{H}_{14} \\ \widetilde{H}_{12}^T & \widetilde{H}_{22} & \widetilde{H}_{23} & \widetilde{H}_{24} \\ \widetilde{H}_{13}^T & \widetilde{H}_{23}^T & \widetilde{H}_{33} & \widetilde{H}_{34} \\ \widetilde{H}_{14}^T & \widetilde{H}_{24}^T & \widetilde{H}_{34}^T & \widetilde{Z}_1 \end{bmatrix} \succcurlyeq 0 \tag{5.32}$$

$$\begin{bmatrix} \widetilde{I}_{11} & \widetilde{I}_{12} & \widetilde{I}_{13} & \widetilde{I}_{14} \\ \widetilde{I}_{12}^T & \widetilde{I}_{22} & \widetilde{I}_{23} & \widetilde{I}_{24} \\ \widetilde{I}_{13}^T & \widetilde{I}_{23}^T & \widetilde{I}_{33} & \widetilde{I}_{34} \\ \widetilde{I}_{14}^T & \widetilde{I}_{24}^T & \widetilde{I}_{34}^T & \widetilde{Z}_2 \end{bmatrix} \succcurlyeq 0 \tag{5.33}$$

$$\begin{bmatrix} \widetilde{J}_{11} & \widetilde{J}_{12} & \widetilde{J}_{13} & \widetilde{J}_{14} \\ \widetilde{J}_{12}^T & \widetilde{J}_{22} & \widetilde{J}_{23} & \widetilde{J}_{24} \\ \widetilde{J}_{13}^T & \widetilde{J}_{23}^T & \widetilde{J}_{33} & \widetilde{J}_{34} \\ \widetilde{J}_{14}^T & \widetilde{J}_{24}^T & \widetilde{J}_{34}^T & \widetilde{Z}_3 \end{bmatrix} \succcurlyeq 0 \tag{5.34}$$

where,

$$\begin{aligned} \widetilde{G}_{11} =& \widetilde{\mathscr{F}_0}^T \widetilde{P} \widetilde{\mathscr{E}} + \widetilde{\mathscr{E}}^T \widetilde{P} \widetilde{\mathscr{F}_0} + \widetilde{\mathscr{E}}^T \widetilde{Q}_1 \widetilde{\mathscr{E}} + \widetilde{\mathscr{E}}^T \widetilde{Q}_2 \widetilde{\mathscr{E}} + \widetilde{\mathscr{F}_0}^T \widetilde{\Xi} \widetilde{\mathscr{F}_0} + \tau_1 \widetilde{\mathscr{E}}^T \widetilde{H}_{11} \widetilde{\mathscr{E}} + \widetilde{\mathscr{E}}^T \widetilde{H}_{14} \widetilde{\mathscr{E}} \\ &+ \widetilde{\mathscr{E}}^T \widetilde{H}_{14}^T \widetilde{\mathscr{E}} + \tau_2 \widetilde{\mathscr{E}}^T \widetilde{I}_{11} \widetilde{\mathscr{E}} + \widetilde{\mathscr{E}}^T \widetilde{I}_{14} \widetilde{\mathscr{E}} + \widetilde{\mathscr{E}}^T \widetilde{I}_{14}^T \widetilde{\mathscr{E}} + (\tau_2 - \tau_1) \widetilde{\mathscr{E}}^T \widetilde{J}_{11} \widetilde{\mathscr{E}} \\ \widetilde{G}_{12} =& \widetilde{\mathscr{E}}^T P \widetilde{\mathscr{F}_1} + \widetilde{\mathscr{F}_0}^T \widetilde{\Xi} \widetilde{\mathscr{F}_1} + \tau_1 \widetilde{\mathscr{E}}^T \widetilde{H}_{12} \widetilde{\mathscr{E}} - \widetilde{\mathscr{E}}^T \widetilde{H}_{14} \widetilde{\mathscr{E}} + \widetilde{\mathscr{E}}^T \widetilde{H}_{24}^T \widetilde{\mathscr{E}} + \tau_2 \widetilde{\mathscr{E}}^T \widetilde{I}_{12} \widetilde{\mathscr{E}} \end{aligned} \tag{5.35}$$

$$+\widetilde{\mathscr{E}}^T\widetilde{I}_{24}^T\widetilde{\mathscr{E}}+(\tau_2-\tau_1)\,\widetilde{\mathscr{E}}^T\widetilde{J}_{12}\widetilde{\mathscr{E}}+\widetilde{\mathscr{E}}^T\widetilde{J}_{14}\widetilde{\mathscr{E}} \tag{5.36}$$

$$\widetilde{G}_{13}=\widetilde{\mathscr{E}}^T\widetilde{P}\widetilde{\mathscr{F}}_2+\widetilde{\mathscr{F}}_0^T\widetilde{\Xi}\widetilde{\mathscr{F}}_2+\tau_1\widetilde{\mathscr{E}}^T\widetilde{H}_{13}\widetilde{\mathscr{E}}+\widetilde{\mathscr{E}}^T\widetilde{H}_{34}^T\widetilde{\mathscr{E}}+\widetilde{\mathscr{E}}^T\widetilde{I}_{34}^T\widetilde{\mathscr{E}}+\tau_2\widetilde{\mathscr{E}}^T\widetilde{I}_{13}\widetilde{\mathscr{E}}-$$
$$\widetilde{\mathscr{E}}^T\widetilde{I}_{14}\widetilde{\mathscr{E}}+(\tau_2-\tau_1)\,\widetilde{\mathscr{E}}^T\widetilde{J}_{13}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{J}_{14}\widetilde{\mathscr{E}} \tag{5.37}$$

$$\widetilde{G}_{22}=-\widetilde{\mathscr{E}}^T\widetilde{Q}_1\widetilde{\mathscr{E}}+\widetilde{\mathscr{F}}_1^T\widetilde{\Xi}\widetilde{\mathscr{F}}_1+\tau_1\widetilde{\mathscr{E}}^T\widetilde{H}_{22}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{H}_{24}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{H}_{24}^T\widetilde{\mathscr{E}}+\tau_2\widetilde{\mathscr{E}}^T\widetilde{I}_{22}\widetilde{\mathscr{E}}+$$
$$(\tau_2-\tau_1)\,\widetilde{\mathscr{E}}^T\widetilde{J}_{22}\widetilde{\mathscr{E}}+\widetilde{\mathscr{E}}^T\widetilde{J}_{24}\widetilde{\mathscr{E}}+\widetilde{\mathscr{E}}^T\widetilde{J}_{24}^T\widetilde{\mathscr{E}} \tag{5.38}$$

$$\widetilde{G}_{23}=\widetilde{\mathscr{F}}_1^T\widetilde{\Xi}\widetilde{\mathscr{F}}_2+\tau_1\widetilde{\mathscr{E}}^T\widetilde{H}_{23}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{H}_{34}^T\widetilde{\mathscr{E}}+\tau_2\widetilde{\mathscr{E}}^T\widetilde{I}_{23}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{I}_{24}\widetilde{\mathscr{E}}+$$
$$(\tau_2-\tau_1)\,\widetilde{\mathscr{E}}^T\widetilde{J}_{23}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{J}_{24}\widetilde{\mathscr{E}}+\widetilde{\mathscr{E}}^T\widetilde{J}_{34}^T\widetilde{\mathscr{E}} \tag{5.39}$$

$$\widetilde{G}_{33}=-\widetilde{\mathscr{E}}^T\widetilde{Q}_2\widetilde{\mathscr{E}}+\widetilde{\mathscr{F}}_2^T\widetilde{\Xi}\widetilde{\mathscr{F}}_2+\tau_1\widetilde{\mathscr{E}}^T\widetilde{H}_{33}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{I}_{34}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{I}_{34}^T\widetilde{\mathscr{E}}+\tau_2\widetilde{\mathscr{E}}^T\widetilde{I}_{33}\widetilde{\mathscr{E}}+$$
$$(\tau_2-\tau_1)\,\widetilde{\mathscr{E}}^T\widetilde{J}_{33}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{J}_{34}\widetilde{\mathscr{E}}-\widetilde{\mathscr{E}}^T\widetilde{J}_{34}^T\widetilde{\mathscr{E}} \tag{5.40}$$

$$\widetilde{\Xi}=\tau_1\widetilde{Z}_1+\tau_2\widetilde{Z}_2+(\tau_2-\tau_1)\,\widetilde{Z}_3 \tag{5.41}$$

Proof. Let P, Q_1, Q_2, Z_i $i\in[1,3]$ be balanced positive semi-definite matrices of rank $2n-2$ and $\Psi=\mathscr{E}X$ using $\mathscr{E}$ from lemma 3.

The Lyapunov-Krasovskii functional is assumed as,

$$V(\Psi(t))=\Psi^T(t)\,P\Psi(t)+\int_{t-\tau_1}^{t}\Psi^T(s)\,Q_1\Psi(s)\,ds+\int_{t-\tau_2}^{t}\Psi^T(s)\,Q_2\Psi(s)\,ds$$
$$+\int_{-\tau_1}^{0}\int_{t+\theta}^{t}\dot{\Psi}^T(s)\,Z_1\dot{\Psi}(s)\,ds d\theta+\int_{-\tau_2}^{0}\int_{t+\theta}^{t}\dot{\Psi}^T(s)\,Z_2\dot{\Psi}(s)\,ds d\theta$$
$$+\int_{-\tau_2}^{-\tau_1}\int_{t+\theta}^{t}\dot{\Psi}^T(s)\,Z_3\dot{\Psi}(s)\,ds d\theta \tag{5.42}$$

$$\dot{V}(\Psi(t))=\dot{\Psi}^T(t)\,P\Psi(t)+\Psi^T(t)\,P\dot{\Psi}(t)+\Psi^T(t)\,Q_1\Psi(t)+\Psi^T(t)\,Q_2\Psi(t)-$$
$$\Psi^T(t-\tau_1)\,Q_1\Psi(t-\tau_1)-\Psi^T(t-\tau_2)\,Q_2\Psi(t-\tau_2)+\tau_1\dot{\Psi}^T(t)\,Z_1\dot{\Psi}(t)$$
$$+\tau_2\dot{\Psi}^T(t)\,Z_2\dot{\Psi}(t)+(\tau_2-\tau_1)\,\dot{\Psi}^T(t)\,Z_3\dot{\Psi}(t)-\int_{-\tau_1}^{0}\dot{\Psi}^T(t+\theta)\,Z_1\dot{\Psi}(t+\theta)\,d\theta$$
$$-\int_{-\tau_2}^{0}\dot{\Psi}^T(t+\theta)\,Z_2\dot{\Psi}(t+\theta)\,d\theta-\int_{-\tau_2}^{-\tau_1}\dot{\Psi}^T(t+\theta)\,Z_3\dot{\Psi}(t+\theta)\,d\theta \tag{5.43}$$

Let,

$$\widehat{X}=\begin{bmatrix}X(t)^T & X(t-\tau_1)^T & X(t-\tau_2)^T\end{bmatrix} \tag{5.44}$$

then,

$$\begin{aligned}\dot{\Psi} &= \mathscr{E}\dot{X} \\ &= \mathscr{E}\begin{bmatrix}\mathscr{A}_0 & \mathscr{A}_1 & \mathscr{A}_2\end{bmatrix}\widehat{X}^T\end{aligned} \tag{5.45}$$

$$\dot{\Psi}^T(t)P\Psi(t) = \widehat{X}\begin{bmatrix}\mathscr{A}_0 & \mathscr{A}_1 & \mathscr{A}_2\end{bmatrix}^T\mathscr{E}^T P\mathscr{E}X(t) \tag{5.46}$$

$$\Psi^T(t)P\dot{\Psi}(t) = X(t)^T\mathscr{E}^T P\mathscr{E}\begin{bmatrix}\mathscr{A}_0 & \mathscr{A}_1 & \mathscr{A}_2\end{bmatrix}\widehat{X}^T \tag{5.47}$$

$$\Psi^T(t)Q_1\Psi(t) = X(t)^T\mathscr{E}^T Q_1\mathscr{E}X(t) \tag{5.48}$$

$$\Psi^T(t)Q_2\Psi(t) = X(t)^T\mathscr{E}^T Q_2\mathscr{E}X(t) \tag{5.49}$$

$$\Psi^T(t-\tau_1)Q_1\Psi(t-\tau_1) = X(t-\tau_1)^T\mathscr{E}^T Q_1\mathscr{E}X(t-\tau_1) \tag{5.50}$$

$$\Psi^T(t-\tau_2)Q_2\Psi(t-\tau_2) = X(t-\tau_2)^T\mathscr{E}^T Q_2\mathscr{E}X(t-\tau_2) \tag{5.51}$$

For $i = \{1,2,3\}$,

$$\dot{\Psi}^T(t)Z_i\dot{\Psi}(t) = \widehat{X}\begin{bmatrix}\mathscr{A}_0 & \mathscr{A}_1 & \mathscr{A}_2\end{bmatrix}^T\mathscr{E}^T Z_i\mathscr{E}\begin{bmatrix}\mathscr{A}_0 & \mathscr{A}_1 & \mathscr{A}_2\end{bmatrix}\widehat{X}^T \tag{5.52}$$

Consider a set of matrices,

$$\begin{bmatrix}H_{11} & H_{12} & H_{13} & H_{14}\\ H_{12}^T & H_{22} & H_{23} & H_{24}\\ H_{13}^T & H_{23}^T & H_{33} & H_{34}\\ H_{14}^T & H_{24}^T & H_{34}^T & Z_1\end{bmatrix} \succcurlyeq 0 \tag{5.53}$$

$$\begin{bmatrix}I_{11} & I_{12} & I_{13} & I_{14}\\ I_{12}^T & I_{22} & I_{23} & I_{24}\\ I_{13}^T & I_{23}^T & I_{33} & I_{34}\\ I_{14}^T & I_{24}^T & I_{34}^T & Z_2\end{bmatrix} \succcurlyeq 0 \tag{5.54}$$

$$\begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{12}^T & J_{22} & J_{23} & J_{24} \\ J_{13}^T & J_{23}^T & J_{33} & J_{34} \\ J_{14}^T & J_{24}^T & J_{34}^T & Z_3 \end{bmatrix} \succcurlyeq 0 \tag{5.55}$$

Where $H_{ij}, Iij, J_{ij}\ \forall i \in [1,3]\ \forall j \in [1,4]$ are some arbitrary matrices to be found by an LMI solver with size $2n \times 2n$.

Let,

$$\widehat{\Psi}_{\dot{\theta}} = \begin{bmatrix} \Psi(t)^T & \Psi(t-\tau_1)^T & \Psi(t-\tau_2)^T & \dot{\Psi}(t+\theta)^T \end{bmatrix} \tag{5.56}$$

then,

$$\int_{-\tau_1}^{0} \widehat{\Psi}_{\dot{\theta}} \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{12}^T & H_{22} & H_{23} & H_{24} \\ H_{13}^T & H_{23}^T & H_{33} & H_{34} \\ H_{14}^T & H_{24}^T & H_{34}^T & Z_1 \end{bmatrix} \widehat{\Psi}_{\dot{\theta}}^T d\theta \geq 0 \tag{5.57}$$

$$\int_{-\tau_2}^{0} \widehat{\Psi}_{\dot{\theta}} \begin{bmatrix} I_{11} & I_{12} & I_{13} & I_{14} \\ I_{12}^T & I_{22} & I_{23} & I_{24} \\ I_{13}^T & I_{23}^T & I_{33} & I_{34} \\ I_{14}^T & I_{24}^T & I_{34}^T & Z_2 \end{bmatrix} \widehat{\Psi}_{\dot{\theta}}^T d\theta \geq 0 \tag{5.58}$$

$$\int_{-\tau_2}^{-\tau_1} \widehat{\Psi}_{\dot{\theta}} \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{12}^T & J_{22} & J_{23} & J_{24} \\ J_{13}^T & J_{23}^T & J_{33} & J_{34} \\ J_{14}^T & J_{24}^T & J_{34}^T & Z_3 \end{bmatrix} \widehat{\Psi}_{\dot{\theta}}^T d\theta \geq 0 \tag{5.59}$$

The matrices in inequalities (5.53) to (5.55) are chosen to satisfy expression in inequalities (5.57) to (5.59), which further simplifies $\dot{V}$ in Eqn. (5.43). Parts of Eqn. (5.43) consisting integrals with multiplication two variable in terms of θ are eliminated when added with inequalities (5.57) to (5.59), since $\dot{V} + \{positive\,semidefinite\} < 0$ implies $\dot{V} < 0$. Substituting Eqns. (5.44) to (5.52) in Eqn. (5.43), adding inequalities (5.57) to (5.59) and further solving leftover in-

tegrals, inequality (5.60) is obtained.

$$\dot{V} \leq \widehat{X} \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{12}^T & G_{22} & G_{23} \\ G_{13}^T & G_{23}^T & G_{33} \end{bmatrix} \widehat{X}^T \tag{5.60}$$

where,

$$\begin{aligned} G_{11} =& \mathscr{A}_0^T \mathscr{E}^T P \mathscr{E} + \mathscr{E}^T P \mathscr{E} \mathscr{A}_0 + \mathscr{E}^T Q_1 \mathscr{E} + \mathscr{E}^T Q_2 \mathscr{E} + \mathscr{A}_0^T \mathscr{E}^T \Xi \mathscr{E} \mathscr{A}_0 + \tau_1 \mathscr{E}^T H_{11} \mathscr{E} + \\ & \mathscr{E}^T H_{14} \mathscr{E} + \mathscr{E}^T H_{14}^T \mathscr{E} + \tau_2 \mathscr{E}^T I_{11} \mathscr{E} + \mathscr{E}^T I_{14} \mathscr{E} + \mathscr{E}^T I_{14}^T \mathscr{E} + (\tau_2 - \tau_1) \mathscr{E}^T J_{11} \mathscr{E} \end{aligned} \tag{5.61}$$

$$\begin{aligned} G_{12} =& \mathscr{E}^T P \mathscr{E} \mathscr{A}_1 + \mathscr{A}_0^T \mathscr{E}^T \Xi \mathscr{E} \mathscr{A}_1 + \tau_1 \mathscr{E}^T H_{12} \mathscr{E} - \mathscr{E}^T H_{14} \mathscr{E} + \mathscr{E}^T H_{24}^T \mathscr{E} + \tau_2 \mathscr{E}^T I_{12} \mathscr{E} + \\ & \mathscr{E}^T I_{24}^T \mathscr{E} + (\tau_2 - \tau_1) \mathscr{E}^T J_{12} \mathscr{E} + \mathscr{E}^T J_{14} \mathscr{E} \end{aligned} \tag{5.62}$$

$$\begin{aligned} G_{13} =& \mathscr{E}^T P \mathscr{E} \mathscr{A}_2 + \mathscr{A}_0^T \mathscr{E}^T \Xi \mathscr{E} \mathscr{A}_2 + \tau_1 \mathscr{E}^T H_{13} \mathscr{E} + \mathscr{E}^T H_{34}^T \mathscr{E} + \tau_2 \mathscr{E}^T I_{13} \mathscr{E} + \mathscr{E}^T I_{34}^T \mathscr{E} - \\ & \mathscr{E}^T I_{14} \mathscr{E} + (\tau_2 - \tau_1) \mathscr{E}^T J_{13} \mathscr{E} - \mathscr{E}^T J_{14} \mathscr{E} \end{aligned} \tag{5.63}$$

$$\begin{aligned} G_{22} =& -\mathscr{E}^T Q_1 \mathscr{E} + \mathscr{A}_1^T \mathscr{E}^T \Xi \mathscr{E} \mathscr{A}_1 + \tau_1 \mathscr{E}^T H_{22} \mathscr{E} - \mathscr{E}^T H_{24} \mathscr{E} - \mathscr{E}^T H_{24}^T \mathscr{E} + \tau_2 \mathscr{E}^T I_{22} \mathscr{E} + \\ & (\tau_2 - \tau_1) \mathscr{E}^T J_{22} \mathscr{E} + \mathscr{E}^T J_{24} \mathscr{E} + \mathscr{E}^T J_{24}^T \mathscr{E} \end{aligned} \tag{5.64}$$

$$\begin{aligned} G_{23} =& \mathscr{A}_1^T \mathscr{E}^T \Xi \mathscr{E} \mathscr{A}_2 + \tau_1 \mathscr{E}^T H_{23} \mathscr{E} - \mathscr{E}^T H_{34}^T \mathscr{E} + \tau_2 \mathscr{E}^T I_{23} \mathscr{E} - \mathscr{E}^T I_{24} \mathscr{E} + \\ & (\tau_2 - \tau_1) \mathscr{E}^T J_{23} \mathscr{E} - \mathscr{E}^T J_{24} \mathscr{E} + \mathscr{E}^T J_{34}^T \mathscr{E} \end{aligned} \tag{5.65}$$

$$\begin{aligned} G_{33} =& -\mathscr{E}^T Q_2 \mathscr{E} + \mathscr{A}_2^T \mathscr{E}^T \Xi \mathscr{E} \mathscr{A}_2 + \tau_1 \mathscr{E}^T H_{33} \mathscr{E} - \mathscr{E}^T I_{34} \mathscr{E} - \mathscr{E}^T I_{34}^T \mathscr{E} + \tau_2 \mathscr{E}^T I_{33} \mathscr{E} + \\ & (\tau_2 - \tau_1) \mathscr{E}^T J_{33} \mathscr{E} - \mathscr{E}^T J_{34} \mathscr{E} - \mathscr{E}^T J_{34}^T \mathscr{E} \end{aligned} \tag{5.66}$$

$$\Xi = \tau_1 Z_1 + \tau_2 Z_2 + (\tau_2 - \tau_1) Z_3 \tag{5.67}$$

The matrices P, Q_1, Q_2, Z_i, H_{ij}, H_{ij}^T, I_{ij}, I_{ij}^T, J_{ij}, J_{ij}^T and $\mathscr{F}_i$ from lemma 4 $\forall i \in [1,3]$, $\forall j \in [1,4]$ will generate corresponding $\widetilde{P}$, $\widetilde{Q}_1$, $\widetilde{Q}_2$, $\widetilde{Z}_i$, $\widetilde{H}_{ij}$, $\widetilde{H}_{ij}^T$, $\widetilde{I}_{ij}$, $\widetilde{I}_{ij}^T$, $\widetilde{J}_{ij}$, $\widetilde{J}_{ij}^T$ and $\widetilde{\mathscr{F}}_i$ $\forall i \in [1,3]$, $\forall j \in [1,4]$ of size $(2n-2) \times (2n-2)$ when multiplied with eigenvector matrices U^T, U at appropriate positions. The corresponding G_{ij} is as given in Eqn. (5.68).

$$\begin{aligned} G_{ij} &= U \begin{bmatrix} \widetilde{G}_{ij_{(2n-2)\times(2n-2)}} & 0_{(2n-2)\times 2} \\ 0_{2\times(2n-2)} & 0_{2\times 2} \end{bmatrix} U^T \\ & \{i,j\} \in [1,3];\ \widetilde{G}_{ji} = \widetilde{G}_{ji}^T \ \forall i \neq j \end{aligned} \tag{5.68}$$

With above set of reduced order matrices, the LMIs given in Eqns. (5.31) to (5.34) can be obtained. □

Feasibility of LMIS proposed in Eqns. (5.31) to (5.34) determine the consensus reachability of the multi-agent system affected by time-delays. The LMIs are completely different when compared to existing results in the literature [77]. They can be solved by using solvers like SeDuMi [115], Matlab LMI Lab solver etc..

5.3 Simulation and Implementation Results

The communication topologies considered for simulation and implementation are depicted in Figs. 5.3a and 5.3b. The graph in Fig. 5.3a is undirected, strongly connected and 2-regular balanced with each node receiving states' information from two neighbours and sending states' information to the same neighbours. The graph in Fig. 5.3b is directed, has a spanning tree and unbalanced. Using the results obtained in Theorems 1 to 3, limits on communication time-delay for given input-delays are calculated for both the topologies. The feasibility of LMIs given in Theorem 3 is solved using SeDuMi [115] solver for Matlab/Octave. The expressions in Theorems 1 and 2 have three unknowns (ω, τ_1, τ_2), a unique solution can be obtained if it is assumed that $\tau_1 = \tau_2$ or else, stable range of τ_2 for a given τ_1 have to be found. Dominant pole for both the topologies in Figs. 5.3a and 5.3b is -1, Nyquist plot used in one of the cases with $\lambda = -1$ and assumption $\tau_1 = 0.4$ is shown in Fig. 5.4. It can be observed that the system is stable if $\tau_2 < 0.48$. The results obtained using Lyapunov-Krasovskii and Nyquist approaches from Theorems 1 to 3 are tabulated in Tables 5.1 and 5.2, corresponding plots of τ_1 vs τ_2 depicting stable regions are given in Figs. 5.6a and 5.6a.

From the results in Tables 5.1 and 5.2 and Figs. 5.6a and 5.6b, it can be deduced that the Lyapunov-Krasovskii approach is conservative compared to Nyquist approach with respect to time-delay. Conservativeness of Lyapunov-Krasovskii approach is more evident for topology in Fig. 5.3b with control law in Eqn. (5.6). Multi-agent systems connected by topologies in Figs. 5.3a and 5.3b reach consensus with full range of time-delay given by Nyquist approach

(a) Four agents. (b) Five agents.

Figure 5.3: Graphs of communication topologies.

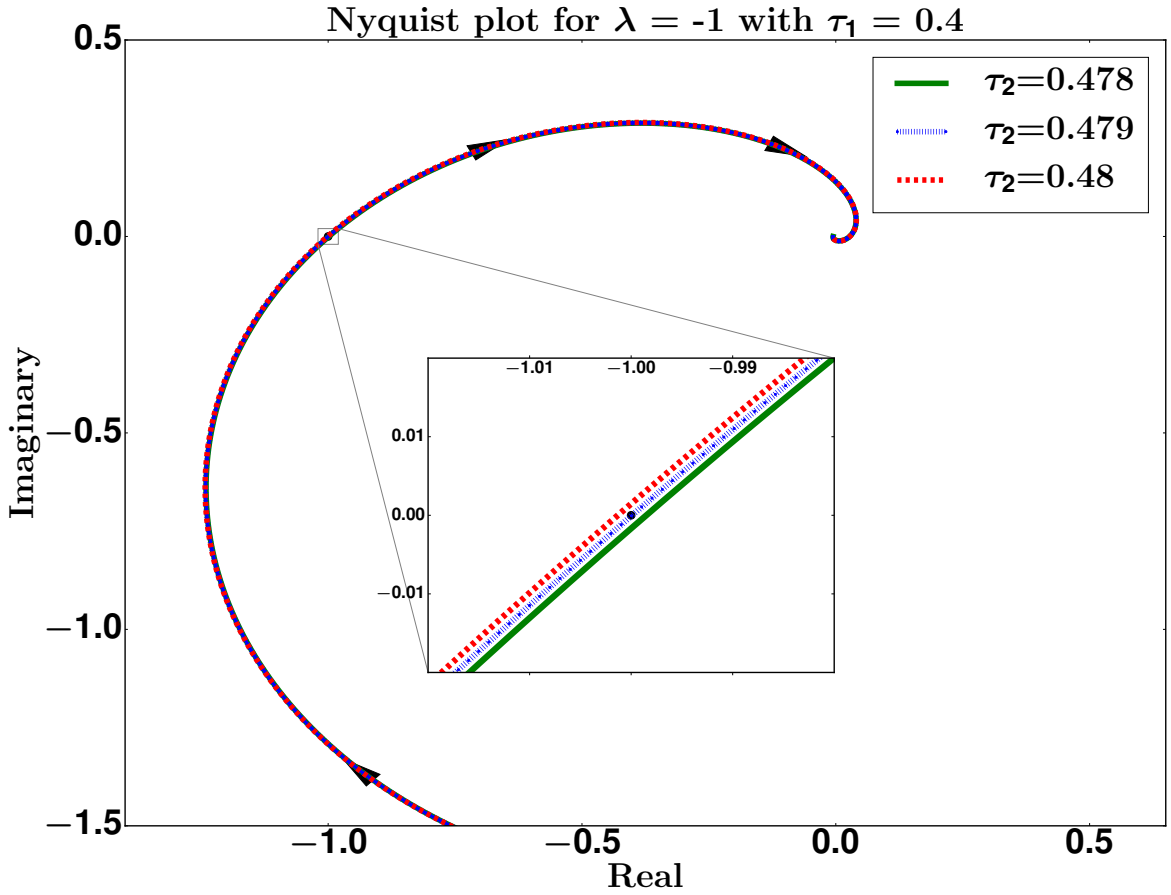


Figure 5.4: Nyquist plot with u_{i1} in Eqn. (5.5) for different time delays.

for control laws in Eqns. (5.5) and (5.6). Nyquist approach is not applicable to control laws given in Eqns. (5.7) and (5.8). Solving LMIs is computationally more intensive compared to solving of equations from Theorems 1 and 2. Moreover, the increase in computational time of solving LMIs is exponentially as the number of nodes are increased and the increase with Nyquist approach is linear.

Simulations are performed using scripts written in C to have uniformity with hardware implementation. Implementations of corresponding simulations are performed on a network of

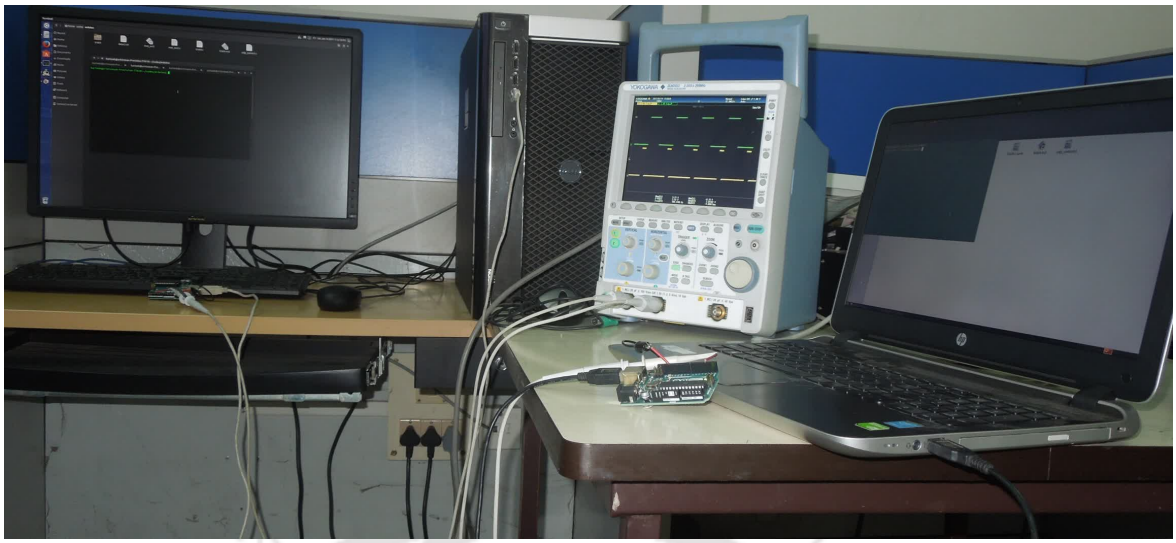


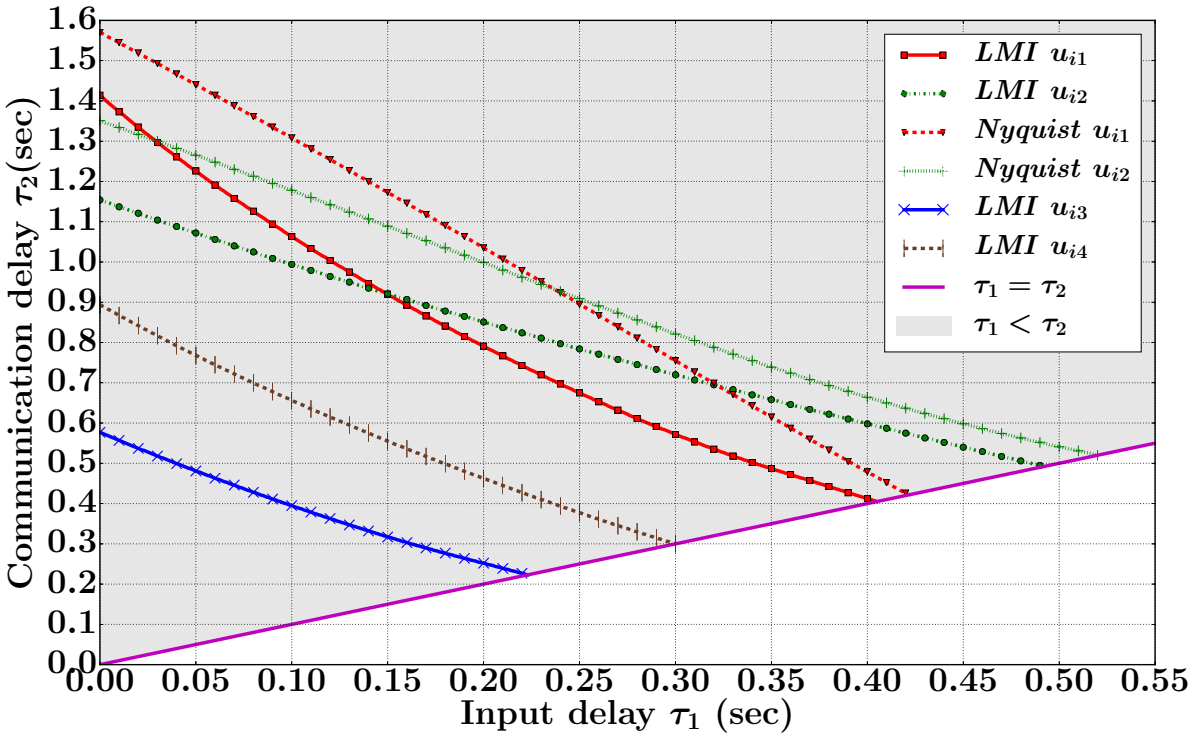
Figure 5.5: Hardware setup connected by LAN.

Table. 5.1: Maximum value of τ_2 for a given τ_1 $(\tau_1 \leq \tau_2)$ with topology in Fig. 5.3a and control inputs given in Eqns. (5.5) to (5.8).

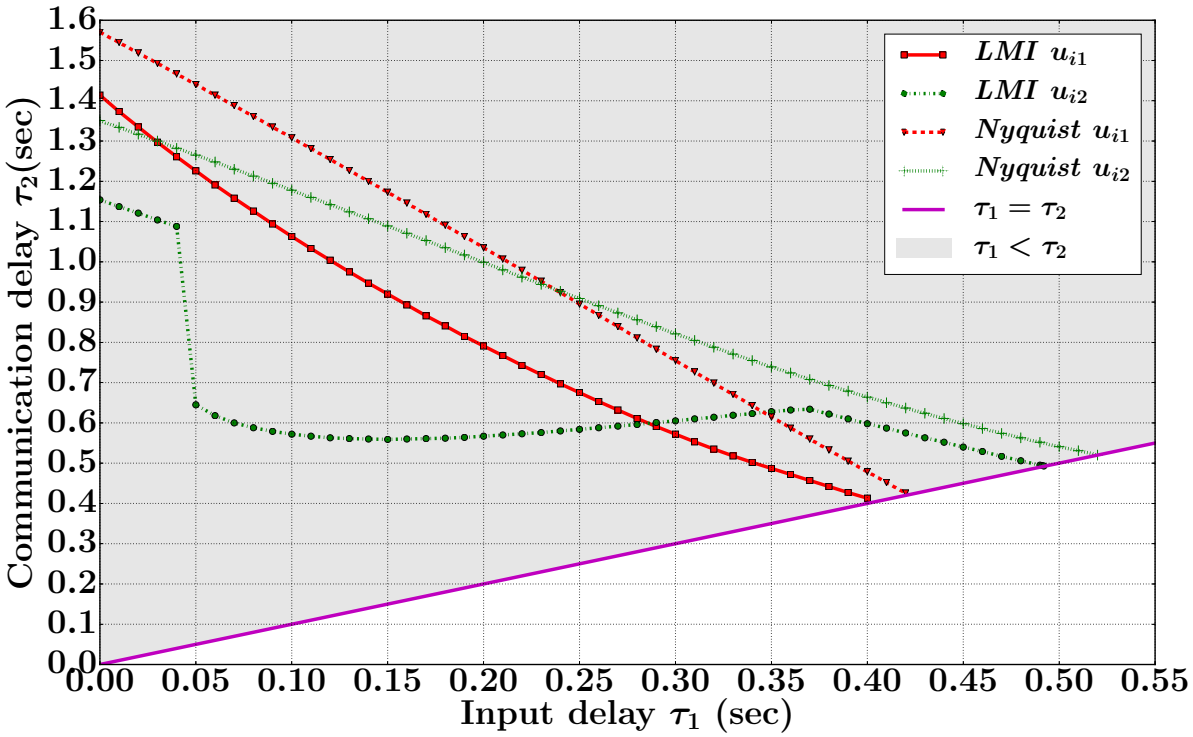
τ_1 **(sec)**	τ_2 **(sec)**					
	Lyapunov Approach				*Nyquist Approach*	
	u_{i1}	u_{i2}	u_{i3}	u_{i4}	u_{i1}	u_{i2}
0	1.414	1.154	0.577	0.894	1.570	1.351
0.1	1.063	0.994	0.395	0.656	1.308	1.178
0.2	0.791	0.851	0.252	0.462	1.035	0.999
0.3	0.572	0.720	—	0.30	0.755	0.821
0.4	0.413	0.598	—	—	0.479	0.664
0.5	—	—	—	—	—	0.541
0.6	—	—	—	—	—	—
$\tau_1 = \tau_2$	0.405	0.492	0.223	0.3	0.421	0.520

Table. 5.2: Maximum value of τ_2 for a given τ_1 $(\tau_1 \leq \tau_2)$ with topology in Fig. 5.3b and control inputs given in Eqns. (5.5) and (5.6).

τ_1 **(sec)**	τ_2 **(sec)**			
	Lyapunov Approach		*Nyquist Approach*	
	u_{i1}	u_{i2}	u_{i1}	u_{i2}
0	1.414	1.154	1.570	1.351
0.1	1.063	0.572	1.308	1.178
0.2	0.791	0.567	1.035	0.999
0.3	0.572	0.605	0.755	0.821
0.4	0.413	0.598	0.479	0.664
0.5	—	—	—	0.541
0.6	—	—	—	—
$\tau_1 = \tau_2$	0.405	0.492	0.421	0.520



(a) Plot of τ_1 vs τ_2 for topology in Fig. 5.3a.



(b) Plot of τ_1 vs τ_2 for topology in Fig. 5.3b.

Figure 5.6: Plots representing stable regions with $\tau_1 \leq \tau_2$.

four agents. Four Arduino-Uno boards for topology in Fig. 5.3a and five Arduino-Uno boards for topology in Fig. 5.3b are considered as nodes of sensor networks, all of them are connected to host computers using serial interface. The mode operation in terms of step size, synchronization and implementation of time-delays are same as discussed in Chapter-4. Hardware setup depicted in Fig. 5.5 consists of two agents with connection settings as described earlier, pulse width of both the agents can be observed in the display of two channel DSO.

Input time-delay τ_1 and communication time-delay τ_2 are user-defined and implemented from code. A minor deviation is visible between simulation and implementation results due to additional delay ($\approx 4ms$) in implementation due to actual processing and communication. The pulse width of PWM wave generated from an Arduino-Uno is considered as state x_i and the rate of change is considered as v_i of each agent.

Initial conditions for both simulation and implementation for topology in Fig. 5.3a are assumed to be, $\{x_{i0}\} = [0, 230, 110, 40]$ and $\{v_{i0}\} = [0, 0, 0, 0]$. Plots with overlapping simulation and hardware results are given in Figs. 5.7 to 5.9 to show the effectiveness of theoretical results given in Tables 5.1 to 5.3 and Figs. 5.6a and 5.6b. For control law in Eqn. (5.7) with input delay $\tau_1 = 0.1s$ and communication delay of $\tau_2 = 0.39s$, Fig. 5.7 depict states $x_i(t)$ and $\dot{x}_i(t)$ vs time in seconds respectively. Similarly, Fig. 5.8 show the plots of states for control law in Eqn. (5.8) with $\tau_1 = 0.2s$ and of $\tau_2 = 0.46s$. It can be observed that the difference in states, Ψ is asymptotically converging to zero and reinforcing the effectiveness of the theoretical results given in Table 5.1 and Fig. 5.6a.

Similarly, some simulations and corresponding hardware validations are performed on a five-agent system with communication topology given in Fig. 5.3b, the corresponding adjacency matrix $\mathscr{A}$ and $\tilde{\mathscr{A}}$ are as given below,

$$\mathscr{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tilde{\mathscr{A}} = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

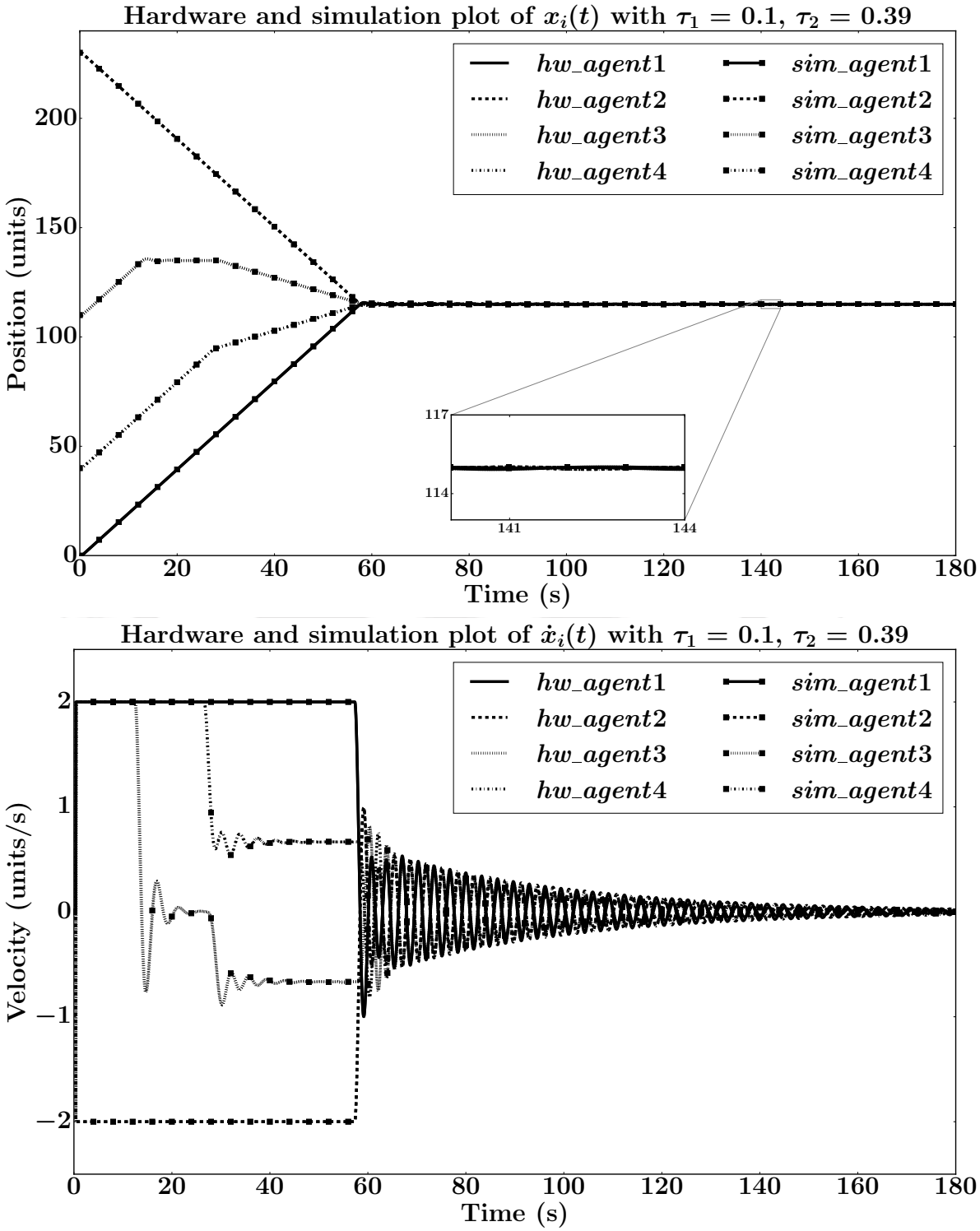


Figure 5.7: Plots with u_{i3} in Eqn. (5.8) for asynchronous time delays.

The initial values for the five-agent system are considered as $x_1(0) = 0$, $x_2(0) = 230$, $x_3(0) = 110$, $x_4(0) = 40$, $x_5(0) = 170$ and $\dot{x}_i(0) = 0$, $i \in [1,5]$. From Fig. 5.9, it can be observed that the system converges at a faster rate with $\tau_1 = 0.2$ and $\tau_2 = 0.79$. With $\tau_1 = 0.2$ and $\tau_2 = 1.03$

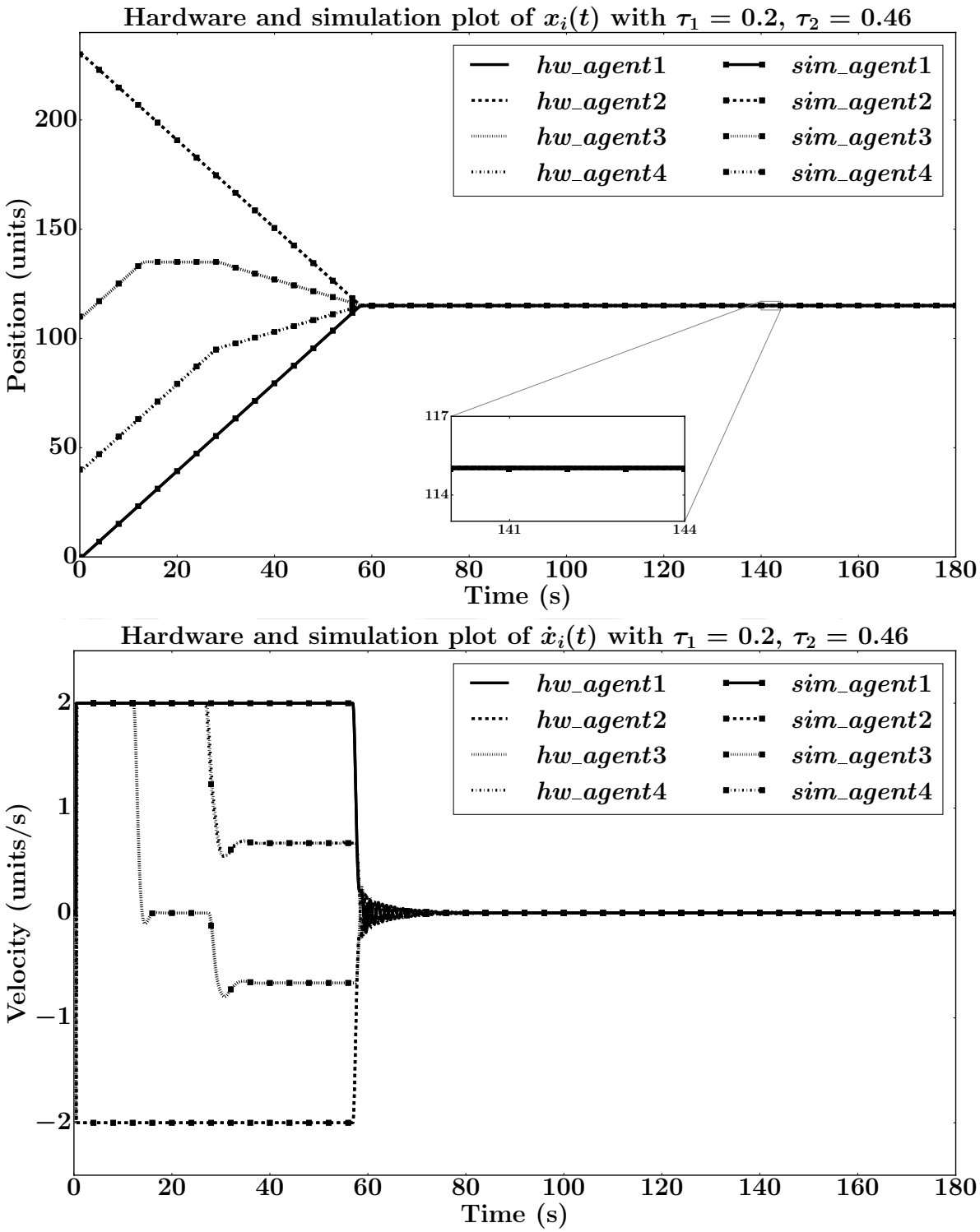


Figure 5.8: Plots with u_{i4} in Eqn. (5.8) for asynchronous time delays.

given by Nyquist approach, the system converges at a very slow rate in simulation and further slower in implementation due to added delay from communication links as discussed earlier in Chapter-4. The corresponding numerical values for Nyquist approach for $\lambda = -1$ are provided

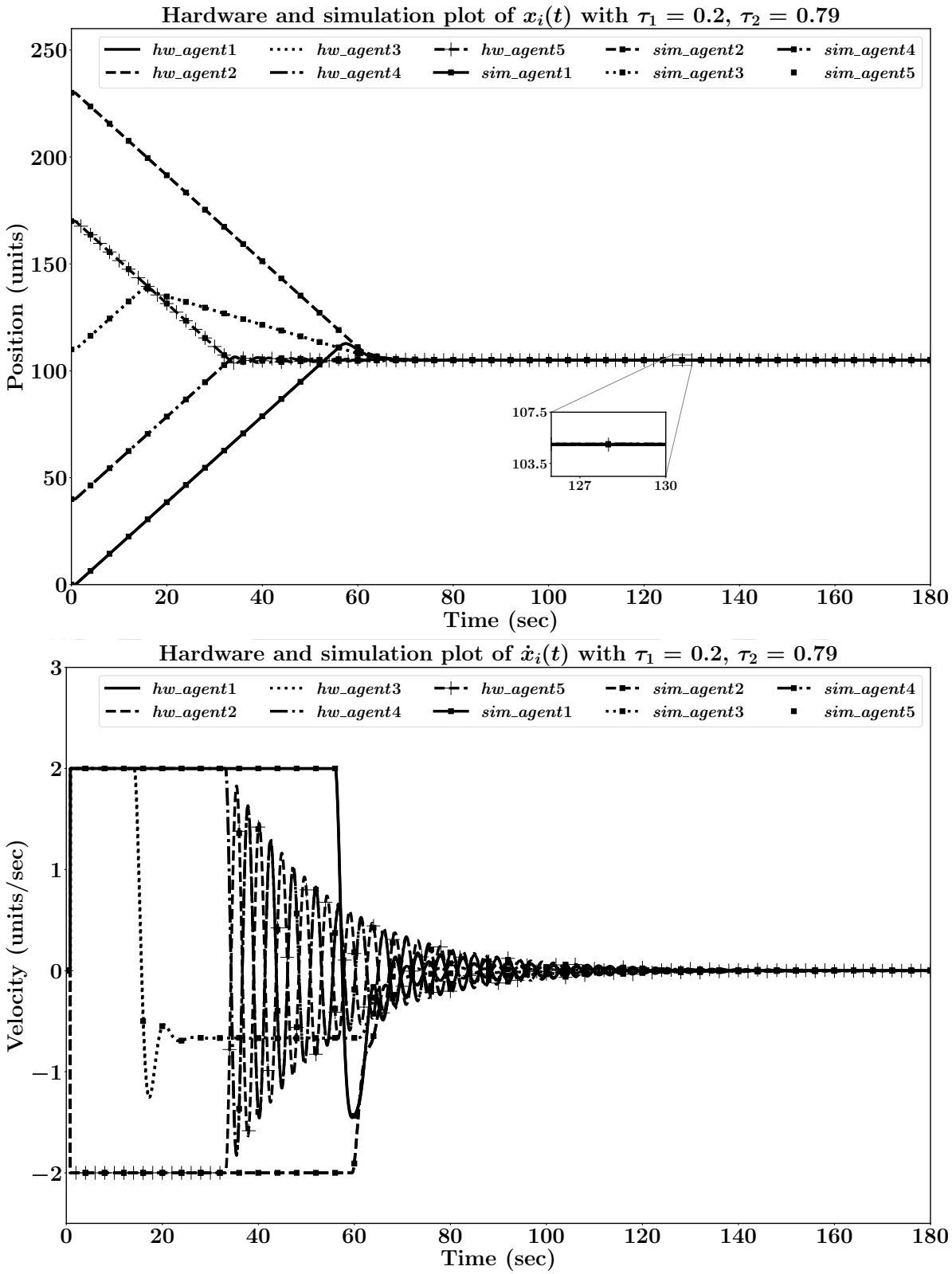


Figure 5.9: Plot with u_{i1} in Eqn. (5.5) for uneven time delays.

in Table 5.3.

Using results in Theorems 1 to 3, stable regions with respect to time-delays for both the

topologies in Figs. 5.3a and 5.3b are calculated. Few of them are validated with the help of simulations and corresponding implementations as given above. Figs. 5.6a and 5.6b depict the calculated stable regions for topologies in Figs. 5.3a and 5.3b respectively.

Table. 5.3: Values of $\overline{\omega}$ & $\left|G_i(\overline{\omega})\right|$ for different values of τ_1 and τ_2 with u_{i1} in Eqn. (5.5) and communication topology in Fig. 5.3b.

τ_1 (sec)	τ_2 (sec)	$\overline{\omega}$	$\left\|G_i(\overline{\omega})\right\|$
0.2	0.79	1.375	0.841
0.2	1.03	1.249	0.997
0.2	1.04	1.244	1.002

5.3.1 Remarks

1. Lyapunov-Krasovskii is applicable to all the four control laws provided the communication topology satisfy the conditions mentioned in Theorem 3, but Nyquist approach is not applicable. For control inputs in Eqns. (5.5) and (5.6), the mathematical analysis given in Eqns. (4.7) to (4.9) and (4.19) to (4.21) is possible when the diagonal elements of $\widetilde{\mathscr{A}}$ are unity.

2. Lyapunov-Krasovskii approach is more conservative with respect to time-delay tolerance, whereas Nyquist approach gives the full range of time-delay. Using Nyquist criterion, we get necessary and sufficient conditions. With Lyapunov approach, we get only sufficient conditions.

3. Describing function analysis allows us to use Nyquist approach on approximated nonlinear multi-agent system, which is better at providing time-delay tolerance ranges compared to Lyapunov approach.

4. Compared to work in [116], we have considered saturation and time-delays in the system. An approximate analysis with the help of describing function is performed. Few conditions in Theorems 1 to 3 are derived for reaching consensus and estimation of non-existence of limit cycles.

5. Compared to the research presented by authors in [90–93, 95], we have considered time-delays along with saturation in second order multi-agent systems. Compared to the work presented by You *et al.* [96], asynchronous time-delays are considered rather than single delay.

5.4 Multi-agent Systems with General Nonlinear Input

In certain applications, the control input is required to limit the velocity rather than operating at the maximum limits. Moreover, the agents can also be heterogeneous with different constraints. In this section, a multi-agent system with both velocity and acceleration constraints is analysed. Consider a network of n second order heterogeneous agents connected by an undirected graph topology, where i^{th} agent have the dynamics as given below:

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= u_i \end{aligned} \tag{5.69}$$

The heterogeneity of the agents is described by the control input. Control law proposed for $x_i \to x_j$ and $v_i \to 0$ as $t \to \infty$ is given in Eqn. (5.70). It can be observed from the control law, that the agents can have different maximum velocities ($vmax_i$) and different scaling factors (β_i) of the control input.

$$u_i = \big(1 - f_1\left(\hat{u}_i, v_i\right) \times f_2\left(vmax_i, v_i\right)\big) \times \hat{u}_i \tag{5.70}$$

$$\hat{u}_i = \beta_i\big[-g_1\left(k_1 v_i\right) + \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_j - x_i\right)\right)\big] \tag{5.71}$$

The function $f_1\left(\hat{u}_i, v_i\right)$ returns 1 when $sign(\hat{u}_i) = sign(v_i)$ and returns 0 when $sign(\hat{u}_i) \neq sign(v_i)$. The function $f_2\left(vmax_i, v_i\right)$ returns 1 when $vmax_i \geq |v_i|$ and returns 0 when $vmax_i < |v_i|$. Thus the control input of each agent is governed by the constraints that are independent of other agents.

The functions g_1 and g_2 are general nonlinear odd functions symmetric to origin with

$xg_1(x) > 0$, $xg_2(x) > 0$; $\forall |x| > 0$ and $g_1(0) = 0$, $g_2(0) = 0$. G_1, G_2 are continuous and differentiable with $\dot{G}_1 = g_1, \dot{G}_2 = g_2$. β_i, k_1, k_2 are user defined positive constants that act as scaling factors. Describing functions cannot be used when there are multiple nonlinearities acting on each agent. If an agent with higher velocity is not receiving information from an agent with lower velocity, consensus can't be guaranteed. To overcome the issue, an undirected and connected network is considered for the analysis. In an undirected network, communication among agents is bidirectional.

Consider a function g, which is an integrable nonlinear odd function and symmetric to origin with $xg(x) > 0$; $\forall |x| > 0$ and $g(0) = 0$. G is a continuous and differentiable function which satisfy $\dot{G}(x) = g(x)$.

Lemma 6. [117] *Consider an integrable nonlinear odd function $g(x)$, that is symmetric to origin. If $\dot{G}(x) = g(x)$, where $G(x)$ is a continuous and differentiable function, then $G(x)$ is an even function.*

Proof. $G(x)$ can be written as,

$$G(x) = \int_0^x g(s)ds \tag{5.72}$$

$$G(-x) = \int_0^{-x} g(s)ds \tag{5.73}$$

Let $s = -t$, then, $ds = -dt$. At $s = 0$, $t = 0$ and at $s = -x$, $t = x$,

$$\begin{aligned} G(-x) &= -\int_0^x g(-t)dt \\ &= \int_0^x g(t)dt \\ &= G(x) \end{aligned} \tag{5.74}$$

□

Lemma 7. *For every $g(x)$, there is always some integral constant (C) such that $G(x) + C > 0$, $\forall x \neq 0$ and $G(0) + C \geq 0$.*

Proof. Based on the definition of function g, $g(x) > 0$; $\forall\, x > 0$.

$$\int g(x) = G(x) + C \tag{5.75}$$

The slope of $G(x)$ is $g(x)$ and it is positive $\forall\, x > 0$, which implies $G(x)$ is increasing $\forall\, x > 0$. If an appropriate integral constant (C) to make $G(0) + C \geq 0$ is chosen, then $G(x) + C > 0$; $\forall\, x > 0$.

From lemma 6, $G(x) + C > 0$, $\forall x \neq 0$ and $G(0) + C \geq 0$. □

Consider following functions for $g(x)$ in Eqns. (5.76), (5.78) and (5.80) and corresponding $G(x) + C$ in Eqns. (5.77), (5.79) and (5.81) to demonstrate the effectiveness of lemmas 6 and 7. Figure 5.10 depict the plots of functions $g(x)$ in Eqn. (5.76) and its corresponding integral in Eqn. (5.77). In a similar way, Figs. 5.11a and 5.11b depict the plots of functions in Eqns. (5.78) to (5.81).

$$g(x) = \tanh(x) \tag{5.76}$$

$$G(x) + C = \log(\cosh(x)) \tag{5.77}$$

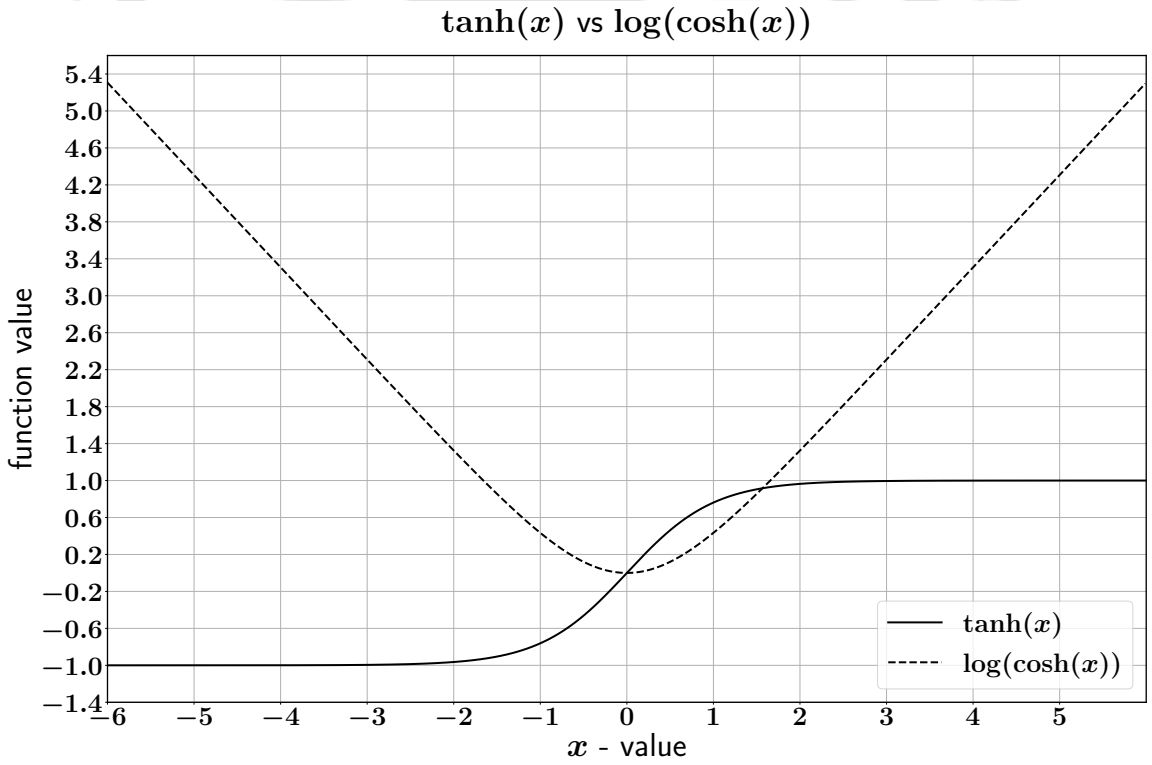


Figure 5.10: Plot of odd function given in Eqn. (5.76) and its integral in Eqn. (5.77) with appropriate constants.

$$g(x) = sat(kx) = \begin{cases} -1; & x \leq -\frac{1}{k} \\ kx; & -\frac{1}{k} < x < \frac{1}{k} \\ 1; & x \geq \frac{1}{k} \end{cases} \tag{5.78}$$

$$G(x)+C = \begin{cases} -x; & x \leq -\frac{1}{k} \\ 0.5\left[\frac{1}{k} + kx^2\right]; & -\frac{1}{k} < x < \frac{1}{k} \\ x; & x \geq \frac{1}{k} \end{cases} \tag{5.79}$$

$$g(x) = \begin{cases} x - sin(3) + 3; & x \leq -3 \\ sin(x); & -3 < x < 3 \\ x + sin(3) - 3; & x \geq 3 \end{cases} \tag{5.80}$$

$$G(x)+C = \begin{cases} 0.5 * x^2 - cos(3) + 1 - 0.5 * 3 * 3; & x \leq -3 \\ -cos(x) + 1; & -3 < x < 3 \\ 0.5 * x^2 - cos(3) + 1 - 0.5 * 3 * 3; & x \geq 3 \end{cases} \tag{5.81}$$

Lemma 8. *In Eqns.* (5.70) *and* (5.71)*, when* $f_1(\hat{u}_i, v_i) = 1$*, then* $v_i \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right) < 0$ *and* $v_i \sum_{j=1}^{n} a_{ij} g_2\left(x_i - x_j\right) < 0$ *are satisfied.*

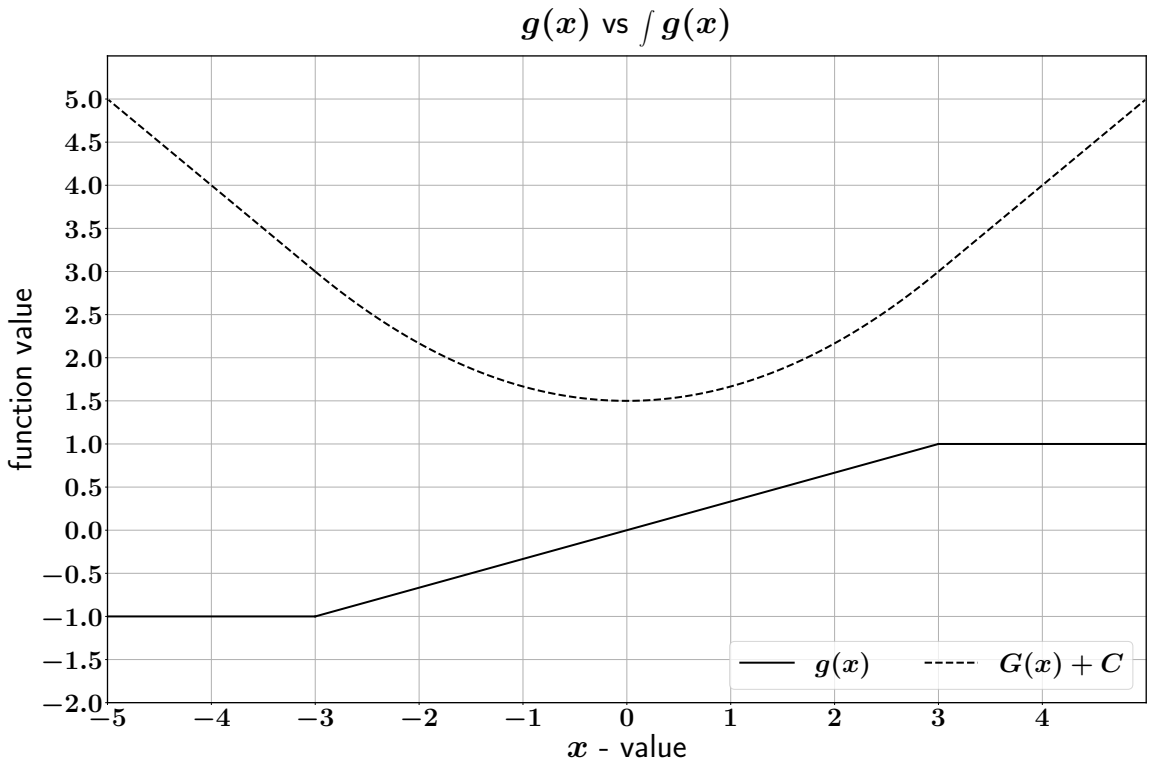
Proof. When $f_1(\hat{u}_i, v_i) = 1$, β_i, k_1, k_2 are positive constants, then according to functionality of f_1,

$$v_i \hat{u}_i > 0 \tag{5.82}$$

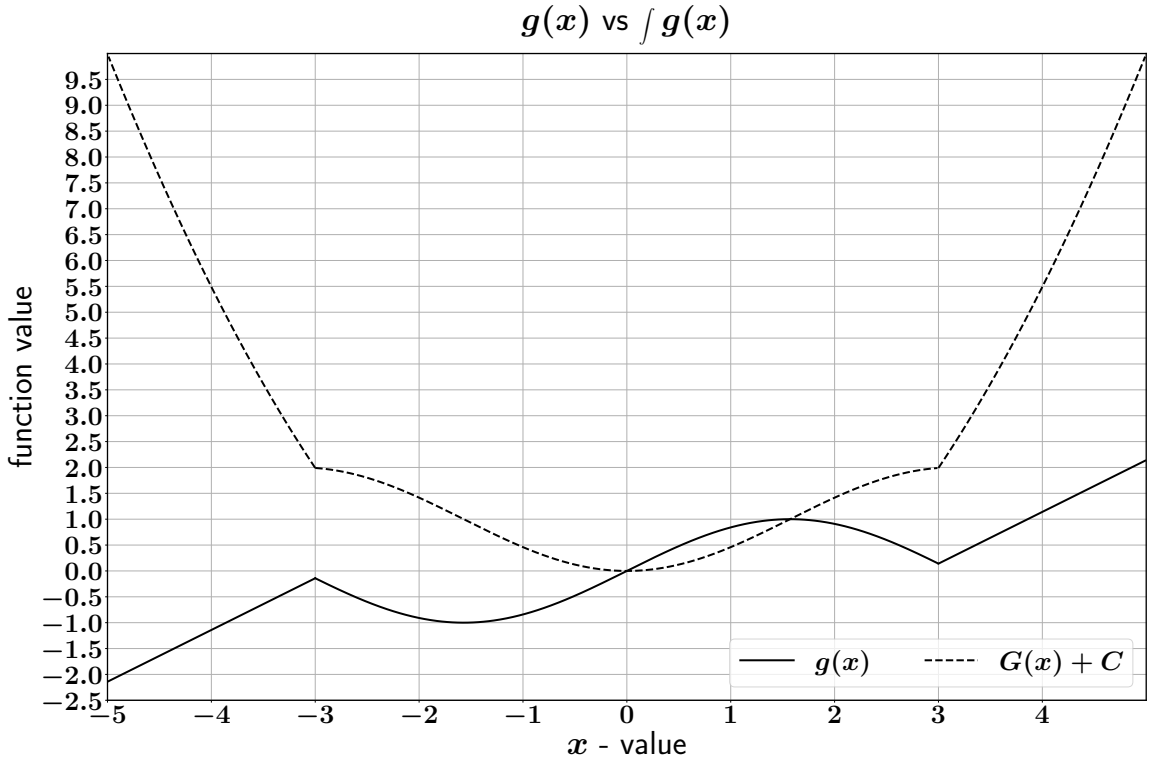
$$v_i \beta_i \left[-g_1(k_1 v_i) + \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_j - x_i\right)\right) \right] > 0 \tag{5.83}$$

$$v_i \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right) < -v_i g_1(k_1 v_i) \tag{5.84}$$

$$v_i \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right) < 0 \tag{5.85}$$



(a) Plot of odd function given in Eqn. (5.78) for $k = \frac{1}{3}$ and its integral in Eqn. (5.79)with appropriate constants.



(b) Plot of odd function given in Eqn. (5.80) and its integral in Eqn. (5.81) with appropriate constants.

Figure 5.11: Plots of odd functions and corresponding integrals.

$$v_i \sum_{j=1}^{n} a_{ij} g_2\left(x_i - x_j\right) < 0 \tag{5.86}$$

□

Theorem 4. *For an undirected network topology that is connected strongly, control law u_i in Eqn.* (5.70) *solves consensus problem with $x_i \to x_j$ and $v_i \to 0$ as $t \to \infty$.*

Proof. Using Lemmas 6 and 7, there is always some $G_2(x) + C \geq 0$ for a given $g_2(x)$ as described above. Let the Lyapunov function be,

$$V = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} k_2^{-1} \left(G_2\left(k_2\left(x_i - x_j\right)\right) + C\right) + \frac{1}{2} \sum_{i=1}^{n} \beta_i^{-1} v_i^2 \tag{5.87}$$

Using control law in Eqn. (5.70),

$$\begin{aligned}
\dot{V} =& \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right)\left(\dot{x}_i - \dot{x}_j\right) + \sum_{i=1}^{n} \beta_i^{-1} v_i \dot{v}_i \\
=& \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right)\left(v_i - v_j\right) + \sum_{i=1}^{n} \beta_i^{-1} v_i \left(1 - f_1\left(\hat{u}_i, v_i\right) \times f_2\left(vmax_i, v_i\right)\right) \times \hat{u}_i \\
=& \sum_{i=1}^{n} v_i \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right) + \sum_{i=1}^{n} \beta_i^{-1} v_i \left(1 - f_1\left(\hat{u}_i, v_i\right) \times f_2\left(vmax_i, v_i\right)\right) \times \hat{u}_i
\end{aligned} \tag{5.88}$$

At any given time, let Φ represent the set of agents which has both f_1 and f_2 values as unity.

$$\begin{aligned}
\dot{V} =& \sum_{i=1;\, i \notin \Phi}^{n} v_i \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right) + \sum_{i=1;\, i \notin \Phi}^{n} v_i \left[-g_1\left(k_1 v_i\right) + \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_j - x_i\right)\right)\right] + \\
& \sum_{i=1;\, i \in \Phi}^{n} v_i \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right)
\end{aligned} \tag{5.89}$$

$$= - \sum_{i=1;\, i \notin \Phi}^{n} v_i g_1\left(k_1 v_i\right) + \sum_{i=1;\, i \in \Phi}^{n} v_i \sum_{j=1}^{n} a_{ij} g_2\left(k_2\left(x_i - x_j\right)\right) \tag{5.90}$$

Using Lemma 8,

$$\dot{V} < 0 \tag{5.91}$$

This implies that consensus is possible with $x_i \to x_j$ and $v_i \to 0$ as $t \to \infty$, for a multi-agent system with an undirected connected graph topology using the control law in Eqn. (5.70). □

The values of scalars k_1 and k_2 will not have any affect on the proof of Theorem 4. The only affect is on convergence time. The convergence time is more for lower values of k_1 and k_2 compared with higher values of k_1 and k_2.

5.5 Simulation and Implementation Results

The simulation and implementation are performed on a similar setup as discussed in sections 4.3 and 5.3. The communication topology considered is an undirected and connected network, it is as shown in Fig. 5.12.

Figure 5.12: Strongly connected graph topology.

The initial initial position and initial velocity of five agents are considered as x_{i0} = [0, 230, 110, 40, 170], v_{i0} = [1, 0.5, 0.8, 2, 0.75]. Maximum velocity of each agent is assumed to be $vmax_i$ = [1.5, 0.75, 1.6, 2.5, 1.0] and control input scaling factors as β_i = [0.6, 0.8, 1.2, 1, 2.5]. The nonlinear functions g_1 and g_2 are assumed to be tanh() function with scaling factors $k_1 = k_2 = 1$. Plots of position and velocity states versus time for both simulation and implementation are depicted in Figs. 5.13 and 5.14. It can be observed that the multi-agent system reaches consensus. This reinforces the theoretical results obtained above.

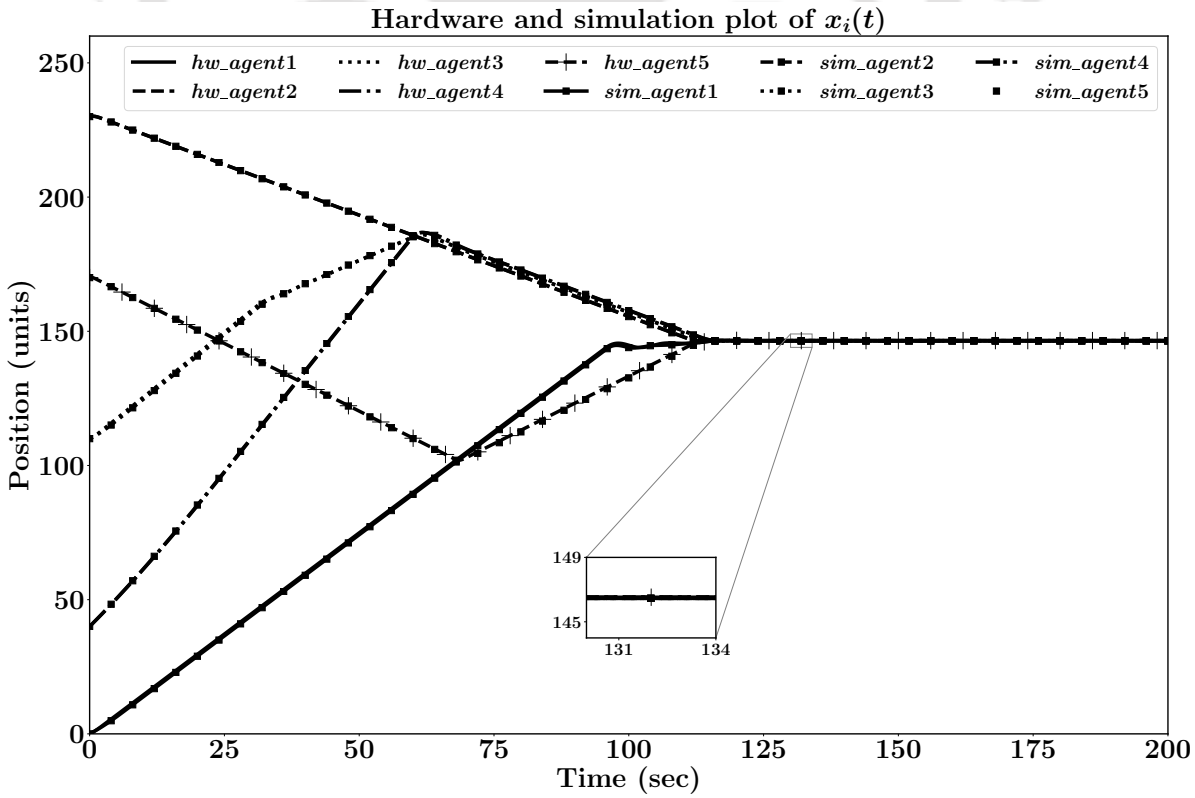


Figure 5.13: Plot of x_i vs t with u_i in Eqn. (5.70) for topology in Fig. 5.12.

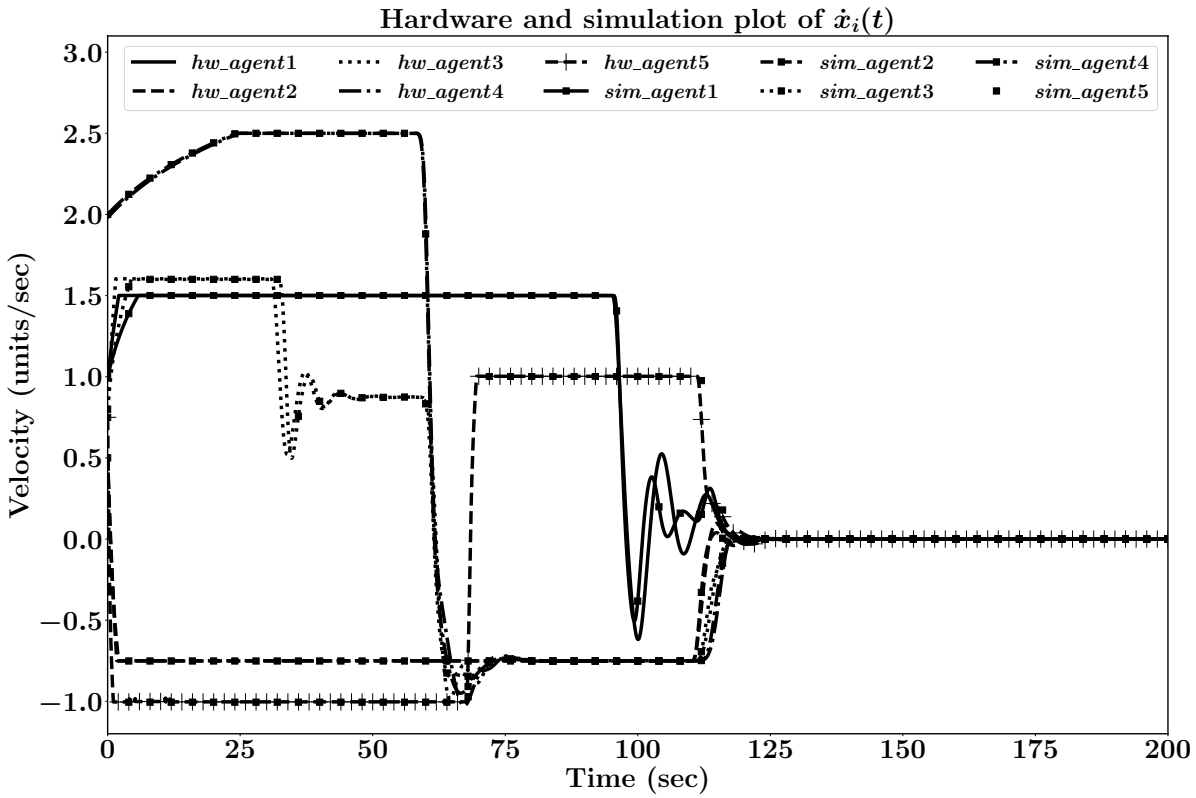


Figure 5.14: Plot of v_i vs t with u_i in Eqn. (5.70) for topology in Fig. 5.12.

5.6 Conclusion

The consensus problem for second order saturated multi-agent system with asynchronous communication and input time-delays is presented in this chapter. An approximate system with separate linear and nonlinear elements is derived using describing function analysis to study the limit cycle behaviour. The instability of limit cycles or consensus reachability is estimated using describing function analysis. Lyapunov-Krasovskii function is used to prove the stability of linear element. Stable ranges of input and communication time-delays are calculated for different control laws and comparative analysis of the results are presented. Justification to the theoretical results is done with the help of simulations and corresponding implementations on hardware. Time-varying delays are not considered in the current analysis, they shall be considered in the future work. Further, a multi-agent system with heterogeneous agents is studied. A general nonlinear input is considered for the analysis. It is proved that, a general nonlinear odd function symmetric to origin is able to solve the consensus problem of an undirected network without time-delays. Simulation and implementation results are presented to demonstrate the

effectiveness of the theoretical results.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusions

In this thesis, two new algorithms are proposed in chapter 3, to overcome the complete loss of communication while reaching consensus. The control input of each agent is computed using the proposed algorithms. A number of static obstacles are assumed to be present in the arena. A sensor based obstacle avoidance algorithm ensures no collision between the agents and the obstacles. The switching occurs depending on the positions of the agents. The algorithms guarantee consensus by letting the agents to either back-track and take a different path or follow the previous consensus point whenever a loss of communication takes place. Simulation results show that when the agents back-track and take another path whenever they lose contact with the network, consensus is reached all the time. However, when they follow the previous history, situations may arise where some of the agents stop at locations near the actual consensus point. Otherwise, the time to reach consensus is less in case of history following with memory enabled agents. Hardware implementation without obstacles is also presented which shows the usefulness of the algorithms for practical purposes.

In chapters 4 and 5, the consensus problem for second order saturated multi-agent system with asynchronous communication and input time-delays is studied. The system is rearranged into linear and nonlinear elements using describing function analysis to study the limit cycle behaviour. The instability of limit cycles or consensus reachability is estimated using describing functions and stability of linear element. The approaches used in the thesis reduce the mathematical complexity of analysis of the nonlinear system. The stability analysis of linear element

is performed with the help of Nyquist stability criterion and Lyapunov-Krasovskii function. Stable ranges of input and communication time-delays are calculated for different control laws using both the approaches and comparative analysis are presented. Justification to the theoretical results is done with the help of simulations and corresponding implementations on hardware. In chapter 5, a linear system with a general nonlinear input is also analysed. It has been observed that a nonlinear odd function $g(x)$ symmetric to origin and $xg(x) > 0$ is able to solve the consensus of a multi-agent system connected by an undirected network. The theoretical results are obtained by Lyapunov approach. Simulation and implementation are performed on a similar network used earlier in chapters 4 and 5.

6.2 Future Scope and Possible Directions

With current control laws, the system is not immune to external disturbances in the state information. Noise in the state information and its mitigation strategies have to be considered in the future research. Moreover, stability of heterogeneous agents using advanced control strategies [118] and nonlinear inputs are receiving attention recently. Most of the practical systems are nonlinear and heterogeneous in nature. For example, supply and load characteristics of various substations in a smart grid are nonlinear and heterogeneous. Developing control strategies for such systems is very much required in the current scenario of smart world. Time-delays are not yet considered in such systems. The system of heterogeneous agents is already complex with nonlinearities and it becomes more complex if time-delays are introduced.

The task allocation to a multi-agent system is manual and human intervention is needed. If the agents are intelligent and can analyse the goal, the coordination among the agents becomes more complex. The concepts of game theory, competition and coordination act simultaneously. The outcome of such intelligent multi-agent system is yet to be explored. For game theory and competition to work, a reward system has to be in place. Implementation of reward system in a distributed system with common goal is also tricky. All these aspects are not explored with respect to multi-agent systems.

APPENDIX A

SUPPLEMENTARY MATERIALS

A.1 Describing Function Analysis

Frequency domain analysis is a very powerful method for the analysis and design of linear time-invariant control systems. It is based on representing a system with complex valued algebraic functions called transfer functions instead of differential equations. However, frequency domain analysis cannot be directly applied on nonlinear systems since the transfer functions cannot be defined for nonlinear systems. Nevertheless, describing function analysis which is an extended version of the frequency domain analysis can be used to approximately analyze the nonlinear systems. Although the describing function analysis is an approximation, the major use of it is to estimate the existence of limit cycles in nonlinear systems.

Limit cycles in most of the control systems are undesirable (few exceptions like electronic oscillators need limit cycles in the output), so prediction of limit cycles is of prime importance in nonlinear systems. The basic assumptions to use describing function analysis are: a) system has single, time-invariant nonlinearity; b) the nonlinearity is odd; c) only fundamental component is considered for a sinusoidal input. Consider a nonlinear system in Fig. A.1 with saturation nonlinearity, slope of the saturation function in linear region is assumed to be unity.

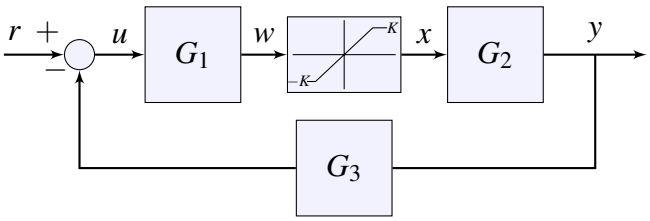


Figure A.1: Block diagram of nonlinear system.

The system in Fig. A.1 can be rearranged into an approximate system, as depicted in Fig. A.2 [119]with linear element in as $G = G_1G_2G_3$.

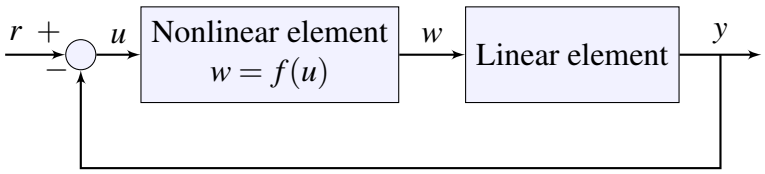


Figure A.2: Block diagram of rearranged system.

For a sinusoidal input, the output of saturation function with unity slope in the linear region is given by Eqn. (A.1).

$$w = \begin{cases} -K & -K > A\sin(\omega t) \\ A\sin(\omega t) & -K < A\sin(\omega t) < K \\ K & K > A\sin(\omega t) \end{cases} \tag{A.1}$$

The describing function of saturation nonlinearity in Eqn. (A.2) is obtained by following aforementioned assumptions [119]. Only fundamental component of the Fourier series is considered.

$$N(A) = \frac{2}{\pi}\left[\arcsin\left(\frac{K}{A}\right) + \frac{K}{A}\sqrt{1-\frac{K^2}{A^2}}\right] \tag{A.2}$$

where, the limit cycles' amplitude is represented by A.

For sustained oscillations with no external input ($r = 0$), the following relations obtained from Fig. A.2 must be satisfied.

$$\begin{aligned} u &= -y \\ w &= N(A)u \\ y &= Gw \end{aligned} \tag{A.3}$$

After rearranging Eqn. (A.3),

$$GN(A) + 1 = 0 \tag{A.4}$$

If Eqn. (A.3) has no solutions, it can be estimated that the system has no sustained oscil-

lations. The nonlinear system exhibits limit cycle behavior when G and $-1/N(A)$ intersect with each other in complex plane. The describing function in Eqn. (A.2) is real valued and $-1/N(A) \in [-1, \infty)$. The limit cycles are stable when the transfer function of linear element G encircles $(-1, 0)$ in a complex plane. In other words, limit cycles are exhibited when the linear element is unstable. Stability analysis of the linear element can be performed using Nyquist stability approach and Lyapunov-Krasovskii approach.

A.2 Lyapunov Stability Theory

In this section, Lyapunov stability theory that is used in the subsequent chapters has been reviewed [119]. Some basic definitions are discussed, Lyapunov stability theory for nonlinear and linear systems are explained.

Consider a set of ordinary differential equations given in Eqn. (A.5), which is widely used to represent unforced nonlinear systems.

$$\dot{\mathbf{x}} = f(\mathbf{x}) \tag{A.5}$$

Here, $\mathbf{x} \in \Re^n$ are called state variables and the evolution of state variables with respect to time is characterized by the differential equation. A basic assumption on the system is such that, future values of the state variables are determined by the current values of the state variables. In other words, if the value of $\mathbf{x}$ at $t = t_0$ is known, then the value of $\mathbf{x}$, $\forall t > t_0$ can be determined.

Definition 8. *Equilibrium point*: A state $\mathbf{x}^*$ is called equilibrium point of the system if once $\mathbf{x}$ reaches $\mathbf{x}^*$, it remains at $\mathbf{x}^*$ for all future time.

The equilibrium points are found by solving $\dot{\mathbf{x}} = 0$

A.2.1 Stability, Asymptotic Stability and Exponential Stability

The study of stability, asymptotic stability and exponential stability in the sense of Lyapunov is performed using the evolution of state variables with respect to time.

Definition 9. For the system in Eqn. (A.5), the equilibrium point $\mathbf{x}^* = 0$ is said to be *stable* if, $\forall \Delta > 0$, there exits some $\delta > 0$, such that if $\|\mathbf{x}(t_0)\| < \delta$, then $\|\mathbf{x}(t)\| < \Delta$, $\forall t \geq t_0$, where $\|\cdot\|$ denotes second norm.

Stability of a system according to Lyapunov theory signify that the state variables can be kept close to origin for all $t > t_0$ by starting close to origin at t_0. The stability guarantees that the states lie in a ball B_Δ centered at origin of radius Δ for all the future time, after starting inside another ball B_δ centered at origin of radius δ at time t_0. In many applications, stability is not enough and there is a need for the states to converge at origin as $t \to \infty$. In some applications, convergence to origin at infinite time is also not sufficient and rate of convergence is required to determine how fast the states converge.

Definition 10. The equilibrium point $\mathbf{x}^* = 0$ is said to be *asymptotically stable* if, it is stable and $\exists \delta > 0$ such that, $\|\mathbf{x}(t_0)\| < \delta \Rightarrow \|\mathbf{x}(t)\| \to \mathbf{0}$ as $t \to 0$.

Definition 11. The equilibrium point $\mathbf{x}^* = 0$ is said to be *exponentially stable* if, there exists positive constants α and β such that, $\|\mathbf{x}(t_0)\| < \delta \Rightarrow \ \|\mathbf{x}(t)\| \leq \alpha \|\mathbf{x}(t_0)\| e^{-\beta(t-t_0)}$, $\forall t > t_0$.

Asymptotic stability implies that, within a ball B_δ of radius δ ($\mathbf{x} \in B_\delta$) the equilibrium point is stable and also the state variables converge to origin as time tends to infinity. Exponential stability implies that the system is asymptotically stable and the states converge to origin with an exponential convergence rate of β. The asymptotic stability and exponential stability defined above are local and valid for states within ball B_δ (also called as domain of attraction).

Definition 12. If asymptotic (or exponential) stability holds for any initial states, then the equilibrium point is said to be *globally* asymptotically (or exponentially) stable.

The global asymptotic (or exponential) stability has no restriction on domain of attraction, δ can be infinitely large. The states converge to origin irrespective of the initial values if the system is globally asymptotically (or exponentially) stable.

A.2.2 Lyapunov's Direct Method

Lyapunov's direct method also known as second method of Lyapunov is based on energy like functions called Lyapunov functions.

Definition 13. A scalar function $V(\mathbf{x})$ is *locally positive definite* if $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) > 0$, $\forall \mathbf{x} \neq 0, \mathbf{x} \in B_\delta$. It is *locally positive semidefinite* if $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) \geq 0$, $\forall \mathbf{x} \neq 0, \mathbf{x} \in B_\delta$. Similarly, *locally negative definite* and *locally negative semidefinite* cases are defined for $V(\mathbf{x}) < 0$ and $V(\mathbf{x}) \leq 0$ respectively .

Definition 14. A scalar function $V(\mathbf{x})$ is *positive definite* if $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) > 0$, $\forall \mathbf{x} \neq 0$. It is *positive semidefinite* if $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) \geq 0$, $\forall \mathbf{x} \neq 0$. Similarly, *negative definite* and *negative semidefinite* cases are defined for $V(\mathbf{x}) < 0$ and $V(\mathbf{x}) \leq 0$ respectively.

In general, energy functions are positive definite. In mechanical systems, when the energy is continuously dissipated, the system eventually converges to equilibrium point. For the energy function to be continuously decreasing and reach origin, its time derivative has to be negative definite. By choosing proper Lyapunov functions, the analogy of energy functions can be extended to study complex systems. Convergence of the Lyapunov functions to origin implies asymptotic stability and divergence implies instability. If the Lyapunov function is neither converging nor diverging, but is bounded in some ball, then the system is stable.

Definition 15. A function $V(\mathbf{x})$ with continuous partial derivative is called *Lyapunov function* if, it is positive definite $(V(\mathbf{x}) > 0)$ and its time derivative is at least negative semidefinite $(\dot{V}(\mathbf{x}) \leq 0)$.

Theorem 5. *If there exists a Lyapunov function $V(\mathbf{x})$, then the equilibrium point at origin is,*

1. *stable if $V(\mathbf{x})$ is locally positive definite, $\dot{V}(\mathbf{x})$ is locally negative semidefinite*
2. *asymptotically stable if $V(\mathbf{x})$ is locally positive definite, $\dot{V}(\mathbf{x})$ is locally negative definite*
3. *globally asymptotically stable if $V(\mathbf{x})$ is positive definite and radially unbounded $(V(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty)$, $\dot{V}(\mathbf{x})$ is negative definite.*

A.2.3 Linear Time-invariant Systems

An unforced linear time-invariant system can be represented by Eqn. (A.6),

$$\dot{\mathbf{x}} = A\mathbf{x} \tag{A.6}$$

The Lyapunov function in general is considered to be quadratic, which includes all the state variables. Let P be a symmetric positive definite matrix of appropriate dimension, then the Lyapunov function can be of the form given in Eqn. (A.7).

$$V = \mathbf{x}^T P\mathbf{x} \tag{A.7}$$

The time derivative of V yields,

$$\dot{V} = \mathbf{x}^T(A^T P + PA)\mathbf{x} \tag{A.8}$$

For the system in Eqn. (A.6) to be considered as globally asymptotically stable, $A^T P + PA = -Q$ should be negative definite or Q be positive definite. A necessary and sufficient condition for the system in Eqn. (A.6) to be asymptotically stable is that, there exists a unique symmetric positive definite P matrix for any symmetric positive definite Q matrix. It can also be expressed as a linear matrix inequality (LMI), $A^T P + PA < 0$ for asymptotic stability. LMIs can be solved using solvers like SeDuMi [115], Matlab LMI Lab solver etc., to find the existence of P. Some solvers like SeDuMi will not handle strict inequalities, in that case the LMI can be modified as $A^T P + PA + Q \leq 0$. The solver will iteratively find the existence of positive definite P and Q in a fairly efficient and effective manner.

A.3 Characteristic Equation

An n^{th} order system with l number of inputs is defined in Eqn. (A.9).

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{A.9}$$

Where $\mathbf{x} \in \Re^{n\times 1}$, $A \in \Re^{n\times n}$, $B \in \Re^{n\times l}$ and $\mathbf{u} \in \Re^{l\times 1}$. The characteristic of Eqn. (A.9) system in equation is as given in Eqn. (A.10).

$$|sI - A| = 0 \tag{A.10}$$

After expanding the Eqn. (A.10),

$$a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} + ... + a_{n-1} s + a_n = 0 \tag{A.11}$$

Let $\{\lambda_1, \lambda_2, ...\lambda_n\}$ be roots of Eqn. (A.11), it can be rewritten as,

$$(s-\lambda_1)(s-\lambda_2)....(s-\lambda_n) = 0 \tag{A.12}$$

In certain systems, each x_i can be a subsystem. In such cases, instead $|sI - A|$, the characteristic equation becomes,

$$|f(s)I - A| = 0 \tag{A.13}$$

By following a similar procedure as described above,

$$|f(s)I - A| = (f(s)-\lambda_1)(f(s)-\lambda_2)....(f(s)-\lambda_n) = \prod_{i=1}^{n}(f(s)-\lambda_i) \tag{A.14}$$

A.4 Patrolbot

A.4.1 Specifications

Table. A.1: Physical Characteristics and Power

Length	59 cm
Width	48 cm
Height (body)	38 cm
Body clearance	5 cm
Weight	45 Kg
Payload	12 Kg
Battery	2x12V lead-acid
Charge	480 wt-hr
Run time	4-6 hrs

Figure A.3: Patrolbot

Table. A.2: Physical Characteristics

Drive wheels	2 foam-filled rubber, with 2 rear-caster balance
Wheel diameter	19 cm
Steering	Differential
Gear ratio	28.9:1
Pushing force	22 kg
Swing radius	29 cm
Turn radius	0 cm
Translate speed max	2000 mm/sec
Rotational speed max	360 degrees/sec
Traversable step max	4 cm
Traversable gap max	8 cm
Max climb grade	20 %
Sonar	4 rear
Position encoders	2 (one each motor); 30,000 ticks per wheel revolution; 195 ticks per mm
Processor	44 MHz Renesas SH2-7144
Position inputs	4
Data bus	32-bit
Comm ports	4 x RS-232
FLASH	128 KB
RAM	32 KB

A.5 Assembled Robots

Motherboard

Beaglebone Black with 1GHz ARM cortex A-8 processor

Two Programmable real time units (PRUs) 32-bit 200MHz micro-controllers

512MB ram clocking at 800MHz

2GB flash memory that can run linux OS

USB host, HDMI port with 3D graphics accelerator

2×46 GPIOs to control external devices

Motors and drivers

Motors: 100 rpm metal geared motors with 36 kg-cm stall torque at limited 4A maximum current.

Drivers: Dual DC motor driver that can handle 5A starting current on each channel.

Communication

Tp-Link 721WN Wi-Fi adapter with atheros chipset inserted into USB host of motherboard.

Sensors

Wheel encoders: MOC7811 based on LED and phototransistor pair along with wheel encoder rings are used to measure the rotation made by wheels.

Power

Lead acid battery: 12V, 7Ah

The image of assembled robot is shown in the figure A.4.

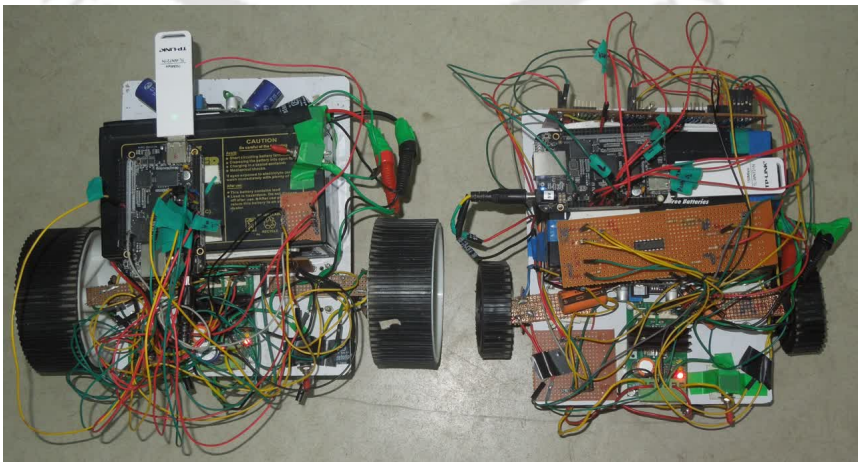


Figure A.4: Assembled robots

LIST OF PUBLICATIONS

Journal Publications:

Published:

1. Y. V. Karteek, Indrani Kar and S. Majhi, "Back-Tracking and History Following of Agents to Reach Consensus in a Network of Agents with Static Obstacles", *International Journal of Robotics and Automation*, vol. 32(4), 2017.
2. Y. V. Karteek, Indrani Kar and S. Majhi, "Consensus of second order multi-agents with actuator saturation and asynchronous time-delays", *IET Control Theory Applications*, vol. 11, no. 17, pp. 3201 – 3210, 2017.

Communicated:

3. Y. V. Karteek, Indrani Kar and S. Majhi, "Consensus of heterogeneous second order multi-agents with general nonlinear input", communicated to *International Journal of Dynamics and Control*.

Conference Papers:

1. Yanumula Venkata Karteek and Indrani Kar, "Consensus of Multi-agent Systems with Back-stepping for Switching Topologies", *IEEE Workshop on Computational Intelligence, IIT Kanpur, India.*July 14, 2013, pp. 143-147.

2. Y. V. Karteek, I. Kar and S. Majhi, "Consensus in multi-agent systems with switching topologies using position based back-stepping", in *2014 Annual IEEE India Conference (INDICON), Pune*, Dec 11-13 , 2014, pp. 1-5.

3. Y. V. Karteek, Indrani Kar and S. Majhi, "Consensus of distributed second order multi-agent system with saturation and uneven time-delays", *Advances in Control & Optimization of Dynamical Systems (ACODS) - 2018*, Feb 18-22.

REFERENCES

[1] Shaolei Zhou and Shi Yan. The stability analysis for a class of Multi-Agent group formation with input saturation constraints. In *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, pages 1618–1623, 2014.

[2] Z. Zhiqiang, Z. Zhicheng, and W. Yijing. Finite-time consensus of double integrator multi-agent systems subject to input saturation. In *Control Conference (CCC), 2015 34th Chinese*, pages 7280–7284, July 2015.

[3] V. Julian and V. Botti. Developing real-time multi-agent systems. *Integr. Comput.-Aided Eng.*, 11(2):135–149, April 2004.

[4] Sheng-Luen Chung and Muder Jeng. Fabulous mess and c/cs: an overview of semiconductor fab automation systems. *IEEE Robotics Automation Magazine*, 11(1):8–18, Mar 2004.

[5] P. Deotare and L. Dole. Overview of automation of smart grid network. In *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–3, Jan 2015.

[6] X. Zhou, P. Xiang, Y. Ma, Z. Gao, Y. Wu, J. Yin, and X. Xu. An overview on distribution automation system. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 3667–3671, May 2016.

[7] H. T. Roman, B. A. Pellegrino, and W. R. Sigrist. Pipe crawling inspection robots: an overview. *IEEE Transactions on Energy Conversion*, 8(3):576–583, Sep 1993.

[8] M. Eslahi, R. Salleh, and N. B. Anuar. Bots and botnets: An overview of characteristics, detection and challenges. In *2012 IEEE International Conference on Control System, Computing and Engineering*, pages 349–354, Nov 2012.

[9] J. Burgner-Kahrs, D. C. Rucker, and H. Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, Dec 2015.

[10] W. Huo, S. Mohammed, J. C. Moreno, and Y. Amirat. Lower limb wearable robots for assistance and rehabilitation: A state of the art. *IEEE Systems Journal*, 10(3):1068–1081, Sept 2016.

[11] U. Keller, H. J. A. van Hedel, V. Klamroth-Marganska, and R. Riener. Charmin: The first actuated exoskeleton robot for pediatric arm rehabilitation. *IEEE/ASME Transactions on Mechatronics*, 21(5):2201–2213, Oct 2016.

[12] J. Konstantinova, A. Jiang, K. Althoefer, P. Dasgupta, and T. Nanayakkara. Implementation of tactile sensing for palpation in robot-assisted minimally invasive surgery: A review. *IEEE Sensors Journal*, 14(8):2490–2501, Aug 2014.

[13] M. Y. Jung, R. H. Taylor, and P. Kazanzides. Safety design view: A conceptual framework for systematic understanding of safety features of medical robot systems. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1883–1888, May 2014.

[14] C. Faria, W. Erlhagen, M. Rito, E. De Momi, G. Ferrigno, and E. Bicho. Review of robotic technology for stereotactic neurosurgery. *IEEE Reviews in Biomedical Engineering*, 8:125–137, 2015.

[15] N. Enayati, E. De Momi, and G. Ferrigno. Haptics in robot-assisted surgery: Challenges and benefits. *IEEE Reviews in Biomedical Engineering*, 9:49–65, 2016.

[16] John D. Lee. Driving safety. *Reviews of Human Factors and Ergonomics*, 1(1):172–218, 2005.

[17] M. Suresh and D. Ghose. Uav grouping and coordination tactics for ground attack missions. *IEEE Transactions on Aerospace and Electronic Systems*, 48(1):673–692, Jan 2012.

[18] L. Jones. Driverless when and cars: where? [automotive autonomous vehicles]. *Engineering Technology*, 12(2):36–40, March 2017.

[19] R. Murray. Driverless cars. *Control Automation*, 18(3):14–17, June 2007.

[20] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, D. Molinari, M. Panciroli, and A. Prioletti. Proud – public road urban driverless-car test. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3508–3519, Dec 2015.

[21] W. B. Rouse. The systems, man, and cybernetics of driverless cars: Challenges and opportunities for the smcs. *IEEE Systems, Man, and Cybernetics Magazine*, 3(3):6–8, July 2017.

[22] M. Liu, G. K. Egan, and F. Santoso. Modeling, autopilot design, and field tuning of a uav with minimum control surfaces. *IEEE Transactions on Control Systems Technology*, 23(6):2353–2360, Nov 2015.

[23] S. Das and S. E. Talole. Robust steering autopilot design for marine surface vessels. *IEEE Journal of Oceanic Engineering*, 41(4):913–922, Oct 2016.

[24] S. Knorn, Z. Chen, and R. H. Middleton. Overview: Collective control of multiagent systems. *IEEE Transactions on Control of Network Systems*, 3(4):334–347, Dec 2016.

[25] C. M. Gifford, R. Webb, J. Bley, D. Leung, M. Calnon, J. Makarewicz, B. Banz, and A. Agah. Low-cost multi-robot exploration and mapping. In *2008 IEEE International Conference on Technologies for Practical Robot Applications*, pages 74–79, Nov 2008.

[26] R. Reid and T. Brunl. Large-scale multi-robot mapping in magic 2010. In *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*, pages 239–244, Sept 2011.

[27] L. Vachhani, A. D. Mahindrakar, and K. Sridharan. Mobile robot navigation through a hardware-efficient implementation for control-law-based construction of generalized voronoi diagram. *IEEE/ASME Transactions on Mechatronics*, 16(6):1083–1095, Dec 2011.

[28] Soumic Sarkar and Indra Narayan Kar. Formation of multiple groups of mobile robots: multi-timescale convergence perspective. *Nonlinear Dynamics*, 85(4):2611–2627, Sep 2016.

[29] Sarat Chandra Nagavarapu, Leena Vachhani, and Arpita Sinha. Multi-robot graph exploration and map building with collision avoidance: A decentralized approach. *Journal of Intelligent & Robotic Systems*, 83(3):503–523, Sep 2016.

[30] J. Cohen, L. Matignon, and O. Simonin. Incremental and adaptive multi-robot mapping for human scene observation. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 678–685, Nov 2016.

[31] 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*, 9(1):427–438, Feb 2013.

[32] Sai-Ming Li, J. D. Boskovic, S. Seereeram, R. Prasanth, J. Amin, R. K. Mehra, R. W. Beard, and T. W. McLain. Autonomous hierarchical control of multiple unmanned combat air vehicles (ucavs). In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 1, pages 274–279 vol.1, May 2002.

[33] F. Giulietti, L. Pollini, and M. Innocenti. Autonomous formation flight. *IEEE Control Systems*, 20(6):34–44, Dec 2000.

[34] Thomas B. Curtin, James G. Bellingham, Josko Catipovic, and Doug Webb. Autonomous oceanographic sampling networks. *Oceanography*, 6, 1993.

[35] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. *IEEE Journal of Oceanic Engineering*, 31(4):935–948, Oct 2006.

[36] Joel George Manathara and Debasish Ghose. Rendezvous of multiple uavs with collision avoidance using consensus. *Journal of Aerospace Engineering*, 25(4):480–489, 2012.

[37] K. Lau, S. Lichten, L. Young, and B. Haines. An innovative deep space application of GPS technology of formation flying spacecraft. In *San Diego, CA: AIAA Guidance, Navigation and Control Conference*, 1996.

[38] V. Kapilal, A. G. Sparks, J. M. Buffington, and Qiguo Yan. Spacecraft formation flying: dynamics and control. In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, volume 6, pages 4137–4141 vol.6, 1999.

[39] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, Nov 2001.

[40] Soumya Ranjan Sahoo, Ravi N. Banavar, and Arpita Sinha. Rendezvous in space with minimal sensing and coarse actuation. *Automatica*, 49(2):519 – 525, 2013.

[41] F. Paganini, Zhikui Wang, J. C. Doyle, and S. H. Low. Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Transactions on Networking*, 13(1):43–56, Feb 2005.

[42] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pages 263–270, New York, NY, USA, 1999. ACM.

[43] J. A. Stankovic, T. E. Abdelzaher, Chenyang Lu, Lui Sha, and J. C. Hou. Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE*, 91(7):1002–1022, July 2003.

[44] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 63–70, April 2005.

[45] Leigh Tesfatsion. Agent-based computational economics: Growing economies from the bottom up. *Artificial life*, 8(1):55–82, 2002.

[46] Leigh Tesfatsion. Agent-based computational economics: modeling economies as complex adaptive systems. *Information Sciences*, 149(4):262–268, 2003.

[47] W Brian Arthur. Out-of-equilibrium economics and agent-based modeling. *Handbook of computational economics*, 2:1551–1564, 2006.

[48] Jafar Ghaisari and Arash Ferdosi. A direct sequence spread spectrum code acquisition circuit for wireless sensor networks. *International Journal of Electronics*, 98(6):793800, Jun 2011. dummy note.

[49] Alessio Lomuscio, Hongyang Qu, and Monika Solanki. Towards verifying contract regulated service composition. *Autonomous Agents and Multi-Agent Systems*, 24(3):345–373, May 2012.

[50] Panagiotis Kouvaros and Alessio Lomuscio. Parameterised verification for multi-agent systems. *Artificial Intelligence*, 234:152 – 189, 2016.

[51] Maryam Fattahi and Ahmad Afshar. Distributed consensus of multi-agent systems with fault in transmission of control input and time-varying delays. *Neurocomputing*, 189:11 – 24, 2016.

[52] Fahimeh Kazempour and Jafar Ghaisari. Stability analysis of model-based networked distributed control systems. *Journal of Process Control*, 23(3):444 – 452, 2013.

[53] Hongjie Li and Hongye Su. Second-order consensus in multi-agent systems with directed topologies and communication constraints. *Neurocomputing*, 173, Part 3:942 – 952, 2016.

[54] Pingsong Ming, Jianchang Liu, Shubin Tan, Songhua Li, Liangliang Shang, and Xia Yu. Consensus stabilization in stochastic multi-agent systems with markovian switching topology, noises and delay. *Neurocomputing*, 200:1 – 10, 2016.

[55] Deqiang Ouyang, Zhiyong Yu, Haijun Jiang, and Cheng Hu. Consensus for general multi-agent networks with external disturbances. *Neurocomputing*, 198:100 – 108, 2016. Advances in Neural Networks, Intelligent Control and Information ProcessingContaining a selection of papers from the 5th International Conference on Intelligent Control and Information Processing (ICICIP2014).

[56] Zhongmei Wang and Huanshui Zhang. Observer-based robust consensus control for multi-agent systems with noises. *Neurocomputing*, 207:408 – 415, 2016.

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

[58] Fei Liu Cheng-lin Liu. Consensus problem of second-order multi-agent systems with input delay. *International Journal of Information and Systems Sciences*, 7(2):175–191, 2011.

[59] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, Aug 1995.

[60] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada. The evolution of robotics research. *IEEE Robotics Automation Magazine*, 14(1):90–103, March 2007.

[61] A. Jadbabaie, Jie Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.

[62] R. O. Saber and R. M. Murray. Consensus protocols for networks of dynamic agents. In *American Control Conference, 2003. Proceedings of the 2003*, volume 2, pages 951–956, June 2003.

[63] L. Moreau. Stability of continuous-time distributed consensus algorithms. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pages 3998–4003 Vol.4, Dec 2004.

[64] Guangming Xie and Long Wang. Consensus control for a class of networks of dynamic agents: Fixed topology. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 96–101, Dec 2005.

[65] Guangming Xie and Long Wang. Consensus control for a class of networks of dynamic agents: switching topology. In *2006 American Control Conference*, pages 6 pp.–, June 2006.

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

[67] Wei Ren, K. Moore, and YangQuan Chen. High-order consensus algorithms in cooperative vehicle systems. In *2006 IEEE International Conference on Networking, Sensing and Control*, pages 457–462, 2006.

[68] W. Ren. On consensus algorithms for double-integrator dynamics. *IEEE Transactions on Automatic Control*, 53(6):1503–1509, July 2008.

[69] P. A. Bliman, A. Nedic, and A. Ozdaglar. Rate of convergence for consensus with delays. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4849–4854, Dec 2008.

[70] Ming Cao, A. Stephen Morse, and Brian D. O. Anderson. Reaching a consensus in a dynamically changing environment: Convergence rates, measurement delays, and asynchronous events. *SIAM Journal on Control and Optimization*, 47(2):601–623, 2008.

[71] F. Xiao and L. Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control*, 53(8):1804–1816, Sept 2008.

[72] D. J. Liu and C. L. Liu. Consensus problem of discrete-time second-order multi-agent network with communication delays. In *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, volume 2, pages 340–344, Nov 2009.

[73] S. Liu, L. Xie, and T. Li. Distributed consensus for multi-agent systems with communication delays and limited data rate. In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pages 904–909, July 2010.

[74] J. Hu and Y. S. Lin. Consensus control for multi-agent systems with double-integrator dynamics and time delays. *IET Control Theory Applications*, 4(1):109–118, Jan 2010.

[75] H. Liu, G. Xie, and L. Wang. Consensus of multi-agent systems with time-varying delay. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3078–3083, Dec 2010.

[76] T. Shida and H. Ohmori. Robustness of consensus algorithm for communication delays and switching topology. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 1373–1380, Sept 2011.

[77] Z. Meng, W. Ren, Y. Cao, and Z. You. Leaderless and leader-following consensus with communication and input delays under a directed network topology. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):75–88, Feb 2011.

[78] W. He and J. Cao. Consensus control for high-order multi-agent systems [brief paper]. *IET Control Theory Applications*, 5(1):231–238, January 2011.

[79] P. Lin, Z. Li, Y. Jia, and M. Sun. High-order multi-agent consensus with dynamically changing topologies and time-delays. *IET Control Theory Applications*, 5(8):976–981, May 2011.

[80] Jun Shen and Jinde Cao. Necessary and sufficient conditions for consensus of delayed fractional-order systems. *Asian Journal of Control*, 14(6):1690–1697, 2012.

[81] Guojian Ren and Yongguang Yu. Robust consensus of fractional multi-agent systems with external disturbances. *Neurocomputing*, pages –, 2016.

[82] Wenguang Zhang, Jizhen Liu, Deliang Zeng, and Tingting Yang. Consensus analysis of continuous-time second-order multi-agent systems with nonuniform time-delays and switching topologies. *Asian Journal of Control*, 15(5):1516–1523, 2013.

[83] Peng Lin, Mingxiang Dai, and Yongduan Song. Consensus stability of a class of second-order multi-agent systems with nonuniform time-delays. *Journal of the Franklin Institute*, 351(3):1571 – 1576, 2014.

[84] Yuan Fan and Jianyu Yang. Average consensus of multi-agent systems with self-triggered controllers. *Neurocomputing*, 177:33 – 39, 2016.

[85] Huaqing Li, Guo Chen, Tingwen Huang, Wei Zhu, and Li Xiao. Event-triggered consensus in nonlinear multi-agent systems with nonlinear dynamics and directed network topology. *Neurocomputing*, 185:105 – 112, 2016.

[86] Fangcui Jiang, Dongmei Xie, and Bo Liu. Static consensus of second-order multi-agent systems with impulsive algorithm and time-delays. *Neurocomputing*, pages –, 2016.

[87] Chang-E Ren, Long Chen, C.L. Philip Chen, and Tao Du. Quantized consensus control for second-order multi-agent systems with nonlinear dynamics. *Neurocomputing*, 175, Part A:529 – 537, 2016.

[88] Gang Wang, Chaoli Wang, Lin Li, and Qinghui Du. Distributed adaptive consensus tracking control of higher-order nonlinear strict-feedback multi-agent systems using neural networks. *Neurocomputing*, 214:269 – 279, 2016.

[89] Xiangyu Meng, Ziyang Meng, Tongwen Chen, Dimos V. Dimarogonas, and Karl Henrik Johansson. Pulse width modulation for multi-agent systems. *Automatica*, 70:173 – 178, 2016.

[90] Y. Li, J. Xiang, and W. Wei. Consensus problems for linear time-invariant multi-agent systems with saturation constraints. *IET Control Theory Applications*, 5(6):823–829, April 2011.

[91] Ziyang Meng, Zhiyun Zhao, and Zongli Lin. On global leader-following consensus of identical linear dynamic systems subject to actuator saturation. *Systems & Control Letters*, 62(2):132 – 142, 2013.

[92] A. Wei, X. Hu, and Y. Wang. Tracking control of leader-follower multi-agent systems subject to actuator saturation. *IEEE/CAA Journal of Automatica Sinica*, 1(1):84–91, Jan 2014.

[93] H. Chu, J. Yuan, and W. Zhang. Observer-based adaptive consensus tracking for linear multi-agent systems with input saturation. *IET Control Theory Applications*, 9(14):2124–2131, 2015.

[94] H. Su and M. Z. Q. Chen. Multi-agent containment control with input saturation on switching topologies. *IET Control Theory Applications*, 9(3):399–409, 2015.

[95] G. Cui, S. Xu, F. L. Lewis, B. Zhang, and Q. Ma. Distributed consensus tracking for non-linear multi-agent systems with input saturation: a command filtered backstepping approach. *IET Control Theory Applications*, 10(5):509–516, 2016.

[96] X. You, C. Hua, D. Peng, and X. Guan. Leader following consensus for multi-agent systems subject to actuator saturation with switching topologies and time-varying delays. *IET Control Theory Applications*, 10(2):144–150, 2016.

[97] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J. L. Deneuborg, and M. Dorigo. The cooperation of swarm-bots: physical interactions in collective robotics. *IEEE Robotics Automation Magazine*, 12(2):21–28, June 2005.

[98] N. Correll and A. Martinoli. Multirobot inspection of industrial machinery. *IEEE Robotics Automation Magazine*, 16(1):103–112, March 2009.

[99] Y. H. Chang, C. W. Chang, C. L. Chen, and C. W. Tao. Fuzzy sliding-mode formation control for multirobot systems: Design and implementation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):444–457, April 2012.

[100] C. Churavy, M. Baker, S. Mehta, I. Pradhan, N. Scheidegger, S. Shanfelt, R. Rarick, and D. Simon. Effective implementation of a mapping swarm of robots. *IEEE Potentials*, 27(4):28–33, July 2008.

[101] M. Pinto, A. P. Moreira, and A. Matos. Localization of mobile robots using an extended kalman filter in a lego nxt. *IEEE Transactions on Education*, 55(1):135–144, Feb 2012.

[102] Yuanzhen Feng, Shengyuan Xu, Frank L. Lewis, and Baoyong Zhang. Consensus of heterogeneous first- and second-order multi-agent systems with directed communication topologies. *International Journal of Robust and Nonlinear Control*, 25(3):362–375, 2015.

[103] Frank L. Lewis, Hongwei Zhang, Kristian Hengster-Movric, and Abhijit Das. *Algebraic Graph Theory and Cooperative Control Consensus*, pages 23–71. Springer London, London, 2014.

[104] C. R. Johnson. Positive definite matrices. *The American Mathematical Monthly*, 77(3):259–264, 1970.

[105] Charles R. Johnson and Robert Reams. Semidefiniteness without real symmetry. *Linear Algebra and its Applications*, 306(1):203 – 209, 2000.

[106] Xiangke Wang, Zhiwen Zeng, and Yirui Cong. Multi-agent distributed coordination control: Developments and directions via graph viewpoint. *Neurocomputing*, 199:204 – 218, 2016.

[107] Ulrich Münz, Antonis Papachristodoulou, and Frank Allgöwer. Delay robustness in consensus problems. *Automatica*, 46(8):1252 – 1265, 2010.

[108] Jean-Jacques E. Slotine and Weiping Li. *Applied nonlinear control*, chapter 5, Describing Function Analysis, pages 157–190. Prentice Hall, Englewood Cliffs (N.J.), 1991.

[109] Jie Chen Keqin Gu, Vladimir L. Kharitonov. *Stability of Time-Delay Systems*. Birkhäuser Boston, 2003.

[110] V.L. Kharitonov and A.P. Zhabko. Lyapunovkrasovskii approach to the robust stability analysis of time-delay systems. *Automatica*, 39(1):15 – 20, 2003.

[111] Vladimir L Kharitonov and S-I Niculescu. On the stability of linear systems with uncertain delay. *IEEE Transactions on Automatic Control*, 48(1):127–132, 2003.

[112] Hongwei Zhang, Zhongkui Li, Zhihua Qu, and Frank L. Lewis. On constructing lyapunov functions for multi-agent systems. *Automatica*, 58:39 – 42, 2015.

[113] P. Lin and Y. Jia. Consensus of a class of second-order multi-agent systems with time-delay and jointly-connected topologies. *IEEE Transactions on Automatic Control*, 55(3):778–784, March 2010.

[114] Peng Lin and Yingmin Jia. Average consensus in networks of multi-agents with both switching topology and coupling time-delay. *Physica A: Statistical Mechanics and its Applications*, 387(1):303 – 313, 2008.

[115] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.05 available from `http://fewcal.kub.nl/sturm`.

[116] Kaien Liu, Guangming Xie, Wei Ren, and Long Wang. Consensus for multi-agent systems with inherent nonlinear dynamics under directed topologies. *Systems & Control Letters*, 62(2):152 – 162, 2013.

[117] John M. Erdman. *Exercises and Problems in Calculus*, page 45. Homepage - John M. Erdman: Portland State University, 2013.

[118] Huaizhu Wang and Jingying Ma. Optimal topology for consensus of heterogeneous multi-agent systems. *Neurocomputing*, 177:594 – 599, 2016.

[119] Jean-Jacques E. Slotine and Weiping Li. *Applied nonlinear control*. Prentice Hall, 1991.