2024

Resilient Cooperative Control of Cyber-Physical Systems: Enhancing Robustness Against Significant Time Delays and Denial-of-Service Attacks

Deepalakshmi Babu Venkateswaran
University of Central Florida

RESILIENT COOPERATIVE CONTROL OF CYBER-PHYSICAL SYSTEMS: ENHANCING ROBUSTNESS AGAINST SIGNIFICANT TIME DELAYS AND DENIAL-OF-SERVICE ATTACKS

by

DEEPALAKSHMI BABU VENKATESWARAN
B.Tech SASTRA University, 2012
M.Sc University of Duisburg-Essen, 2016

A Dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida

Summer Term
2024

Major Professor: Zhihua Qu

ABSTRACT

A cyber-physical control system (CPS) typically consists of a set of physical subsystems, their remote terminal units, a central control center (if applicable), and local communication networks that interconnect all the components to achieve a common goal. Applications include energy systems, autonomous vehicles, and collaborative robots. Ensuring stability, performance, and resilience in CPS requires thorough analysis and control design, utilizing robust algorithms to handle delays, communication failures, and potential cyber-attacks.

Time delays are a challenge in CPS, particularly in teleoperation systems, where human operators remotely control robotic systems. These delays cause chattering, oscillations, and instability, making it difficult to achieve smooth and stable remote robot control. Applications like remote surgery, space exploration, and hazardous environment operations are highly susceptible to these disruptions. To address this issue, a novel passivity-shortage framework is proposed, that enables systems to maintain stability and transparency despite time-varying communication delays and environmental disturbances.

CPS are prone to attacks, particularly Denial-of-Service (DoS) attacks, which disrupt the normal functioning of a network by overwhelming it with excessive internet traffic, rendering the communication channels unavailable to legitimate users. These attacks threaten the stability and functionality of CPS. To enhance resilience in multi-agent systems, novel distributed algorithms are proposed. These graph theory-based algorithms mitigate network vulnerabilities by incorporating strategically placed additional communication channels, thereby increasing tolerance to attacks in large, dynamic networks.

The effectiveness of these proposed approaches is validated through simulations, experiments, and numerical examples. The passivity-shortage teleoperation strategies are tested using Phantom

Omni devices and they show reduced chattering and better steady-state error convergence. A case study demonstrates how the proposed distributed algorithms effectively achieve consensus, even when some agents are disconnected from the network due to DoS attacks.

To my little Krish.

It is because of you, I've become resilient too.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Zhihua Qu, for his continuous support, guidance, and encouragement throughout my PhD journey. I sincerely appreciate the flexibility and understanding he showed during this period. His insights and expertise have been invaluable to my research and personal growth. I am also profoundly grateful to my committee members, Dr. Gita Sukthankar, Dr. Yaser Fallah, Dr. Chinwendu Enyioha, and Dr. Sang-Eun Song, for their time, valuable feedback, and suggestions which greatly enhanced the quality of this dissertation. My heartfelt thanks extend to the University of Central Florida and the Department of Electrical and Computer Engineering for providing me with this incredible opportunity to pursue my doctoral studies.

Special thanks to Dr. Azwirman Gusrialdi, my coauthor, whose collaboration and shared knowledge significantly contributed to my research. I also wish to thank my peers, Kwasi Opoku, Ganesh Marasini, and others, for their insightful discussions on smart grid systems, which have shaped my work. I am particularly thankful to Dr. Giji Skaria and David Douglas for their assistance in setting up and conducting experiments in the lab. Their help was crucial for the practical aspects of my research.

I would also like to acknowledge my role as a Graduate Research Assistant, which provided valuable financial support and resources for my research. Additionally, I appreciate the financial support and project opportunities provided by Avra Medical Robotics and Duke Energy. Their support and the opportunities to work on projects with them have been instrumental in furthering my research and practical experience.

I extend my deepest appreciation to my husband, Sriram Bharadwaj, for his love, support, and continuous encouragement, and for pointing me in the right direction at every step. My parents,

Babu Venkateswaran and Saraswathi, have always believed in me, and my sister, Kripalakshmi Babu Venkateswaran, has been a constant support, instilling in me the confidence that a small-town girl can achieve high academic pursuits. I am immensely thankful to my in-laws, Dr. C.R. Ranganathan and Geetha, for being there to help in every way possible, and for motivating and pushing me to do better.

I thank all my friends and extended family for their continuous support and encouragement, particularly in motivating me to never give up and for always helping me find the right focus during times of self-doubt. Special mention to my friends Deepika, Swetha, Gayathri, Sravya, Saranya, Poorna, Akila, Aswath, Sowmya, Virdeepak, Fathima, Ambreesh, Nithya, Pradeep, Meenakshi, Sathya and my cousin Abhinaya, for their unwavering support and motivation.

To my son, Krish, who came along the way in this journey and made it all the more worthwhile. His presence has taught me to think beyond myself, helping me grow both personally and professionally. His laughter and curiosity have been a constant source of joy and motivation, reminding me of the importance of balance and perseverance.

Last, but certainly not least, I thank all the divine forces of the universe for blessing me with such a supportive village and the privilege and luxury to follow my dreams.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION AND LITERATURE REVIEW

Overview of Cyber-Physical Systems

A cyber-physical control system (CPS) typically consists of a set of physical subsystems, their remote terminal units, a central control center (if applicable), and local communication networks that interconnect all the components to achieve a common goal. These systems find applications in diverse fields, including energy systems, autonomous vehicles, and collaborative robots. Ensuring stability, performance, and resilience in CPS requires comprehensive analysis and sophisticated control design. This involves developing robust algorithms capable of handling delays, communication failures, and potential cyber-attacks. By integrating advanced computational and communication technologies with physical processes, CPS can monitor, control, and optimize operations in real-time. This seamless integration enables enhanced efficiency, reliability, and adaptability, making CPS essential in modern technological applications. Effective CPS design not only improves operational performance but also ensures the system's ability to withstand and recover from adverse conditions, thereby maintaining continuous and reliable operation.

Teleoperation systems are an application of cyber-physical systems that rely heavily on communication capabilities. They feature a master robot, operated by a human, that sends real-time commands to a slave robot at a remote location. The slave robot replicates the actions of the master robot, allowing precise control over long distances. Some examples of teleoperation systems include robotic-assisted surgical systems, remotely operated underwater vehicles (ROVs) for deep-sea exploration, and robotic manipulators used in handling hazardous materials in nuclear power plants.

Very long delays can cause communication channels to be disconnected, and similarly, DoS attacks

can disrupt these channels. Addressing these issues is crucial in multi-agent systems, which rely on networked systems to enable multiple autonomous agents to communicate, share information, and coordinate their actions to achieve common goals. This coordination is essential for the effective functioning of autonomous vehicle fleets, robotic swarms, and distributed sensor networks, smart grid systems. By facilitating effective communication and collaboration, networked systems ensure that these specialized systems can perform complex, coordinated operations efficiently. The following subsections delve into the specifics of teleoperation and consensus in multi-agent systems, illustrating the critical roles networked systems play in these advanced applications.

Introduction to Teleoperation Systems

Teleoperation is a research area within the field of robotics and control systems that focuses on the remote control and manipulation of robotic systems by human operators. Typically, the system has two main components: the master and the slave. The master side is controlled by a human operator and includes a controller and an interface. The interface can be a joystick, a haptic device, an exoskeleton, or any other advanced human-machine interface that makes control intuitive and precise. On the slave side, a remote robotic system receives the control signals from the master side and carries out the desired actions. The system also relays sensory feedback to the operator, obtained through sensors and actuators, allowing the operator to perceive the remote environment and adjust their control inputs accordingly. A bidirectional or two-way communication channel between the human operator and the robotic system facilitates the interaction between the master and the slave side [2] in bilateral teleoperation.

The primary objectives of teleoperation are to achieve real-time interaction, transparent haptic feedback, and dexterity in a remote environment. The effective realization of bilateral teleoperation requires addressing some critical aspects such as control algorithms and communication protocols.

Control algorithms play an important role in synchronizing the outputs or actions of the master and slave systems, ensuring stability, responsiveness, and accurate replication of the commands. On the other hand, communication protocols are essential for facilitating delay-free and reliable transmission of control signals and sensory feedback between the operator and the remote robotic system.

Multilateral teleoperation is an extension of bilateral teleoperation where multiple operators are involved in controlling a remote system. In multilateral teleoperation, multiple operators collaboratively control different aspects or components of the remote system. It involves multi-directional communication and coordination among the operators to achieve a shared objective. It enables a group of operators to collaboratively control a remote system in a coordinated manner, sharing information and coordinating their actions to achieve a common objective, leveraging their collective knowledge and skills. This collaborative approach allows for improved task allocation, workload distribution, fault tolerance, and adaptability, making multilateral teleoperation a valuable framework for various applications, such as telemedicine, robotic surgery, disaster response, and space exploration [3].

Introduction to Consensus in Multi-Agent Systems

Multi-agent systems (MAS) is an area of research within the field of autonomous systems that focuses on the study of systems composed of multiple autonomous entities, known as agents. These agents can be software-based, physical robots, or a combination of both and are capable of perceiving their environment, making decisions, and interacting with each other to achieve individual and collective goals. Each agent acts distributedly using its local knowledge and capabilities, and they communicate and exchange information through a communication network or environment [4].

In multi-agent systems, the main obstacles are to coordinate and cooperate with all the agents

locally to work together towards their common objectives, despite having conflicting individual goals. Additionally, agents may face challenges due to incomplete information in their dynamic environment, while communication channels may be impacted by delays, bandwidth constraints, and noise. Ensuring secure communication channels, authentication mechanisms, and privacy-preserving techniques to protect the confidentiality and integrity of agent interactions is also a significant challenge in multi-agent systems.

Research Problems and Related Works

In this dissertation, two distinct research problems associated with the communication channels within the networked systems are addressed. In this section, the research problems are introduced and an overview of relevant existing works in the literature is presented.

Stability in Teleoperation with Time-Varying Delays

The first issue addresses the delays in the communication channel between the master and the slave side in a teleoperation setup. The goal of the teleoperation system considered is to achieve stability and a synchronized output between the operator and robots while achieving optimal transparency of the haptic feedback. Introducing time-varying delays in the communication channels causes the system to have instability and affects the performance of the overall system.

In literature, the issue of transmission time delay in remote manipulative control systems was first discussed in [5], where they highlight that longer delays can lead to decreased performance, reduced precision, and even instability in remote manipulative control systems. The solutions provided in this paper are to anticipate the negative effects of time delay and compensate for it to improve stability and performance. Following that, the effects of time delay on bilateral control

are investigated in [6], where the authors analyze the use of different control architectures such as position-position, force-position, and force-force architectures, to transmit and process information between the operator and the remote system. The paper investigates the effect of time delay on system stability and performance and presents passivity-based control as a potential approach to mitigate the effects of delays.

In [7] the authors use a simple passivity-based proportional-derivative control with explicit position feedback through delayed proportional action, enabling effective coordination between the master and slave positions. The term passivity refers to a property of a system that characterizes its ability to absorb, store, and release energy without inducing instability. A passive system dissipates or absorbs energy over time, maintaining the stability and boundedness of its states. Later chapters in this dissertation will provide detailed discussions on the concept of passivity and the properties of passive systems.

The paper [8] uses the concept of passivity to model the behavior of the teleoperation system with a time delay to establish passivity conditions. The concept of scattering transformation is used to achieve the desired passivity conditions in the delayed channel. Scattering transformation decomposes the teleoperation system into separate transmission and scattering sub-systems to represent the master and slave sides. The signal flow is between the two sub-systems represented using wave variables defined at the interfaces between the master and slave devices, and they characterize the components of position, force, and velocity. This enables the overall system to satisfy the passivity requirements [9–13].

The wave variable method from which the scattering transformation is derived was first introduced in [14], where efforts and flow variables of the classical passivity-based network design are replaced by velocity and force variables. However, the conventional wave variable transformation has limitations, such as instability under time-varying delays, tracking errors due to biased terms,

position drift without direct position tracking, and signal variations caused by wave reflections [15]. Most of these approaches only consider the maximum delay or a time-varying gain and design the system for the worst-case scenario [16, 17]. Several approaches have been proposed to overcome these limitations, including wave integrals, wave predictions, biased term compensation channels, and modified wave variable transformations. Despite these advancements, the wave-based systems have not fully addressed all the challenges, particularly in dealing with time-varying delay issues while maintaining stability and transparency without any trade-off.

Some of the examples of improvements of the wave-variable approach are as follows: In [18] a passivity-based time delay compensator that simplifies the communication channel to reduce wave reflection and distortion with local force feedback is employed to enhance stability and transparency. In [19], a force-reflecting control scheme is introduced, where an extended-state observer (ESO) with a dynamic gain to estimate force information, and a corrective wave variable method similar to scattering transformation is employed to design the force-reflecting control strategy. The paper [20] proposes a two-layer architecture for direct force reflection, where the master and slave controllers duplicate the energy transferred between the operator and environment to adapt damping gains based on energy flows to ensure stability and reduce operator effort.

To address the issue of time-varying delays, some of the other approaches in the literature used several other control schemes on top of wave-variable or passivity-based control including neural network-based control, adaptive control, and passivity-based control approaches such as the time-domain passivity approach (TDPA). For instance, the paper [21] proposes an adaptive fuzzy control strategy that combines nominal dynamics, partial feedback linearization, and approximation properties of fuzzy logic systems. This approach was tested in a trilateral teleoperation setup with stochastic delays and unmodeled dynamics with uncertainties. In [22], the time-varying delay is compensated for using a disturbance estimation path via a virtual block, along with an adaptive controller. The time-domain passivity approach is utilized in [23] to handle varying time delays

using a Passivity Observer (PO) that monitors energy in real-time and a Passivity Controller (PC) that dissipates the required amount of energy based on the observations of the PO. Similarly, [24] introduces a wave-based TDPA that monitors power flows under arbitrary time delays and employs neural networks to estimate and eliminate dynamic uncertainties.

On the other side, some approaches have eliminated the use of wave variables and scattering transformations by using optimization-based control methods such as model predictive control (MPC), and integral quadratic constraint (IQC), μ synthesis, H_∞ and so on. For example, in [25] a modified MPC approach that includes both current measurement information and a correction signal that reflects the difference between measured information and its prediction is proposed. In [26] an adaptive Smith predictor based on Padé approximation and active observer is developed to model and estimate the time delay, and a sliding mode control is incorporated on the master side to cancel out the effects of time delay on the system.

The existing methods developed for passive systems have demonstrated stability and satisfactory tracking performance. However, their applicability is limited, as they are not designed to support a broader class of systems that may not necessarily exhibit passivity. The concept of dissipativity, which forms the basis of these methods, relies on an energy-based input-output relationship. Passive systems enforce an upper bound on the change in energy by the injected input-output power. On the other hand, passivity-short systems allow their energy changes to be upper bounded by the weighted sum of the injected power, input energy, and output energy.

Passive systems are typically constrained to minimum phase systems with relative degrees of 1 or 0. In contrast, passivity-short systems include a wider range of systems, including non-minimum phase systems and those with higher relative degrees. In practical applications, the majority of systems fall into the passivity-short category.

The first problem statement in this dissertation addresses the control challenges associated with

coordinating master and slave robots in teleoperation scenarios. Specifically, the goal is to design a control system that achieves precise coordination between the master and slave robot while ensuring stability for both passive and passivity-short systems. The existing methods designed for passive systems lack comprehensive support for a broader class of systems that are not necessarily passive. Moreover, the available methods lose stability when applied to passivity-short master and slave systems.

To overcome these challenges, a negative feedback interconnection with an individual compensation system for teleoperation is proposed. By incorporating this approach, stability for passivity-short systems is expected to be achieved, while also outperforming existing methods designed for passive systems. The proposed control system will be thoroughly evaluated, and its effectiveness will be assessed based on the ability of the slave robot to accurately follow the master robot with significantly reduced error and lag compared to the scattering transformation.

The issue of instability and performance caused by time delays is addressed by designing an appropriate controller. Over the past decade, several controls were reported for systems with either small or large delays. In the case of large delays, the so-called virtual environment-based approach, such as [27–30], was used to generate new models for teleoperation scenarios, but no stability proof was reported. The stability proof is generally based on the passivity properties of robot dynamics (from torque input to velocity output). A detailed survey of the existing passivity-based approaches is provided in [9]. The classical approaches include two-port formulation such as scattering transformation (linear transformation of input-output signals to wave variables) approaches [6, 14], force reflecting algorithm using wave variable based four-channel approach discussed in [11, 18], time-domain passivity approach (TDPA) discussed in [23, 31], all of which address stability issues related to delays by passifying the communication channel, under the de-facto assumption that the robot dynamics are passive.

It is interesting to note that most systems in real life are not passive since passivity is very restrictive. It requires the system to be of minimum phase with a relative degree equal to zero or one. Specifically, the robot dynamics is not passive in the position-force domain for all frequency ranges [32–34]. Even when the system is not linear and not passive, many methods aim to correct the uncertainties to make the master and slave systems passive. These methods include integral quadratic constraints (IQC) [1, 35, 36], that compensate for the delay in the communication channel, saturation and monotone nonlinearity of the environment by considering them as a separate block connected in negative feedback with the linear passive system by using appropriate multipliers to make the overall system passive and compensate for the lack of passivity, by introducing an excess of passivity, and [24] where a neural network is proposed to eliminate the dynamic uncertainties of the systems, and [19] eliminates the constraints on the varying delay using adaptive laws, [37] proposes a switching control to guarantee the passivity of teleoperation, with position feedback to improve their tracking performance. Stability proof of all these approaches requires the overall system to be passive.

In addition, there exist studies in the position-force domain that does not make any passivity assumptions. The authors of [38] consider the telerobotics systems as a functional differential equation, and use adaptive control to design stable bilateral teleoperation. A high-gain velocity observer is used in [39], to show the convergence of position error, in the presence of time delays. PD-like controllers are used in [40], where the solutions of linear matrix inequalities are used to analyze the stability, [41] uses the same approach with the terminal sliding mode controller to estimate its velocity. For multi-lateral teleoperation research has proposed various control methods, including model-based controllers [42, 43], and dissipativity-based controllers like the time-domain passivity approach [44] and wave-variable and scattering transformation approach [45]. However, traditional passivity is restrictive. In these existing approaches, knowledge of the dynamics of the system is required to design the controller.

Denial-of-Service Attacks on Multi-Agent Systems

In the past 20 years, the study of multi-agent systems has focused on cooperative behaviors inspired by nature. Consensus and flocking have proven to be effective in achieving collective objectives. One key approach for this is distributed control, where agents work together towards a common goal by using local information and making decentralized decisions. Distributed control has been applied in domains such as smart grid control, transportation systems, multi-robot systems, cyber-physical systems, and medical robotics [46].

When multiple autonomous agents with their own local information interact and exchange data to reach a common decision on a particular quantity of interest, it is known as consensus in a multi-agent system. The primary objective of consensus is to achieve a collective state or value that satisfies all agents in the system, despite communication constraints, uncertainties, and differences in agent dynamics.

Practical constraints in real-world multi-agent systems pose additional challenges to achieving consensus. Communication delays, packet loss, intermittent connectivity, and link failures are among the common constraints encountered in communication channels [33,47,48]. These constraints can significantly impact system performance and stability, making it crucial to address communication issues in multi-agent systems. Researchers have explored techniques such as adaptive consensus algorithms that incorporate adaptive mechanisms to handle uncertainties, parameter variations, and changes in the environment [49, 50].

In the context of multi-agent systems (MASs), cyber-attacks can pose significant threats to their security and operation. The two most common types of cyber-attacks on MASs are the DoS attack and the Deception attack.

A DoS attack targets the communication channels or resources of MASs to disrupt their regular

operation. The attacker sends a high volume of malicious traffic or consumes system resources, resulting in an overwhelmed and unresponsive network or system. This can cause service interruptions, performance degradation, or system failure, ultimately impacting the safety, efficiency, and reliability of MASs. On the other hand, a deception attack aims to manipulate the information exchanged within MASs to deceive the agents. The attacker injects false data or misleading messages into the communication channels, altering the integrity and authenticity of the transmitted information. This can mislead the decision-making processes, leading to incorrect actions or compromised system behavior. As a result, deception attacks can cause the misalignment of agent behaviors, reduced system performance, and compromised security [51].

Both of these attacks have a significant impact on the consensus and pose risks to the security and operation of MASs. To maintain the reliability, safety, and integrity of distributed cooperative control in MASs, it is crucial to detect and mitigate these cyber threats.

To achieve secure and resilient consensus among agents and to mitigate risks, several approaches have been discussed in the literature. The most common approach is using event-based control, addressing several types of DoS attacks such as single channel vs multiple channels, synchronous vs asynchronous DoS attacks, and periodic vs aperiodic DoS attacks. For example, [52] introduces a distributed event-based secure controller for achieving secure consensus in linear multi-agent systems under DoS attacks, addressing the challenge of positive inter-execution time intervals. It contributes to resilient coordination by enabling secure average consensus and avoiding Zeno behavior. [53] proposes an event-triggered resilient control mechanism for multi-agent systems under asynchronous DoS attacks, allowing consensus achievement with reduced communication burden. In [54], the authors propose a secure consensus control scheme in the presence of aperiodic DoS attacks based on an event-triggered mechanism, ensuring stability and reduced communication resources. Current methods for mitigating DoS attacks in multi-agent systems primarily involve event-triggered control, which reduces network traffic by making system updates or send-

ing information only when a certain condition or *event* is met, rather than at regular intervals. Some existing approaches include [55], which employs event-triggered control to optimize transmission times, [56], which introduces event-triggered consensus for multi-agent systems with a focus on episodic network disruptions due to intermittent random DoS attacks. [57] explores a switching-like event-triggered control approach to enhance communication efficiency and maintain control performance in networked control systems (NCSs) under intermittent DoS attacks. Another work, [58], explores an event-triggered mechanism combined with observer-based control to mitigate the effects of DoS attacks in stochastic NCSs. These approaches are vulnerable to stealthy attacks that can manipulate the event-triggering conditions, causing either an overload or a silence of critical communications.

In addition, model-based approaches have been discussed in the literature. In [59], a secure consensus in MIMO (Multiple-Input Multiple-Output) linear MASs (multi-agent systems) under malicious attacks, specifically DoS attacks, is developed based solely on the relative outputs of the agents using the concept of a UIO (Unknown Input Observer) design. [60] addresses fault-tolerant control in nonlinear multi-agent systems under DoS attacks and actuator faults, proposing a distributed control strategy with adaptive schemes, state-feedback control gains, and a switching mechanism to ensure mean square consensus. The paper [61] presents an observer-based event-triggered control protocol for achieving secure consensus in linear multi-agent systems under DoS attacks and unmeasurable states.

Predictor-based and observer-based controls utilize predictive models or state-estimators to compensate for missing or delayed data due to network disruptions caused by DoS attacks such as [62] and [63]. Techniques range from transforming systems affected by DoS attacks into auxiliary systems for stability analysis [64], utilizing interval partition techniques and linear matrix inequalities to counteract DoS attacks in cyber-physical systems [65]. [60] employs adaptive fault-tolerant control (FTC) for nonlinear multi-agent systems in the presence of DoS attacks and actuator faults,

utilizing adaptive schemes and state-feedback control gains. These strategies heavily rely on accurate models or parameter estimation to forecast future states. If the model does not accurately reflect the system's dynamics, the predictions and subsequent control actions may be ineffective or even detrimental.

As DoS attack models evolve to target users with persistent and continually adapting attacks, adaptive control strategies that modify control tactics based on ongoing assessments of system performance and external conditions have emerged. For instance, [66] employs a Distributed Model-Free Adaptive Control (DMFAC) algorithm and an Extended Discrete State Observer (EDSO) to address DoS attacks in multi-agent systems to ensure consensus control. Similarly, [67] introduces an improved dynamic linearization method to develop an equivalent linear data model and an attack compensation mechanism designed to mitigate the impacts of DoS attacks. These approaches have limitations in managing the algorithm without knowledge of the global communication topology and require the DoS attacks to be somewhat predictable.

Graph-based defense mechanisms that utilize either the introduction of additional communication layers or adaptation of existing communication layers work well for persistent, evolving DoS attacks. Some research focuses on network topology optimization algorithms against DoS attacks, which rearrange network connections during attacks such as [68, 69]. However, the assumption that the network configuration is known and unchanging cannot hold true for many cases.

Key Contributions

This dissertation addresses critical challenges in networked control systems, with a focus on teleoperation and multi-agent systems. The key contributions are summarized as follows:

- Developed novel control algorithms to manage communication delays wireless networks,

ensuring stable and real-time interaction for teleoperated systems.

- Extended bilateral teleoperation concepts to multilateral teleoperation, enabling collaborative control by multiple operators.
- Demonstrated improved stability and performance through simulations in Matlab/Simulink and experiments using Phantom Omni haptic devices.
- Created distributed algorithms to identify and eliminate critical edges in networks, enhancing connectivity and robustness.
- Proposed a systematic approach to maintain network connectivity and mitigate the effects of DoS attacks.
- Introduced algorithms for simplifying large networks using strongly connected components, facilitating efficient analysis and improvement of network robustness.

These contributions offer practical solutions to improve the stability, performance, and resilience of networked control systems, addressing key challenges in teleoperation and multi-agent systems.

Organization of Dissertation

This dissertation is organized into six chapters, each contributing to the overarching theme of resilient cooperative control in networked systems. The structure is designed to guide the reader through the fundamental concepts, problem statements, proposed solutions, and validation of these solutions in a coherent manner.

Chapter 1 presents the **Introduction and Literature Review**, setting the stage for the dissertation by discussing the importance of networked systems, particularly in teleoperation and multi-agent

contexts. It provides a comprehensive review of existing literature, highlighting the gaps and challenges that this research aims to address.

Chapter 2 delves into the **Background**, offering essential definitions and properties of passivity-short systems, which are central to the control strategies proposed in this dissertation. It also introduces fundamental concepts from graph theory that are pertinent to the analysis and design of resilient networked systems.

Chapter 3 focuses on **Stability and Performance in Bilateral and Multilateral Teleoperation with Time-Varying Delays**. This chapter provides a detailed analysis of passivity-short systems, proposes novel control frameworks, and presents simulation and experimental results to validate the effectiveness of these frameworks in maintaining stability and performance.

Chapter 4 addresses the resilience of **Multi-Agent Systems Against DoS Attacks**. It introduces distributed algorithms for ensuring connectivity and critical edge elimination, enhancing the robustness of multi-agent systems. The chapter also extends the discussion to directed graphs and evaluates the proposed strategies through simulations.

Chapter 5 presents **Advanced Approaches**, that focuses on identifying critical edges in large graphs by reducing them into smaller, manageable components to enhance network robustness and resilience. It also presents a case study on a smart grid as a cyber-physical system, demonstrating the significance of consensus algorithms for efficient and stable system operations.

Finally, Chapter 6 provides the **Conclusion and Future Work**, summarizing the key findings and contributions of the dissertation. It also outlines potential directions for future research, aiming to inspire further advancements in the field of resilient cooperative control of networked systems.

The chapters are interconnected through the theme of enhancing robustness and performance in networked systems against various challenges, particularly focusing on time delays in teleopera-

tion and DoS attacks in multi-agent systems. Each chapter builds upon the previous one, creating a cohesive narrative that guides the reader through the theoretical foundations, problem identification, proposed solutions, and practical validations.

CHAPTER 2: BACKGROUND

Introduction

This chapter provides the foundational background necessary to understand the advanced concepts and methodologies discussed in subsequent chapters. The fundamental theories and principles that underpin this research are explored, with a particular focus on passivity-short systems and their relevance in control theory and multi-agent systems.

The concept of dissipativity, introduced by Willems in 1972, forms the bedrock of the discussion. Dissipativity characterizes the relationship between a system's input and output, encapsulated through a supply rate and a storage function. This concept leads to various subclasses, including passivity, where the system's energy consumption is bounded by the energy supplied. Passivity is crucial for ensuring the stability of dynamic systems, particularly those involving feedback loops.

However, traditional passivity can be restrictive, as it requires systems to have a relative degree of zero or one. To address this limitation, the notion of passivity-shortage extends the framework of dissipativity, allowing for a broader range of dynamical behaviors while preserving stability. Passivity-short systems introduce flexibility by incorporating weights that balance the input and output energies, thus accommodating systems with higher relative degrees and non-minimum phase behaviors.

In addition to passivity and passivity-shortage, the properties and stability conditions of these systems are examined. Passivity-short systems retain finite-gain L_2 stability, and their interconnection properties, including feedback and series configurations, are explored. The preservation of passivity-short properties in interconnected systems is vital for maintaining overall system stability, especially in complex multi-agent networks.

Furthermore, the application of passivity theory to robotic systems is explored, providing an example with n-link robots. This example illustrates the practical implications of passivity-shortage in ensuring the stability and performance of robotic control systems.

The chapter also extends its focus to the principles of graph theory, which are essential for analyzing connectivity and cooperative control in multi-agent systems. Graph theory provides a powerful framework for modeling and understanding the interactions among agents in a network. Critical concepts such as edge connectivity, critical edges, and strongly connected components are discussed, which are fundamental for ensuring the robustness and resilience of networked systems.

Distributed algorithms for identifying network connectivity and neighbor structures are introduced, which are crucial for maintaining communication and coordination in multi-agent systems. These algorithms enable the detection of network disconnections and the implementation of cooperative control strategies that enhance system resilience against disruptions such as DoSattacks.

Overall, this background chapter sets the stage for the detailed exploration of resilient multi-agent systems and their control strategies. By grounding the discussion in the theoretical underpinnings of passivity, passivity-shortage, and graph theory, the reader is equipped with a solid understanding of the essential concepts that drive this research on enhancing the robustness and resilience of networked systems.

Passivity-Short Systems

The concept of dissipativity, initially introduced in [70], revolves around the relationship between the input and output of a dynamical system. It is characterized by an inequality that involves a supply rate, which is a scalar function relating to the input and output of the system, and a storage function, which is a bounded function capturing the system's energy or another form of resources.

The central idea of dissipativity is that the increase in storage over time must be bounded by the supply rate.

Dissipativity encompasses different subclasses depending on the specific form of the supply rate function. One prominent subclass is passivity, where the supply rate is determined by the energy supplied to the system. In a passive system, the increase in energy cannot surpass the energy introduced to the system. Essentially, a passive system dissipates or absorbs energy in a controlled manner, ensuring stability and the absence of uncontrollable energy growth.

Consider the following general class of dynamic systems:

$$\dot{x} = f(x,u), \quad x(0) = x_0, \; y = h(x,u), \tag{2.1}$$

where $u \in \Re^m$, $x \in \Re^n$, and $y \in \Re^p$. Dynamic mapping $\mathcal{P}$ of system (2.1) denotes the input-output mapping from input u to output y.

Definition 1. *Dynamic mapping $\mathcal{P}$ in* (2.1) *is passive if there exists a positive semi-definite storage function $V(x)$, such that,*

$$V(x(t)) - V(x(0)) \leq \int_0^t y^T(\tau)u(\tau)d\tau. \tag{2.2}$$

Should storage function $V(x)$ be continuously differentiable,

$$\dot{V} = \frac{dV(x)}{dt} = \left(\frac{\partial V}{\partial x}\right)^T f(x,u), \tag{2.3}$$

the passivity condition in (2.2) *can be alternatively expressed as,*

$$\dot{V} \leq y^T u \tag{2.4}$$

It is important to note that the storage functions are not limited to physical energies. Any non-negative Lyapunov function can act as a virtual energy function of the system and can be a potential storage function.

In a free system without external input, passivity corresponds to stability. From Equation (2.4), it is apparent that the zero dynamics of a passive system are stable. Additionally, passivity exhibits an influential property known as the interconnection property, which states that the negative feedback interconnection of passive systems remains passive. However, it is worth noting that the class of passive systems is somewhat restrictive since the relative degree of passive systems must be either 0 or 1.

To overcome the limitations of the passive systems class and offer more flexibility, a broader and less restrictive subclass of dissipativity, known as passivity-shortage, was introduced in [71]. The passivity-shortage subclass offers a broader framework that accommodates systems with higher relative degrees and extends the understanding of the relationship between input, output, and storage functions in dynamic systems.

Definition 2. *Dynamic mapping $\mathcal{P}$ of system* (2.1) *is said to be input passivity-short or simply passivity-short if there exists a positive definite and continuously differentiable storage function $V(x)$ and non-negative weights $\{\epsilon, \varrho\}$ such that*

$$\dot{V} = \left(\frac{\partial V}{\partial x}\right)^T f(x,u) \leq u^T y + \frac{\epsilon}{2}\|u\|^2 - \frac{\varrho}{2}\|y\|^2. \tag{2.5}$$

This definition presents a fundamental property of the system where the rate of change of the storage function $\dot{V}$ is bounded by a combination of the input-output relationship and the energy terms involving weights ϵ and ϱ. In simpler terms, the passivity-short property implies that the system's energy changes are limited by a weighted sum of the power injected through the input

signal u, the input energy, and the output energy. The inequality (2.5) ensures that the increase in the storage function V is upper bounded by the power injected through u and the difference between the input and output energies, weighed by ϵ and ϱ, respectively.

Expanding on the sub-classes of dissipativity derived from the passivity-short inequality (2.5), several additional categories outlined in [72] can be obtained. These subclasses provide further insights into the system's energy behavior and stability properties.

If $\epsilon \leq 0$, the system is referred to as *output strictly passive*. In this case, the inequality $\dot{V} \leq u^T y - \frac{\varrho}{2}|y|^2$ holds, which means that the rate of change of the storage function $\dot{V}$ is bounded by the power injected through u and a term related to the squared norm of the output y. The negative term $-\frac{\varrho}{2}|y|^2$ in the inequality indicates energy dissipation due to the output, contributing to system stability.

Similarly, if the condition $\epsilon' = -\epsilon$ is satisfied, the system is classified as *input strictly passive*. This implies that $\dot{V} \leq u^T y - \frac{\epsilon'}{2}|u|^2$, where the rate of change of the storage function $\dot{V}$ is bounded by the power injected through u and a term related to the squared norm of the input u. The negative term $-\frac{\epsilon'}{2}|u|^2$ in the inequality represents energy dissipation due to the input, contributing to system stability.

Furthermore, when both ϵ and ϱ are equal to zero, the system is classified as *lossless passive*. This aligns with the condition mentioned earlier in (2.4), where the rate of change of the storage function $\dot{V}$ is solely determined by the power injected through u, without any energy dissipation terms.

It is worth noting that most linear systems with Lyapunov stability can be made passivity-short or inherently possess passivity-short properties through the use of appropriate output feedback control [73]. Passivity-short systems offer a broad framework for analyzing and designing stable

systems, encompassing a wide range of linear dynamics.

It is also interesting to note that passivity-short systems can achieve the same properties as passive systems, but they also encompass a broader range of dynamics and stability characteristics.

In particular, passivity-short systems have the ability to include non-minimum phase systems, which are systems where the output responds before the input is applied. Non-minimum phase systems pose challenges in achieving stability and robustness, but the passivity-short property allows for the analysis and design of such systems, enabling the consideration of their unique dynamics.

Furthermore, passivity-short systems can handle systems with relative stability greater than 1. Relative stability refers to the stability properties of the system's poles in the complex plane. Systems with relative stability greater than 1 exhibit oscillatory behavior and can be more challenging to analyze and control. However, passivity-shortage provides a useful framework for studying and designing systems with these characteristics, contributing to the understanding of their stability and energy behavior.

Properties of Passivity-Short Systems

Passivity-short and passive systems have unique characteristics that affect their stability and interconnection outcomes. This section explores some of the unique characteristics of passivity-short systems and how they differ from passive systems.

Stability Properties

Finite-gain L_2 stability, also referred to as L_2 stability, is a concept used to assess the input-output stability of a system when its dynamics are not known or available. This concept evaluates the

connection between the input and output signals of the system based on their energy properties. Refer to [74] for proof.

Definition 3. *The system* (2.1) *is* L_2 *stable if*

$$\|y\|_{L_2} \leq \kappa \|u\|_{L_2} + \rho \tag{2.6}$$

where κ *is a non-negative constant referred to as the* L_2 *gain,* ρ *is also a non-negative constant called the bias term, and* $\|.\|$ *denotes the functional 2-norm.*

The system (2.1) is said to be L_2 stable if the 2-norm of the output signal (y) remains bounded by a weighted sum of the 2-norm of the input signal (u) and a bias term (ρ). The $L2$ stability condition ensures that for any bounded input signal with finite energy, the resulting output signal also has finite energy and remains bounded. It provides a quantitative measure of how the energy of the input signal propagates to the energy of the output signal.

Property 1. *System* (2.1) *is passivity-short in the form of* (2.5)*, and it is* L_2 *stable if the storage function* $V(x)$ *is positive definite and the weight* ϱ *is positive definite.*

When the weight ϱ in equation (2.5) is set to zero, system (2.1) loses its L_2 stability property. However, it is possible to restore L_2 stability to the system by implementing a negative feedback control law.

Property 2. *System* (2.1) *is passivity-short in the form of* (2.5)*, and when* $\varrho = 0$*, system* (2.1) *can recover its* L_2 *stability using a negative feedback control* $u(t) = v - \frac{y(t)}{\epsilon}$ *for the input-output pair,* $\{v, y\}$

Proof: Consider the control $u(t) = v - ky(t)$ in (2.5),

$$\begin{aligned}\dot{V} &\leq (v - ky)^T y + \frac{\epsilon}{2}\|v - ky\|^2 - \frac{\varrho}{2}\|y\|^2 \\ &\leq v^T y - k\|y\|^2 + \frac{\epsilon}{2}\|v\|^2 + \frac{\epsilon}{2}k^2\|y\|^2 - \epsilon k v^T y - \frac{\varrho}{2}\|y\|^2\end{aligned}$$

When $\varrho = 0$, and $k = \frac{1}{\epsilon}$ then $\dot{V} \leq -\frac{1}{2\epsilon}\|y\|^2 + \frac{\epsilon}{2}\|v\|^2$, thus by property 1, the system is L_2 stable.

That is, by choosing the control input $u(t)$ as $v(t) - ky(t)$, where $v(t)$ is the desired input and $y(t)$ is the system's output, the system can regain its L_2 stability. This negative feedback control law introduces a corrective term that depends on the output $y(t)$ and is proportional to the system gains $k = \frac{1}{\epsilon}$. The negative feedback control effectively adjusts the input $v(t)$ based on the discrepancy between the desired output and the actual output.

By employing this negative feedback control, the system compensates for any deviations between the desired input and the system's output, ensuring that the energy of the output signal remains bounded. This restores the L_2 stability property to system (2.1), even in the absence of a positive definite weight term ϱ.

Property 3. *System* (2.1) *is passivity-short in the form of* (2.5)*,and if the autonomous system* $\dot{z} = F(z, -ky)$ *and* $y = h(z)$ *is zero-state observable, it is asymptotically stable under the output feedback control* $u = -ky$ *with* $0 < k < \frac{2}{\epsilon}$.

Proof: Consider the control $u(t) = v - ky(t)$ in (2.5),

$$\begin{aligned}\dot{V} &\leq (-ky)^T y + \frac{\epsilon}{2}\| - ky\|^2 - \frac{\varrho}{2}\|y\|^2 \\ &\leq (-k + k^2\epsilon - \frac{\varrho}{2})\|y\|^2 \qquad (2.7)\end{aligned}$$

Thus, by applying the output feedback control law with an appropriate gain parameter, the system

will converge to a stable equilibrium point over time, $\lim_{t\to\infty} y = 0$. It is important to note that the condition of zero-state observability ensures that the system's internal state can be accurately estimated from the output measurements, which is crucial for stability analysis and control design, under which the system can achieve asymptotic stability.

Property 4. *Nyquist plot of a passive system lies completely on the left half of the s-plane, but an L_2 stable passivity-short system is not limited to the left half of the s-plane. It can lie slightly on the right half plane.*

The Nyquist plot is a graphical representation of the frequency response of a system. For a system that satisfies the property of passivity, the Nyquist plot lies entirely in the left half of the s-plane. On the other hand, an L_2 stable passivity-short system, which satisfies the passivity-short inequality, is not limited to the left half of the s-plane [74].

Connectivity Properties

The passivity-short property can be preserved in certain interconnections of passivity-short systems. This section lays out the summary of an important existing result of feedback interconnection of passivity short systems.

The following lemma provides conditions for the passivity-short property of the overall system in a negative feedback interconnection of passivity-short systems:

Lemma 1. *Consider two systems, P_i and P_j, connected in a negative feedback configuration as shown in Figure 2.1. Suppose that P_i and P_j are passivity-short with parameters $\epsilon_i, \epsilon_j > 0$ as defined in* (2.5)*, while $\varrho_i, \varrho_j = 0$. If the gain k_i is chosen such that $k_i \in \left(0, \max\left(\frac{1}{\epsilon_i}, \frac{1}{\epsilon_j}\right)\right)$, then the overall system is passivity-short with respect to the input-output pairs $v = [v_i, v_j]$ and $y = [y_i, y_j]$.*

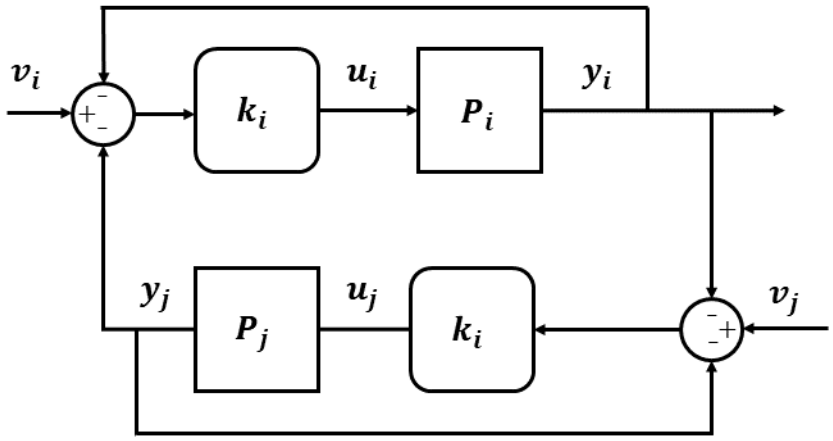


Figure 2.1: Interconnection of passivity-short systems

This lemma highlights the conditions under which the passivity-short property is preserved in a negative feedback interconnection. By appropriately choosing the gain k_i within the specified range, the passivity-shortness of the individual systems extends to the overall interconnected system. Lemma 1 does not include time delays. The following lemma extends lemma 1 by taking into consideration, the effect of time delay.

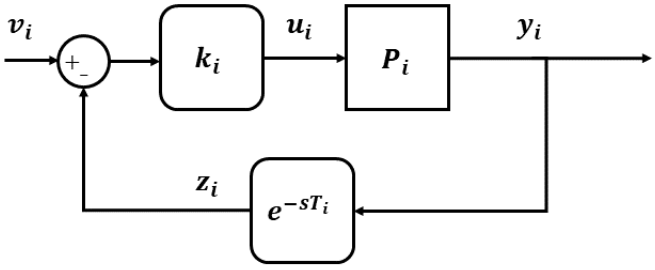


Figure 2.2: Passivity-short system with feedback delay

Lemma 2. *Suppose that the system P_i in Figure 2.2 is passivity-short with parameters $\epsilon_i, \varrho_i > 0$ as defined earlier in* (2.5)*. The overall system is passivity-short and $\mathcal{L}_2$ stable, independent on the time delay T_i, and with respect to its input-output pair v_i and y_i, if the following two conditions are satisfied:*

$$\begin{aligned} &\varrho_i > 1/\epsilon_i, \\ &k_i \in \left(0, \frac{-1+\sqrt{2\epsilon_i\varrho_i+1}}{2\epsilon_i}\right). \end{aligned} \tag{2.8}$$

***Proof*:** The input to the system P_i is given by, $u_i(t) = k_i(v_i(t) - z_i(t))$, where $z_i(t) = y_i(t - T_i)$. Let us consider the storage function V as,

$$V(t) = \frac{1}{k_i}\left(V_i(t) + V_{d_i}(t, T_i)\right), \tag{2.9}$$

where $k_i > 0$, V_i is the storage function of P_i and $V_{d_i}(t, T_i)$ is the storage function of the delay channel given by,

$$V_{d_i}(t, T_i) = \left(\frac{1}{2} + \epsilon_i k_i\right) \int_{t-T_i}^{t} \|y_i(\tau)\|^2 d\tau.$$

Differentiating (2.9) and substituting $\dot{V}_i(t)$ given by definition of passivity-short systems as presented in (2.5) yield,

$$\begin{aligned} \dot{V} \leq & v_i(t)^T y_i(t) + y_i^T(t - T_i) y_i(t) \\ & + \frac{\epsilon_i}{2} k_i \left\|v_i(t) - y_i(t - T_i)\right\|^2 - \frac{\varrho_i}{2}\|y_i(t)\|^2 + \dot{V}_{d_i}(t, T_i) \\ \leq & v_i(t)^T y_i(t) + \epsilon_i k_i \|v_i(t)\|^2 - \left(\frac{\varrho_i}{2k_i} - (\epsilon_i k_i + 1)\right)\|y_i(t)\|^2, \end{aligned}$$

which is passivity-short and $\mathcal{L}_2$ stable from input v_i to output y_i according to (2.8). ■

The property described in Lemma 2 can be extended to positive feedback interconnections and

series interconnections, resulting in passivity-short and $\mathcal{L}_2$ stable overall systems.

In the case of positive feedback interconnection, where the output of one passivity-short system is fed back to another, the overall system remains passivity-short and $\mathcal{L}_2$ stable if $\varrho_i > 0$ for each subsystem and the feedback gain k_i is sufficiently small. The specific condition on k_i ensures that the feedback does not introduce instability or violate the passivity-short property.

In the case of series interconnection, where the delay and the passivity-short system are connected in an open loop, the overall system can also maintain the passivity-short property and $\mathcal{L}_2$ stability. This means that the stability and input-output behavior of the passivity-short system are preserved even in the presence of a time delay.

Example: Robot Dynamics

In this section, a well-known example of passivity theory, which is the property of robot dynamics, is discussed.

Consider the dynamics of an n-link robot:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau, \tag{2.10}$$

where q, $\dot{q}$, $\ddot{q}$ are joint displacement, velocity and acceleration, respectively; $M(q)$ is the inertia matrix, $C(q, \dot{q}) \leq \xi_c(q)\|\dot{q}\|$ is the Coriolis matrix, $\xi_c(q)$ is either a known constant if the arm is all-revolute-joint or a known function if the arm has prismatic joint(s), $g(q)$ is the gravity vector, and τ is the torque control input.

Lemma 3. *Consider an n-link robot* (2.10)*, equipped with the control given by,*

$$\tau = v + g(q) - k_p q - k_v \dot{q}, \tag{2.11}$$

which is a simple PD feedback control, with gravity compensation, and v is the overall system input (for force or torque). If the gains are chosen according to the following[1]*:*

$$\begin{aligned} k_p >& 0, \\ k_v \geq& \lambda_{max}(M(q)) + \|q\|\xi_c(q) + \frac{1}{2\epsilon_q}, \end{aligned} \tag{2.12}$$

where $\lambda_{max}(M(q))$ is the maximum eigen value of the inertia matrix $M(q)$, then the corresponding robotic system has the following properties :

- *passive from input v to velocity $\dot{q}$,*
- *passivity-short from input v to position q, but not passive.*

Proof: Let us recall the following properties of robotic manipulators [75]:

(i) Inertia matrix is symmetric, positive definite and bounded such that, $\underline{m}I < M(q) < \bar{m}I$, where I is an identity matrix and $\underline{m}$ and $\bar{m}$ are known positive constants.

(ii) The Coriolis term $C(q, \dot{q})$ is bounded by $\dot{q}$ such that $C(q, \dot{q})\dot{q} \leq \xi_c(q)\|\dot{q}\|^2$ where $\xi_c(q)$ is a defined function for a prismatic joints or $\xi_c(q) = \xi_c$ is a known constant for revolute joints.

(iii) The matrix $[\dot{M}(q) - 2C(q, \dot{q})]$ is skew-symmetric and $\|\dot{M}(q)\| \leq \xi_c\|\dot{q}\|$.

[1]Under the assumption that the configuration space of the rigid body dynamics is finite, inequality (2.12) can always be satisfied.

Consider a storage function $V_1(q)$ as,

$$V_1(q) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}(k_r)\|q\|^2. \tag{2.13}$$

Taking derivative, substituting the control law in (2.11), and applying property (iii) yield,

$$\dot{V}_1(q) = \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{M}(q)\dot{q} + k_r q^T \dot{q} = \dot{q}^T v - k_v\|\dot{q}\|^2,$$

which shows passivity from input v to velocity $\dot{q}$.

Now, let us consider a storage function $V(q) = V_1(q) + V_2(q)$, where V_1 is given by (2.13) and V_2 is given by,

$$V_2(q) = q^T M(q)\dot{q} + \frac{1}{2}k_v\|q\|^2. \tag{2.14}$$

Differentiating (2.14), applying the control law (2.11) and properties (i)-(iii) yield,

$$\begin{aligned}
\dot{V}(q) &= \dot{V}_1 + q^T M(q)\ddot{q} + \dot{q}^T M(q)\dot{q} + q^T \dot{M}(q)\dot{q} + k_v x^T \dot{q} \\
&\leq \dot{q}^T v - k_v\|\dot{q}\|^2 + q^T v - k_r\|q\|^2 + \lambda_{max}(M(q))\|\dot{q}\|^2 + \|q\|\xi_c(q)\|\dot{q}\|^2, \\
&\leq \dot{q}^T v + q^T v - k_r\|q\|^2 - \varrho_x\|\dot{q}\|^2 - \frac{1}{2\epsilon_x}\|\dot{q}\|^2.
\end{aligned}$$

Since $\dot{q}^T v \leq \frac{1}{2\epsilon_x}\|\dot{q}\|^2 + \frac{\epsilon_x}{2}\|v\|^2$, we can re-write $\dot{V}(q)$ as,

$$\dot{V}(q) \leq q^T v - k_r\|q\|^2 + \frac{\epsilon_x}{2}\|v\|^2, \tag{2.15}$$

where $\varrho_x = k_v - \lambda_{max}(M(q)) - \|q\|\xi_c(q) + \frac{1}{2\epsilon_x}$ is positive according to (2.12). Hence, passivity shortage from input v to position q is shown.

To show that the dynamics from v to q is not passive, note that the closed-loop system is given by,

$$M\ddot{q} + C(q,\dot{q})\dot{q} + k_r q + k_v\dot{q} = v. \tag{2.16}$$

which is non-linear and of relative degree 2 [72]. It is well known that passive systems should have a relative degree 1. Hence the dynamics (2.10) is not passive from input v to position q. ■

Alternatively, the following Lyapunov function can also be used to prove the Lemma 3.

$$\begin{aligned} V &= \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix} \begin{bmatrix} \alpha I & \sigma M \\ \sigma M & M \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \\ &= \alpha q^T q + \dot{q}^T M(q)\dot{q} + 2\sigma \dot{q}^T M(q) q, \end{aligned} \tag{2.17}$$

where $\alpha > 0$ is a constant, σ is a non-negative constant satisfying $\sigma^2 \leq \alpha\lambda_{\min}(M)/\lambda^2_{\max}(M)$. It is straightforward to verify that, under the choices of α and σ, the Lyapunov function (2.17) is positive definite. Based on standard properties of robotic dynamics summarized in [75], it is shown in [33] that the robot dynamics in (2.10) is passivity-short and L_2 stable from input v_i to position output x_i, with passivity-short indices $[\varrho_i, \epsilon_i] = [k_{p_i}, 1/(2\alpha_i - \lambda_{max}(M_i))]$ in (2.5).

In summary, the example and the lemma therein discuss a control scheme for an n-link robot using a proportional-derivative (PD) feedback control with gravity compensation. The analysis reveals that by choosing suitable control gains, the robotic system exhibits different properties. Specifically, it is shown to be passive from the input to the velocity, indicating energy dissipation and stability. However, it is only passivity-short from the input to the position, where the increase in energy is bounded.

Effects of Varying Time Delay

In this section, time-varying delayed interconnections of passivity-short systems are investigated in serial and feedback configurations. It is shown that passivity-short systems arise naturally from these configurations. The conditions for parameter selection are presented, which would eliminate

the potentially destabilizing effect of varying time delays and preserve passivity-shortness and in turn stability of the overall system.

Serial Connection

Consider a serial interconnection of dynamic mapping $\mathcal{P}$ of input-output pair $\{u, y\}$ and a time-varying delay channel whose output is $z(t) = y(t - T(t))$, as shown in Figure 2.3a.

In what follows, an assumption regarding the bounds on the rate of change of delay is discussed. In general, bounds can be set on either the delay itself or on the rate of change of delay, which is less restrictive than the former.

Assumption 1. *The maximum rate of change of delay is lesser than* 1*,* $\dot{T}_{max} < 1$.

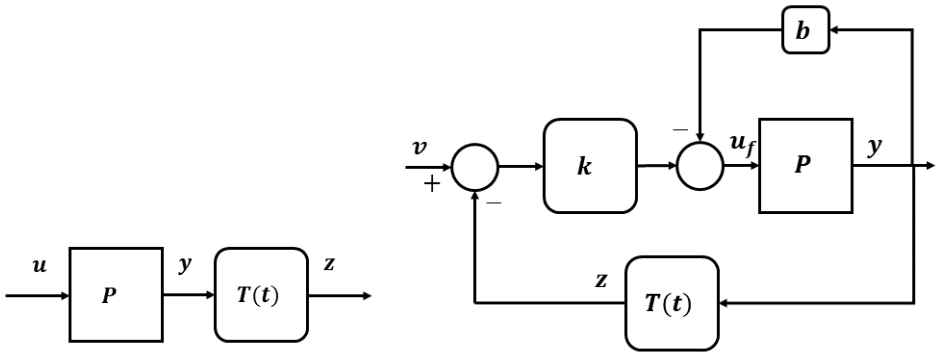
Note that for a causal continuous system, $\dot{T} \leq 1$ is naturally guaranteed [15]. For different values of $\dot{T}$, the following outcomes are observed on the sample time $t_s = (t - T(t))$ and the delayed signal:

- If the delay increases with $\dot{T} < 1$, the sample time increases slowly, and the delayed signal is stretched.
- If the delay increases as fast as the time itself with $\dot{T} = 1$, the delayed signal becomes a constant.
- If the delay increases faster with $\dot{T} > 1$, then the sample time goes backward, making the system non-causal.

To preserve all the data, and the order in which the data is transmitted, the rate of change of delay should be less than 1. In the case of discretized systems, the rate of change of delay can

be higher than one. Then, a discretized counterpart of the preliminary results in this paper can be derived using the conditions in [73]. This paper only considers continuous-time systems, and hence Assumption 1 is held throughout the paper.

The following lemma summarizes the stability properties, and its proof is included in the appendix.



(a) Serial configuration (b) Feedback configuration

Figure 2.3: Interconnection of a passivity-short system and a time-varying delay

Lemma 4. *Consider passivity-short mapping $\mathcal{P}$ as defined in* (2.5) *and with weights $\{\epsilon, \varrho\}$. Suppose that there exist positive constants c_1, c_2 and ϱ' such that*

$$\varrho \geq c_1 + \frac{1}{c_2}, \tag{2.18}$$

$$0 \leq \inf_t \left[\frac{c_1}{2} \left(1 - \dot{T}\right) - \frac{1}{2c_2} \right] \triangleq \varrho'. \tag{2.19}$$

Then, the dynamic mapping $\mathcal{P}'$ from u to z is also passivity-short with weights $\{\epsilon', \varrho'\}$, where $\epsilon' = \epsilon + 2c_2$ and ϱ' is defined by (2.19).

Proof: It follows from Figure 2.3a that $z(t) = y(t - T(t))$. Choose the storage function of the overall system as

$$L = V + V_d, \tag{2.20}$$

where V is the storage function of mapping $\mathcal{P}$ and V_d is the storage function associated with delay for positive constant c_1 is

$$V_d(t, T(t)) = \frac{c_1}{2} \int_{t-T(t)}^{t} \|y(\tau)\|^2 d\tau. \tag{2.21}$$

Then, it follows from (3) that

$$\begin{aligned}
\dot{L} &\leq u^T(t)y(t) + \frac{\epsilon}{2}\|u(t)\|^2 - \frac{\varrho}{2}\|y(t)\|^2 + \dot{V}_d \\
&\leq u^T(t)y(t) + \frac{\epsilon}{2}\|u(t)\|^2 - \frac{\varrho}{2}\|y(t)\|^2 + \frac{c_1}{2}\|y(t)\|^2 - \frac{c_1}{2}\left(1 - \frac{dT}{dt}\right)\|z(t)\|^2 \\
&\leq u^T(t)z(t) + \left[\frac{\epsilon}{2} + c_2\right]\|u(t)\|^2 - \left[\frac{\varrho}{2} - \frac{c_1}{2} - \frac{1}{2c_2}\right]\|y(t)\|^2 - \left[\frac{c_1}{2}\left(1 - \frac{dT}{dt}\right) - \frac{1}{2c_2}\right]\|z(t)\|^2 \\
&\leq u^T(t)z(t) + \frac{\epsilon'}{2}\|u(t)\|^2 - \frac{\varrho'}{2}\|z(t)\|^2,
\end{aligned} \tag{2.22}$$

where c_2 is any positive constant. This together with (6) completes the proof.

The Assumption 1 ensures inequality (2.19) and in turn existence of $\varrho' > 0$ since inequalities (2.18) and (2.19) can be satisfied by choosing small c_1 and large c_2. Thus, L_2 stability of the serial connection in Figure 2.3a is assured.

The expression $\epsilon' = \epsilon + 2c_2$ has two important implications. First, it demonstrates the property of passivity-shortage being invariant under time delay. This means that the passivity-short property of a system remains unchanged even in the presence of time delays. Second, it highlights the fact that a pure time delay itself is not passive (as indicated by its Nyquist plot being a unit circle), but when combined with a passive system, it results in a passivity-short system. This implies that a time delay or a delayed dynamic system inherently possesses passivity-shortness. These reasons motivate the adoption of the passivity-short framework in this paper to investigate the stability of teleoperation systems.

It is worth noting that when $\varrho = 0$, the mapping $\mathcal{P}$ is passivity-short but not L_2-stable. In this

case, inequality (2.18) cannot be satisfied, leading to an additional positive term and the loss of L_2 stability. However, according to property P2, the L_2 stability can be restored by introducing a feedback control law $u(t) = v - bz(t)$ with the modified input-output pair v, z. This feedback control enables the system to achieve L_2 stability despite the absence of passivity in the original passivity-short mapping.

Feedback Configurations

Passivity is known to be preserved in a delayed negative feedback interconnection. This subsection extends this property to passivity-short systems, showing that the passivity-short property is also preserved in a delayed negative feedback interconnection. Additionally, it is demonstrated that a positive feedback interconnection with varying time delays of any system is passivity-short. In both cases, L_2 stability can be achieved through a simple feedback interconnection. The stability results are summarized in the following lemma. Due to space limitations, the proof of this lemma is omitted but can be obtained by following a similar approach to that of Lemma 4.

Lemma 5. *Consider passivity-short mapping $\mathcal{P}$ as defined in* (2.5) *with input-output pair $\{u_f, y\}$ and weights $\{\epsilon, \varrho\}$. Suppose that Assumption 1 is satisfied and there exist positive constants c_1, c_2, k, and b such that*

$$0 \leq \inf_t \left[\frac{\varrho + 2b}{2k} - \frac{c_2}{2} - \frac{b\epsilon_f + c_1}{2k} \right] \triangleq \varrho_f, \tag{2.23}$$

$$0 < \epsilon_f < \left[\frac{c_1}{2k} \left(1 - \dot{T} \right) - \frac{1}{2c_2} \right], \tag{2.24}$$

$$\text{where } \epsilon_f = \epsilon \left(k + \frac{b}{2} \right). \tag{2.25}$$

Then, the dynamic mapping $\mathcal{P}_f$ from v to y is also passivity-short with weights $\{\epsilon_f, \varrho_f\}$ defined by (2.25) *and* (2.23) *respectively.*

The above lemma introduces two conditions. Firstly, condition (2.23) ensures the existence of a positive value ϱ_f, guaranteeing L_2 stability of the overall system depicted in Figure 2.3b. Secondly, the value of ϵ_f is directly obtained from equation (2.25). The condition (2.24) for ϵ_f implies that the resulting system can be passivity-short but not necessarily passive.

The gains are chosen as follows: given the weights ϵ, ϱ, a small positive value for c_1 is selected to satisfy the quadratic inequality $\frac{\epsilon b^2}{2} - 2b < \varrho - \frac{c_1}{2}$ for a positive value of b. By substituting the chosen values of c_1 and b into $\varrho \geq c_2 k + c_1$, and considering $k \in \left[0, \frac{1}{c_2 + \epsilon b}\left\{\varrho + 2b - \frac{\epsilon b^2}{2} - \frac{c_1}{2}\right\}\right]$, two linear inequalities with two unknown variables c_2 and k are obtained. Solving these inequalities yields a large positive value for c_2 and a small positive value for k. With small c_1, large c_2, and small k, condition (2.24) is satisfied as long as Assumption 1 is valid. Notably, the left-hand side of (2.24) is satisfied since $k > 0$ in (2.25), resulting in $\epsilon_f > 0$.

Lemma 5 can also be applied to a passivity-short mapping $\mathcal{P}$ without L_2 stability ($\varrho = 0$). In this case, the L_2 stability of the negative feedback is achieved by selecting a local gain $b > 0$. Conversely, if the mapping $\mathcal{P}$ is passive ($\epsilon = 0$), then the overall negative feedback system is passive.

From Lemma 5, it is evident that if the system $\mathcal{P}$ in Figure 2.3b is passivity-short, then the overall system is also passivity-short, and L_2 stability can always be achieved for this interconnection. If $\mathcal{P}$ is passive, then the overall system is passive. This property can be utilized to design a teleoperation controller that includes either passive or passivity-short master and slave systems.

Connectivity Analysis of Multi-Agent Systems

Consider a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \ldots, n\}$ represents the set of agents and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ denotes the set of edges. In directed graphs, an edge $(i, j) \in \mathcal{E}$ signifies direct influence

or communication from agent j to agent i. Conversely, in undirected graphs, an edge (i, j) implies bidirectional communication between agents i and j. The adjacency matrix $A = [a_{i,j}]$ is defined where $a_{i,j} > 0$ if $(i, j) \in \mathcal{E}$; otherwise, $a_{i,j} = 0$. The in-degree d_i and out-degree d_i^o are determined by $d_i = \sum_j a_{i,j}$ and $d_i^o = \sum_j a_{ji}$, respectively, in directed graphs. For undirected graphs, the degree of agent i, denoted as d_i, is $\sum_j a_{i,j}$. The graph's Laplacian matrix $\mathcal{L} = \mathcal{D} - A$ includes $\mathcal{D}$, a diagonal matrix with diagonal elements d_i.

In a directed graph, the network is termed *strongly connected* if there exists a directed path from any node to every other node within the graph. Conversely, a directed graph is considered *weakly connected* if replacing all its directed edges with undirected edges results in an undirected graph that is connected, meaning there is a path between any pair of nodes, regardless of the direction of the edges, in the underlying undirected graph. In contrast, an undirected graph is *connected* if there is a path between any pair of nodes.

The p^{th} *neighbor* of a node i, denoted as $j \in \mathcal{N}_i^{(p)}$, is identified by the shortest path comprising p edges between nodes i and j, where p represents the *distance* between these nodes.

Cycles in both undirected and directed graphs are closed paths where a sequence of distinct edges or directed edges leads back to the starting node without retracing any edges. A *critical edge* in an undirected graph is an edge whose removal disconnects the network, often referred to as a bridge. In directed graphs, a critical edge, upon removal, disrupts the strong connectivity, turning the graph into a weakly connected or disconnected one.

Edge connectivity quantifies the minimum number of edges that must be removed to disconnect the network or make it not strongly connected in the case of directed graphs. This metric reflects the network's robustness against edge failures. Networks with an edge connectivity of 1 have at least one critical edge, indicating a vulnerability where its removal can split the network. Conversely, a higher edge connectivity, such as 2 or more, implies a robust structure with redundant paths that

maintain connectivity even when one or more edges fail.

Strongly connected components (SCCs) in directed graphs are maximal subsets of nodes where each node within the subset is reachable from every other node via directed paths. In undirected graphs, these components are simply called connected components. If $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is not strongly connected as a whole, it can be decomposed into several SCCs. This decomposition is evident from the diagonal blocks of the network's adjacency matrix when reordered into its lower triangular canonical form, highlighting core connectivity clusters and the hierarchical structure of the network.

Each SCC in a directed graph can be classified as either a *source* or a *sink*. *Source SCCs* have no incoming edges from nodes outside the component, while *sink SCCs* have no outgoing edges. This classification impacts the overall connectivity and potential information flow across the graph, making it crucial for designing resilient systems capable of maintaining connectivity despite disruptions or failures.

The analysis of network structures is enriched by matrix properties. A matrix E with positive or nonnegative entries is considered positive (denoted $E > 0$) or nonnegative ($E \geq 0$) respectively. A nonnegative matrix $E \in \Re^{n \times n}$ is irreducible if no permutation matrix can transform it into a lower triangular form. This irreducibility is critical for ensuring that the network does not decompose into isolated subcomponents, which is expressed as:

$$P^T E P = \begin{bmatrix} E'_{11} & 0 & \cdots & 0 \\ E'_{21} & E'_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ E'_{p1} & E'_{p2} & \cdots & E'_{pp} \end{bmatrix} = E'_{\triangle},$$

where each E'_{ii} is either an irreducible submatrix or a scalar. These blocks delineate the SCCs

of the graph, highlighting the underlying connectivity structure essential for robust consensus and control in multi-agent systems.

Distributed Identification of Node's Neighbor Structure

In this section, a novel algorithm is presented to verify network connectivity and identify the in-neighbors along with the shortest in-neighbor distances for every node. This algorithm effectively maps out the in-neighbor structure of the entire graph. It is applicable to both directed and undirected networks and serves as the foundational step for both Approach 1 and Approach 2, which are discussed in the following sections.

To facilitate its development, the following assumption is made.

Assumption 2. *Every node in $\mathcal{G}$ is uniquely indexed from 1 to n. Every node knows the indices of its neighbors and the total number of nodes n. Furthermore, when necessary, any node can establish a new edge with another node.*

For each node $i \in \mathcal{N}$, consider the following two states:

$$\xi_i(k) = [\xi_{i,1}(k) \ \cdots \ \xi_{i,n}(k)]^T \in \mathbb{R}^n$$

denoting the node i's estimate of its in-neighbors, and

$$\omega_i(k) = [\omega_{i,1}(k) \ \cdots \ \omega_{i,n}(k)]^T \in \mathbb{R}^n$$

denoting node i's estimate of its in-neighbor structure.

The distributed maximum and minimum protocols operate over n consecutive iterations, defined

as follows for each iteration:

$$\xi_{i,j}(k+1) = \max_{l \in \mathcal{N}_i \cup i} \xi_{l,j}(k), \quad j \in \mathcal{N}, \quad k = 0, \cdots, (n-1) \tag{2.26}$$

$$\omega_{i,j}(k+1) = \begin{cases} \omega_{i,j}(k) & \text{if } \xi_{i,j}(k+1) = \xi_{i,j}(k), \\ \min_{l \in \mathcal{N}_i}(\omega_{l,j}(k)+1) & \text{if } \xi_{i,j}(k+1) > \xi_{i,j}(k) \end{cases}$$

$$k = 0, \cdots, (n-1) \tag{2.27}$$

$$\rho_i(k+1) = \max_{j \in \mathcal{N}_i \cup \{i\}} \rho_j(k), \quad k = n \cdots, 2n \tag{2.28}$$

where $\mathcal{N}_i$ is the in-neighbor set of the node i and the initial conditions are given as

$$\xi_{i,j}(0) = \begin{cases} 1, & \text{if } j = i \\ 0, & \text{otherwise} \end{cases}$$

$$\omega_{i,j}(0) = \begin{cases} 0, & \text{if } j = i \\ \infty, & \text{otherwise} \end{cases}$$

$$\rho_i(n) = \begin{cases} 0 & \text{if } \xi_i(n) = \mathbf{1}_n \\ 1, & \text{if } \xi_i(n) \neq \mathbf{1}_n. \end{cases} \tag{2.29}$$

Theorem 1. *Consider a network $\mathcal{G}$ in which each node i executes two concurrent distributed n-step algorithms:* (2.26)*,* (2.27) *and the next n step algorithm* (2.28)*. Then, the following conclusions can be drawn:*

1. *Node j is one of the p^{th} in-neighbors of node i: that is, $\{j\} \in \mathcal{N}_i^{(p)}$ if $\omega_{i,j}(n) = p > 0$.*

2. *Node i has the information of every other node in the graph $\mathcal{G}$, if and only if all of $\xi_{i,j}(n)$*

elements are non zero: that is,

$$\xi_{i,j}(n) \neq 0, \quad \forall j \in \mathcal{N}. \tag{2.30}$$

3. *Each node i knows that the graph $\mathcal{G}$ is strongly connected, if and only if $\rho_i(2n) = 0$.*

Proof: At $k = 0$, $\xi_{i,j}(0)$ is set to 1 if $j = i$ and 0 otherwise, indicating each node is initially only aware of itself. Correspondingly, $\omega_{i,j}(0)$ is initialized to 0 for itself and to ∞ for all other nodes, indicating that the distance to itself is known to be zero, and the distance to any other nodes are unknown.

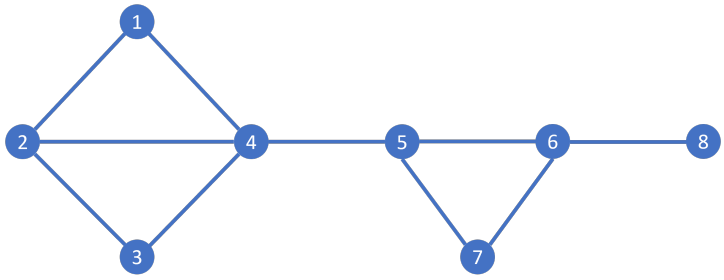
For each iteration $(k + 1)$, it follows $\xi_{i,j}(k + 1)$ updates to 1 if j is reachable from i within $k + 1$ steps, determined by neighbor l of i having $\xi_{l,j}(k) = 1$. If $\xi_{i,j}(k + 1)$ increases (indicates new reachability), $\omega_{i,j}(k + 1)$ updates to capture the shortest path distance from i to j, that is the minimum of $\omega_{l,j}(k) + 1$ across all neighbors l of i that can reach j, thereby indicating the path length.

It can be seen that $\xi_{i,j}(k)$ is binary and non-decreasing. Note that $\omega_{i,j}(k)$ will remain zero or ∞ until $\xi_{i,j}(k)$ changes from 0 to 1. In addition, it is known by induction that, if $\{j\} \in \mathcal{N}_i^{(p)}$, $\xi_{i,j}(k)$ switches from 0 to 1 precisely at step $k = p$. Afterwards, $\xi_{i,j}(k + 1) - \xi_{i,j}(k) \equiv 0$, and invariant, and so is $\omega_{i,j}$. Accordingly, the conclusion is drawn. □

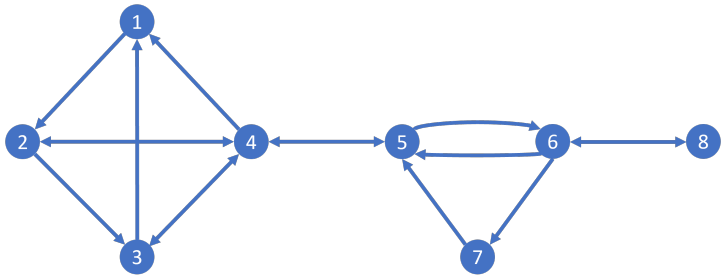
Algorithm 1 operates over n steps ($k = 1, \cdots, n$), by each node i to identify its neighbor structure in terms of its p^{th} neighbor set $\mathcal{N}_i^{(p)}$ by executing update laws (2.26) and (2.27).

The following observations are worth noting. First, network $\mathcal{G}$ is connected if $\omega_{i,j}(n) > 0$ for all $i \in \mathcal{N}$ and for all $j \neq i$. Second, if there are alternate paths between nodes i and j, the result $\omega_{i,j}(n)$ from algorithm (2.27) corresponds to the shortest path.

Numerical Example



(a) Undirected Network $\mathcal{G}_{ex_u}$



(b) Directed Network $\mathcal{G}_{ex_d}$

Figure 2.4: Example networks

To illustrate the algorithm, consider the undirected graph in Figure 2.4(a).

Example 1. *Let us apply the algorithm to identify the in-neighbor structure and shortest in-neighbor distances for each node.*

Initial Setup

For each node $i \in \mathcal{N}$, initialize the following states:

$$\xi_i(0) = [\xi_{i,1}(0) \ \cdots \ \xi_{i,n}(0)]^T \in \mathbb{R}^n$$

$$\omega_i(0) = [\omega_{i,1}(0) \ \cdots \ \omega_{i,n}(0)]^T \in \mathbb{R}^n$$

where $\xi_{i,j}(0) = 1$ *if* $j = i$*, otherwise* $\xi_{i,j}(0) = 0$*, and* $\omega_{i,j}(0) = 0$ *if* $j = i$*, otherwise* $\omega_{i,j}(0) = \infty$*.*

Given graph:

- *Nodes:* $\{1, 2, 3, 4, 5, 6, 7, 8\}$
- *Edges:* $\{(1, 2), (1, 4), (2, 3), (2, 4), (3, 4), (4, 5), (5, 6), (5, 7), (6, 8)\}$

Iterative Steps

The algorithm proceeds with n *iterations, updating* $\xi_{i,j}(k + 1)$ *and* $\omega_{i,j}(k + 1)$ *at each step.*

Let us demonstrate the first two iterations for node 4 to illustrate the process.

$$\xi_{i,j}(1) = \max_{l \in \mathcal{N}_i \cup \{i\}} \xi_{l,j}(0)$$

$$\omega_{i,j}(1) = \begin{cases} \omega_{i,j}(0) & \text{if } \xi_{i,j}(1) = \xi_{i,j}(0), \\ \min_{l \in \mathcal{N}_i}(\omega_{l,j}(0) + 1) & \text{if } \xi_{i,j}(1) > \xi_{i,j}(0) \end{cases}$$

For node 4:

- *Neighbors are nodes 1, 2, 3, and 5.*

Initial State:

- $\xi_4(0) = [0, 0, 0, 1, 0, 0, 0, 0]$
- $\omega_4(0) = [\infty, \infty, \infty, 0, \infty, \infty, \infty, \infty]$

After 1st Iteration:

$$\xi_4(1) = [\max(0,1), \max(0,1), \max(0,1), 1, \max(0,1), 0, 0, 0]$$
$$\xi_4(1) = [1, 1, 1, 1, 1, 0, 0, 0]$$
$$\omega_4(1) = [1, 1, 1, 0, 1, \infty, \infty, \infty]$$

State after 2nd Iteration:

$$\xi_4(2) = [\max(1, \max(1,0)), \cdots, \cdots, 1, \cdots, \max(1,0), \max(1,0), 0]$$
$$\xi_4(2) = [1, 1, 1, 1, 1, 1, 1, 0]$$
$$\omega_4(2) = [1, 1, 1, 0, 1, 2, 2, \infty]$$

Continuing this process for n iterations for each node, the final states will provide the in-neighbor structure and shortest in-neighbor distances.

Final State for All Nodes

- ***Node 1:***
  - $\xi_1(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_1(8) = [0, 1, 2, 1, 2, 3, 3, 4]$
- ***Node 2:***
  - $\xi_2(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_2(8) = [1, 0, 1, 1, 2, 3, 3, 4]$
- ***Node 3:***

- $\xi_3(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
- $\omega_3(8) = [2, 1, 0, 1, 2, 3, 3, 4]$

- ***Node 4:***
  - $\xi_4(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_4(8) = [1, 1, 1, 0, 1, 2, 2, 3]$

- ***Node 5:***
  - $\xi_5(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_5(8) = [2, 2, 2, 1, 0, 1, 1, 2]$

- ***Node 6:***
  - $\xi_6(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_6(8) = [3, 3, 3, 2, 1, 0, 1, 1]$

- ***Node 7:***
  - $\xi_7(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_7(8) = [3, 3, 3, 2, 1, 1, 0, 2]$

- ***Node 8:***
  - $\xi_8(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_8(8) = [4, 4, 4, 3, 2, 1, 2, 0]$

By following this process, each node in the network has identified its in-neighbors and the shortest path to them, ensuring robust and efficient communication within the network.

Example 2. *The algorithm works similarly for a directed graph, for example, consider the directed graph in Figure 2.4(b).*

For node 4:

- *In-neighbors are nodes 2, 3, and 5.*

Initial State:

- $\xi_4(0) = [0, 0, 0, 1, 0, 0, 0, 0]$
- $\omega_4(0) = [\infty, \infty, \infty, 0, \infty, \infty, \infty, \infty]$

After 1st Iteration:

$$\xi_4(1) = [0, \max(0, 1), \max(0, 1), 1, \max(0, 1), 0, 0, 0]$$
$$\xi_4(1) = [0, 1, 1, 1, 1, 0, 0, 0]$$
$$\omega_4(1) = [\infty, 1, 1, 0, 1, \infty, \infty, \infty]$$

State after 2nd Iteration:

$$\xi_4(2) = [\max(1, 0), \cdots, \cdots, 1, \cdots, \max(1, 0), \max(1, 0), 0]$$
$$\xi_4(2) = [1, 1, 1, 1, 1, 1, 1, 0]$$
$$\omega_4(2) = [2, 1, 1, 0, 1, 2, 2, \infty]$$

Continuing this process for n iterations for each node, the final states will provide the in-neighbor structure and shortest in-neighbor distances.

Final State for All Nodes

- ***Node 1:***
  - $\xi_1(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_1(8) = [0, 2, 1, 1, 2, 3, 3, 4]$
- ***Node 2:***
  - $\xi_2(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_2(8) = [1, 0, 2, 1, 2, 3, 3, 4]$
- ***Node 3:***
  - $\xi_3(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_3(8) = [2, 1, 0, 1, 2, 3, 3, 4]$
- ***Node 4:***
  - $\xi_4(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_4(8) = [2, 1, 1, 0, 1, 2, 2, 3]$
- ***Node 5:***
  - $\xi_5(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_5(8) = [3, 2, 2, 1, 0, 1, 1, 2]$
- ***Node 6:***
  - $\xi_6(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_6(8) = [4, 3, 3, 2, 1, 0, 2, 1]$

- ***Node 7:***
  - $\xi_7(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_7(8) = [5, 4, 4, 3, 2, 1, 0, 2]$
- ***Node 8:***
  - $\xi_8(8) = [1, 1, 1, 1, 1, 1, 1, 1]$
  - $\omega_8(8) = [5, 4, 4, 3, 2, 3, 1, 0]$

Instantaneous Detection Algorithm

The above algorithm can also operate in continuous time, where each agent continuously updates its estimate of in-neighbors in the current network's layers based on the maximum protocol. The estimates are initialized such that an agent is connected only to itself. Agents then update their estimates at any time t by considering information from their neighbors in the current network represented by Laplacian $\mathcal{L}^{(k)}(t)$. By observing whether any estimate $\xi_{i,j}^{(k)}(t)$ becomes zero for some pair (i, j) after a certain duration, agents can detect network disconnection. This continuous-time formulation captures the evolving network connectivity dynamics.

For agent $i \in \mathcal{N}$, consider the state $\xi_i^{(k)}(t) = [\xi_{i,1}^{(k)}(t) \ \cdots \ \xi_{i,n}^{(k)}(t)]^T \in \mathbb{R}^n$ denoting the agent i's estimate of its in-neighbors in the kth layer of the network whose Laplacian is $\mathcal{L}^{(k)}(t)$. Apply the following instantaneous detection algorithm in terms of the standard maximum protocol: for any $\delta t > 0$,

$$\xi_{i,j}^{(k)}(t + \delta t) = \max_{l \in \mathcal{N}_i^{(k)}(t) \cup \{i\}} \xi_{l,j}^{(k)}(t), \quad j \in \mathcal{N}, \tag{2.31}$$

where $\mathcal{N}_i^{(k)}$ denotes the neighbor set of agent i in the original/hidden network represented by

Laplacian $\mathcal{L}^{(k)}(t)$, and the initial condition is given by

$$\xi_{i,j}^{(k)}(t) = \begin{cases} 1, & \text{if } j = i \\ \\ 0, & \text{otherwise} \end{cases}. \tag{2.32}$$

Using (2.31) and (2.32), value $\xi_{i,j}^{(k)}(t + n\delta t)$ (after iterating n-steps) enables ith agent to determine whether it is connected to the jth agent. That is, if $\xi_{i,j}^{(k)}(t + n\delta t) = 0$ for any pair (i, j), agents i and j as well as all other agents know that Laplacian $\mathcal{L}^{(k)}(t)$ becomes disconnected.

Recalling that $\mathcal{L}^{(k)}(0) = \mathcal{L}(0)$ is connected and also taking the limit of $\delta t \to 0$ in the expression of $\xi_{i,j}^{(k)}(t + n\delta t)$ yield the following result.

Theorem 2. *Consider the following distributed, instantaneous protocol over the time-varying network of Laplacian $\mathcal{L}^{(k)}(t)$:*

$$\xi_{i,j}^{(k)}(t) = \max_{l \in \mathcal{N}_i^{(k)}(t) \cup \{i\}} \xi_{l,j}^{(k)}(t), \quad i, j \in \mathcal{N}, \tag{2.33}$$

where $\xi_{i,i}^{(k)}(t) = 1$. Given that $\mathcal{L}^{(k)}(0) = \mathcal{L}(0)$ is connected, the graph of Laplacian $\mathcal{L}^{(k)}(t)$ becomes disconnected at time $t > 0$ if and only if $\xi_{i,j}^{(k)}(t) = 0$ for some pair (i, j) at time t.

No matter where, how many, and when DoS attacks are launched on the graph of Laplacian $\mathcal{L}^{(k)}(t)$, algorithm (2.33) provides an instantaneous detection of $\mathcal{L}^{(k)}(t)$ becoming disconnected. Hence, it can be assumed without loss of any generality that $t_d^{(k)}$ be the time instant when loss of connectivity is detected for Laplacian $\mathcal{L}^{(k)}(t)$.

Delays in Teleoperation System

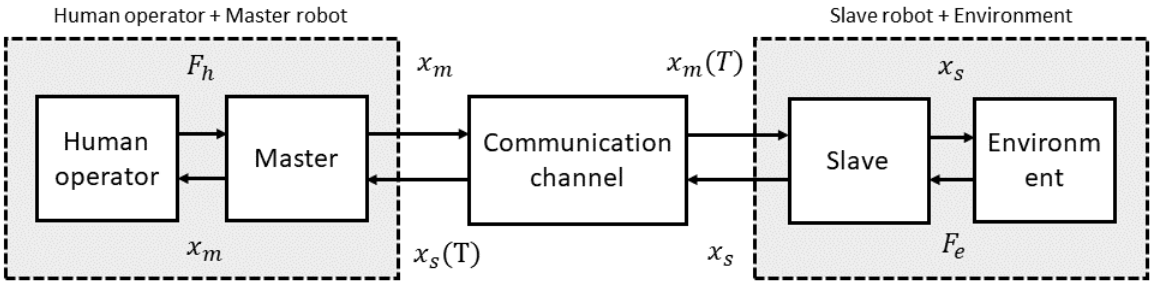


Figure 2.5: Architecture of bilateral teleoperation

Teleoperation is a negative feedback configuration of robotic systems with delayed communication channels as shown in Figure 2.5. In a force-position-based teleoperation cycle, the following steps take place: the master system receives an operator force F_h; the slave system receives a delayed version of the transmitted signal from the master (torque τ_m); the slave system which is subjected to an environmental force (F_e), sends its feedback (x_s) back to the master. Since passivity is a special case of passivity-short systems, the master/slave systems can be passive (if the output/input is velocity) or passivity-short. The master robot sends its state as well as the received slave state back to the operator. The communication channels back and forth may have different delays, which may vary with respect to time, which is the main issue that the proposed feedback control design should tackle.

Resilience against DoS attacks

Consider a multi-agent system Σ_s composed of n agents, each governed by the individual dynamics:

$$\dot{x}_i = u_i, \quad i = \{1, \ldots, n\}, \tag{2.34}$$

where $x_i \in \mathbb{R}$ represents the state of agent i, and u_i is the control input. The agents communicate through a network of graph $\mathcal{G}$, with the goal of achieving consensus as: for some $c \neq 0$,

$$x_i \to c, \quad \forall i. \tag{2.35}$$

The standard consensus algorithm employed by each agent is:

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i), \quad a_{ij} \geq 0, \tag{2.36}$$

which results in the overall system dynamics:

$$\dot{x} = -Lx, \tag{2.37}$$

where $A = [a_{ij}]$ is the adjacency matrix of graph $\mathcal{G}$, $\mathbf{1}$ is the vector of 1s, $D = A\mathbf{1}$, and $\mathcal{L} = D - A$ is the Laplacian matrix. If $\mathcal{G}$ is connected, the states of all agents will converge to the average of their initial conditions as

$$x \to = \left(\frac{1}{n} \sum_{i=1}^{n} x_i(0) \right) \mathbf{1}_n. \tag{2.38}$$

DoS attacks pose a significant threat to the functionality and reliability of multi-agent systems by targeting agents' communication network. When launching a DoS attack, malicious actors flood certain communication channels with overwhelming traffic, rendering them inoperable. These

communication disruptions captured by a time-varying Laplacian matrix $\mathcal{L}(t)$ make the consensus algorithm less effective due to compromised communication links. Should DoS attacks become persistently intensified, they progressively degrade connectivity and eventually lead to disintegration of graph $\mathcal{G}$. This loss of connectivity transforms Laplacian matrix $\mathcal{L}$ into a reducible matrix and results in multiple, local consensus values, despite that the standard consensus algorithm (2.37) aims to connect all the agents and achieve a common consensus.

The problems that needs to be addressed revolve around enhancing the robustness and resilience of networked systems, particularly in the face of vulnerabilities and cyber threats. Both directed and undirected networks contain critical edges, or bridges, whose removal results in network disconnection, thus exposing significant points of vulnerability. Efficient identification of these critical edges is paramount for maintaining network robustness and ensuring uninterrupted connectivity. However, existing methodologies for identifying and reinforcing these critical edges are predominantly centralized. This centralization poses scalability issues and creates single points of failure, particularly problematic in large, dynamic networks. Consequently, there is a pressing need for scalable, distributed algorithms capable of functioning effectively across extensive networks without reliance on a central authority.

In addition to structural vulnerabilities, multi-agent systems, which are widely utilized in applications such as robotics, smart grids, and communication networks, are highly susceptible to DoS attacks. These attacks disrupt critical communication channels by overwhelming them with traffic or blocking them entirely, leading to significant connectivity losses and impaired system performance. Thus, developing resilient control strategies and algorithms that can detect and mitigate the effects of DoS attacks in real-time is crucial for maintaining the operational integrity and performance of these multi-agent systems.

The task of identifying and reinforcing critical edges in large-scale networks is computationally

demanding. To address this challenge, it is necessary to simplify and reduce large graphs into smaller, more manageable subgraphs while preserving their essential connectivity properties. This reduction facilitates more efficient analysis and enhancement of network robustness, enabling the deployment of distributed algorithms that can operate autonomously across the network. By addressing these interconnected problems, the goal is to develop a comprehensive approach that enhances the resilience and robustness of networked systems against structural vulnerabilities and cyber threats.

Objectives of the Dissertation

In summary, the problems addressed in this dissertation are as follows:

In a teleoperation system with master and slave robots

1. The first objective is to ensure that the overall teleoperation system remains input-to-state stable with respect to the operator's input force. This stability must be maintained despite the presence of finite time-varying delays in the communication channels. Input-to-state stability is crucial for ensuring that the behavior of the overall system remains bounded and well-behaved. This includes scenarios involving multiple masters and slaves, where each component's stability contributes to the system's overall robustness.

2. The second objective is to design controllers that can maintain system stability even if the communication channels are interrupted. Communication interruptions can occur due to various factors, such as network failures, signal losses, or intentional interference. It is crucial to ensure that the teleoperation system remains stable and does not exhibit undesirable behavior when communication channels are temporarily disrupted. This requires the development of robust control strategies that can adapt to varying levels of communication quality.

3. Since the operator directly controls the master device and aims to control the slave system through delayed channels, it is important to minimize the lag and errors between the outputs of the master and slave systems. The proposed control strategies should aim to reduce any discrepancies between the desired and actual positions or forces experienced by the slave system. Minimizing these discrepancies allows for more accurate and responsive teleoperation, enhancing the operator's ability to perform precise tasks.

4. The fourth objective is to maintain L_2 stability of the overall system, even when the environment in which the slave system operates is unstable. L_2 stability guarantees that the system's energy remains bounded, which is essential for the safe and reliable operation of the teleoperation system. Despite any disturbances or uncertainties in the environment, the control design should ensure that the overall system remains stable and does not exhibit uncontrolled or excessive behavior. This requires robust control mechanisms that can handle environmental variability.

In a multi-agent system with possible DoSattacks

1. There is a pressing need for scalable, distributed algorithms capable of functioning effectively across extensive networks without reliance on a central authority. These algorithms must autonomously identify and reinforce critical edges to enhance network robustness and eliminate single points of failure. The ability to scale and operate in a decentralized manner is crucial for large, dynamic networks where central control is impractical or undesirable. Effective decentralized algorithms can enhance network resilience by ensuring that individual nodes can make informed decisions independently.

2. Multi-agent systems, widely used in applications such as robotics, smart grids, and communication networks, are highly susceptible to DoS attacks. These attacks can disrupt critical

communication channels by overwhelming them with traffic or blocking them entirely, leading to significant connectivity losses and impaired system performance. Developing real-time control strategies and algorithms that detect and mitigate the effects of DoS attacks is crucial for maintaining the operational integrity and performance of these systems. This includes creating detection mechanisms that can identify the onset of an attack and response strategies that can reconfigure the network to maintain functionality.

3. Identifying and reinforcing critical edges in large-scale networks is computationally demanding. To address this challenge, it is necessary to simplify and reduce large graphs into smaller, more manageable subgraphs while preserving their essential connectivity properties. This reduction facilitates more efficient analysis and enhancement of network robustness. By breaking down large networks into simpler components, it becomes easier to deploy distributed algorithms that can operate autonomously across the network. This process involves techniques for graph partitioning and reduction that maintain the network's structural integrity while enabling more efficient processing and analysis.

CHAPTER 3: STABILITY AND PERFORMANCE IN BILATERAL AND MULTILATERAL TELEOPERATION WITH TIME-VARYING DELAYS

Introduction

Teleoperation systems, which enable human operators to control remote robotic systems, play a crucial role in various applications such as surgery, space exploration, and hazardous environment operations. These systems can be categorized into bilateral teleoperation, where a single master controls a single slave, and multilateral teleoperation, involving multiple masters and slaves. A fundamental challenge in teleoperation systems is the presence of time-varying delays in communication channels, which can significantly affect system stability and performance.

In bilateral teleoperation, the master and slave systems exchange position and force information through communication channels. The inherent delays in these channels can lead to instability, oscillations, and degraded performance, making it critical to design control strategies that mitigate these effects. Multilateral teleoperation extends this complexity by involving multiple agents, requiring robust control mechanisms to ensure coordinated and stable operation despite communication delays.

This chapter addresses the challenges associated with time-varying delays in teleoperation systems, focusing on both stability and performance aspects. The concept of passivity-short systems is explored as a framework for analyzing and designing control strategies that ensure stability under varying delay conditions. Passivity-short systems, a generalization of passive systems, provide a less restrictive yet powerful approach to maintaining system stability and performance.

Our main objective in this chapter is to develop a teleoperation framework that ensures both sta-

bility and high-performance position tracking, going beyond the limitations of passivity-based approaches. This work builds upon the findings of our previous conference version [33], but now takes into account the impact of time-varying communication delays and robot-environment interactions. The key highlights of the results presented in this chapter are as follows:

1. Analyzes stability conditions for teleoperation systems with time-varying delays, showing that passivity-short control ensures L_2 stability and robust performance. The framework includes PD control, gravity compensation, and specific gain adjustments to maintain stability and minimize phase lag between master and slave robots [33, 76].

2. Extends passivity-short control to multilateral teleoperation with multiple master and slave agents. It introduces a scalable control design using leader-follower consensus protocols and statistical feature control to achieve input-to-state stability despite delays, enhancing system flexibility and resilience in real-world applications [77].

3. Experimental validation using Phantom Omni devices supports the effectiveness of our proposed approach. The results indicate that our framework exhibits reduced chattering and improved convergence of steady-state errors compared to existing approaches.

The control design problem is to synthesize both master and slave controllers to meet the following objectives:

- Ensure input-to-state stability of the overall system with respect to the operator input, even in the presence of finite time-varying delays in the communication channels.
- Maintain system stability even in cases where the communication channels are interrupted.
- Minimize the lag and errors between the outputs of the master and slave systems, taking into

account the direct control of the master device by the operator and the delayed control of the slave system.

- Maintain L_2 stability of the overall system, even when the environment is not stable.

In the subsequent section, the aforementioned negative feedback configuration is extended to a multi-loop feedback configuration for bilateral teleoperation, as depicted in Figure 2.5.

Stability Analysis and Performance Conditions for Teleoperation Systems

In this section, stability analysis and performance conditions are discussed for a teleoperation configuration. The master and slave robots are considered to be passivity-short, with individual PD control and gravity compensation and the communication channel is assumed to have a time-varying delay. The slave robot is subjected to an environmental force, and the environment is considered to be passivity-short. It is shown that the overall system is passivity-short and L_2 stability is achieved under certain conditions.

Consider passivity-short mappings $\mathcal{P}_m$ with input v_m and output x_m, and $\mathcal{P}_s$ with input v_s and output x_s to represent the master and slave systems, with the rigid body dynamics (2.10), and for positive definite storage function (2.17), they are L_2 stable with parameters $[\varrho_m, \epsilon_m], [\varrho_s, \epsilon_s]$ respectively.

The bilateral teleoperation configuration can be represented by a master and slave negative feedback design. Figure 3.1 shows the negative feedback representation of teleoperation with additional simple individual position-only negative feedback for the master and slave systems to improve the performance of the overall system and to handle the potential instability issues already discussed in previous sections.

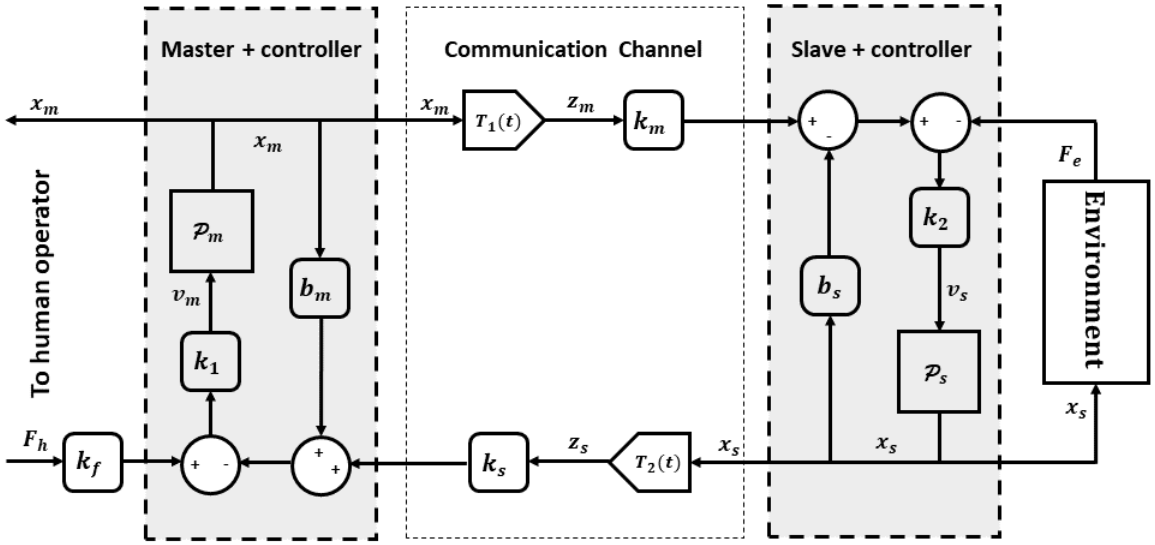


Figure 3.1: Proposed passivity-shortage based framework

The control inputs v_m and v_s can be written as:

$$v_m(t) = k_1 \left(k_f F_h(t) - k_s z_s(t) - b_m x_m(t)\right), \tag{3.1a}$$

$$v_s(t) = k_2 \left(k_m z_m(t) - b_s x_s(t)\right) - F_e(t), \tag{3.1b}$$

where b_m and b_s are individual feedback gain for the master and slave systems respectively; k_m and k_s are communication channel gains; and k_1 and k_2 are additional gains introduced to improve the performance characteristics, and k_f is the input scaling factor. The variables $z_m(t) = x_m(t-T_m(t))$ and $z_s(t) = x_s(t - T_s(t))$ denote the varying-time delayed outputs.

The following theorem outlines the stability results of this system based on the passivity-short properties (which include passive systems as a special case).

Theorem 3. *Consider passivity-short mappings $\mathcal{P}_m$, $\mathcal{P}_s$. The overall system in Figure 3.1 with control input* (3.1a) *and* (3.1b)*, is passivity-short and L_2 stable, from input $[F_h, F_e]$, to output*

$[x_m, x_s]$, *and new weights* $\{\epsilon', \varrho'\}$, *if Assumption 1 holds, and if there exists positive gains and arbitrary constants* c_1, c_2 *and* c_3, *such that:*

$$
\begin{aligned}
0 &\leq \inf_t \left[\frac{\varrho_m}{2} + k_1 b_m - \frac{3 b_m^2 \epsilon_m k_1^2}{2} - \frac{c_1}{2} - c_m \right] \triangleq \varrho'_m, \\
0 &\leq \inf_t \left[\frac{\varrho_s}{2} + k_2 b_s - \frac{3 b_s^2 \epsilon_s k_2^2}{2} - c_2 - \frac{c_s}{2} \right] \triangleq \varrho'_s, \\
\epsilon'_m &= \left[\frac{3}{2} \epsilon_m k_1 k_f \right], \qquad \epsilon'_s = \left[\frac{3}{2} \epsilon_s k_2^2 \right].
\end{aligned} \tag{3.2}
$$

The weights c_m, c_s associated with the delay channel are chosen as:

$$
\begin{aligned}
c_m &\geq \frac{1}{(1 - \dot{T}_{1,max})} \left(3 \epsilon_s k_2^2 k_m^2 + \frac{k_2^2 k_m^2}{c_2} \right), \\
c_s &\geq \frac{1}{(1 - \dot{T}_{2,max})} \left(3 \epsilon_m k_1^2 k_s^2 + \frac{k_1^2 k_s^2}{c_1} \right).
\end{aligned} \tag{3.3}
$$

To achieve the above condition, and in-turn L_2 stability of the overall system, gains k_m, k_s, b_m, b_s, k_1, k_2, k_f need to be picked, such that (3.2) and (3.3) are satisfied.

Proof: Consider the following storage function L_t:

$$
L_t = \frac{1}{k_1 k_f} V_m(t) + V_s(t) + V_{d_m}(t, T_1) + V_{d_s}(t, T_2),
$$

where $V_m(t)$ and $V_s(t)$ are the storage functions in the form of (2.17) (i.e., by replacing subscript $_i$ by $_m$ or $_s$), and $V_{d_m}(t, T_1)$ and $V_{d_s}(t, T_2)$ are delay channel storage functions in the form of (2.21),

with associated positive constants c_m and c_s. Then, the time derivative of L_t becomes

$$\begin{aligned}\dot{L}_t \leq \frac{1}{k_1 k_f}\Big[&v_m^T(t)x_m(t) + \frac{\epsilon_m}{2}\|v_m(t)\|^2 - \frac{\varrho_m}{2}\|x_m(t)\|^2\Big] \\ &+ v_s^T(t)x_s(t) + \frac{\epsilon_s}{2}\|v_s(t)\|^2 - \frac{\varrho_s}{2}\|x_s(t)\|^2 \\ &+ \frac{c_m}{2}\Big[\|x_m(t)\|^2 - \|z_m(t)\|^2(1-\dot{T}_1)\Big] \\ &+ \frac{c_s}{2}\Big[\|x_s(t)\|^2 - \|z_s(t)\|^2(1-\dot{T}_2)\Big].\end{aligned}$$

Substituting the proposed controllers in (3.1a) and (3.1b) into the above yields

$$\begin{aligned}\dot{L}_t \leq\ & F_h^T x_m + F_e^T x_s + \frac{3\epsilon_m k_1 k_f}{2}\|F_h\|^2 + \frac{3\epsilon_s k_2^2}{2}\|F_e\|^2 \\ &- \left[\frac{\varrho_m}{2} + k_1 b_m - \frac{3b_m^2\epsilon_m k_1^2}{2} - c_1 - c_s\right]\frac{\|x_m\|^2}{k_f k_1} \\ &- \left[\frac{\varrho_s - \varrho_e}{2} + k_2 b_s - \frac{3b_s^2\epsilon_s k_2^2}{2} - c_2 - c_m\right]\|x_s\|^2 \\ &- \left[\frac{c_m}{2}(1-\dot{T}_{1,max}) - \frac{k_2^2 k_m^2}{c_2} + \frac{3}{2}\epsilon_s k_2^2 k_m^2\right]\|z_m\|^2 \\ &- \left[\frac{c_s}{2k_f k_1}(1-\dot{T}_{2,max}) - \frac{3}{2k_f}\epsilon_m k_1 k_s^2 + \frac{k_1 k_s^2}{c_1 k_f}\right]\|z_s\|^2.\end{aligned}$$

It follows from passivity-shortage parameters ϵ_m, ϵ_s and ϱ_m, ϱ_s in (3.2) and from delay channel parameters in (3.3) that passivity-shortage is established as

$$\dot{L}_t \leq u^T y + \frac{\epsilon'}{2}\|u\|^2 - \frac{\varrho'}{2}\|y\|^2,$$

where $u = [F_h, F_e]$ and $y = [x_m, x_s]$. Positive values of the parameters $\epsilon' = [\epsilon'_m, \epsilon'_s]$, $\varrho' = [\varrho'_m, \varrho'_s]$ are ensured by (3.2). Thus, passivity-shortage and L_2 stability of the overall system is proved.

However, in addition to achieving stability, teleoperation needs improved performance such as minimum error between the output of the master and slave systems, and minimum phase lag. Such

performance improvements are model-specific. This is achieved by ensuring a unity DC gain in the closed-loop transfer function between the master and slave subsystems, as discussed in [33]. The gains can be chosen using a simple iterative search under the above conditions.

It is also interesting to note, Theorem 3 implies that the overall system is passivity-short, but not passive because the gains k_1 and k_f are positive, hence ϵ'_m and ϵ'_s are positive.

Application to Multilateral Teleoperation

Passivity-shortage framework can be extended to delayed multilateral teleoperation with distributed multi-agent systems. The goal is to achieve consensus or dispersion among slave agents based on operator commands, using features of their output probability distribution rather than direct position or velocity control.

Figure 3.2 shows the architecture of the multilateral teleoperation setup, with master and slave systems, communication channels, and human operator commands. Slave agents communicate through an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Master agents send commands to and receive feedback from slave agents over delayed communication channels.

The objective is to design a controller that ensures input-to-state stability despite delays and disturbances, for various configurations of master commands and slave agents. Instead of controlling individual agents directly, the focus is on controlling statistical features of the output distribution. This approach allows for scalability and flexibility in the number of slave agents.

The control design involves propagating desired features to all agents using a leader-follower consensus protocol [78]. Stability and convergence are analyzed using the properties of passivity-shortage [73, 74]. The master controllers are designed to ensure L_2 stability and independence

from communication delays. By controlling the statistical features of the slave agents' output distribution, the proposed method removes the limitation on the number of slave agents and achieves scalable and flexible control.

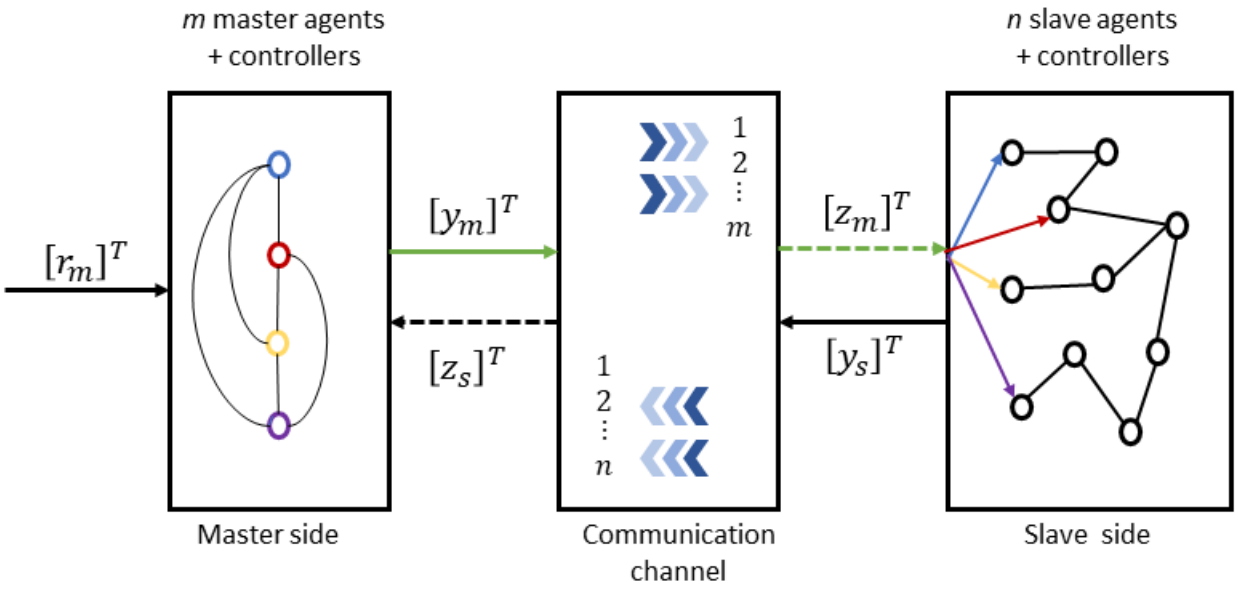


Figure 3.2: Architecture of multilateral teleoperation

Control Design

Figure 3.3 illustrates the control design of the overall system. The operator input is denoted as r_m, and the controller $y_m = [y_{m_q}, y_{m_\sigma}, y_{m_{min}}, y_{m_{max}}]$ in the master side is designed using passivity-shortage properties. The individual components $\mathcal{P}m_k, k \in 1, 2...m$ of the master controller are both passivity-short and L_2 stable. The master feedback loop includes gain matrices

$$g_m = [gm_1, g_{m_2}..., g_{m_m}]$$

and

$$k_m = [k_{m_1}, k_{m_2}..., k_{m_m}],$$

respectively.

The communication channel introduces delays, which are represented by the s-domain function e^{-sT_k} with delays T_k from the master side to the slave side, and T_i in the opposite direction.

Rather than controlling the output of the robot agents, this paper proposes an approach to control specific features of the robots, represented as cooperative states x_s. This removes the limitation on the maximum number of robots on the slave side.

Each individual agent $\mathcal{P}s_i, i \in 1, 2...n$ on the slave side is considered passivity-short. The controller for the slave robot consists of negative feedback loops with gains $gs = [g_{s_1}, g_{s_2}...g_{s_n}]$ and $k_s = [k_{s_1}, k_{s_2}...k_{s_n}]$. The delayed output from the slave robot z_s is fed back to the master robot. Additional gains k_p and k_d ensure appropriate damping and unity DC gain. Assuming that the environmental force is zero, the input and output can be expressed as $v = [r_m, 0]$ and $y = [y_m, y_s]$.

The following theorem establishes the passivity and passivity-short properties of this system, along with the convergence of the error $e = y_s - r_m$.

Theorem 4. *Consider the figure 3.2, with m master side controllers and n slave side agents, and input-output pairs $[u_m, y_m]$, $[u_s, y_s]$ respectively, that are passivity-short with master/slave parameters ($\epsilon_{m_i}, \varrho_{m_i} > 0$), ($\epsilon_{s_i}, \varrho_{s_i} > 0$): $i = 1, 2..m$. Then,*

- *The overall system is passivity-short and $\mathcal{L}_2$ stable, from the input r_m to the output y_s, if there exists positive values for the gains for every agent i and controller k, such as:*

  (i) $k_{p_k}, k_{d_i} > 0$

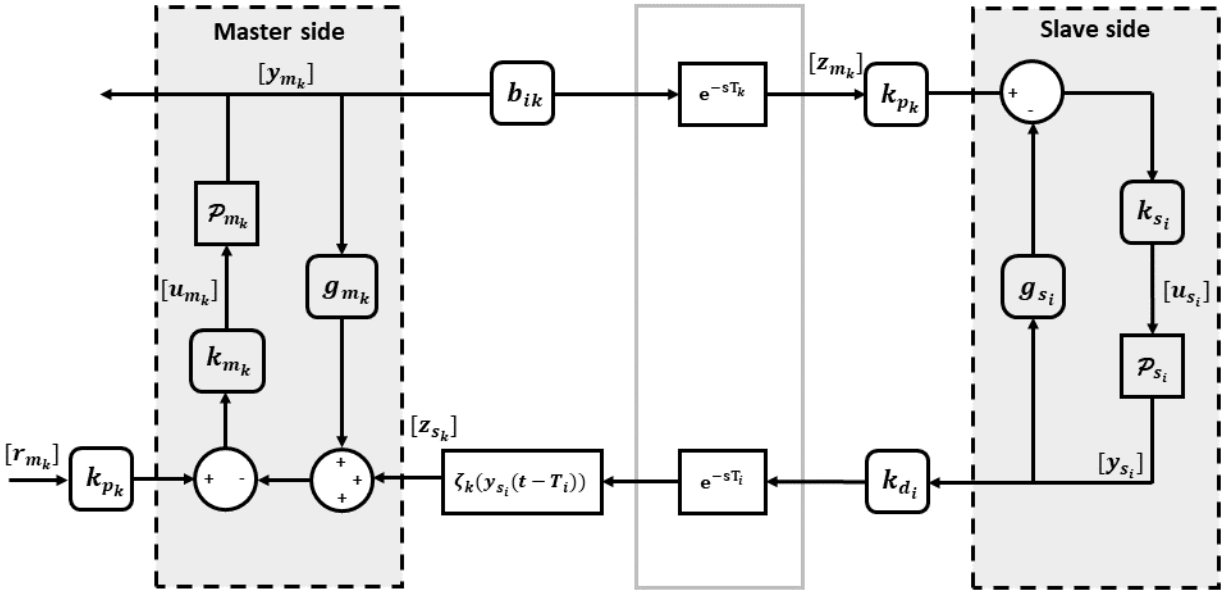


Figure 3.3: Controller design for multilateral teleoperation

(ii) $k_{m_k}, k_{s_k} > 0$ *and*

$$
\begin{aligned}
k_{m_k} &\in \left(0, \frac{2g_{m_i} - \frac{1}{k_{p_i}}\left(1 + 2k_{s_i}^2\epsilon_{s_i}\right)}{3b_{m_i}^2\epsilon_{m_k}}\right), \\
k_{s_i} &\in \left(0, \frac{g_{s_i} - \frac{k_{d_i}^2}{2}\left(3\epsilon_{m_k}k_{m_k} + 1\right)}{g_{s_i}^2\epsilon_{s_i}}\right), \\
g_{m_k} &> \frac{1}{k_{p_k}}\left(\frac{1}{2} + k_{s_i}^2\epsilon_{s_i}\right),
\end{aligned} \tag{3.4}
$$

then the overall system is L_2 *stable and independent of the time delays* T_1 *and* T_2*.*

- *Additionally, if the overall system is incrementally passivity-short, then the error* e *asymptotically converges to zero, such that,* $\lim_{t\to\infty} \|e\| = 0$

Proof: The input u_{m_k} to the master controller i and the input u_{s_i} to the slave feature i are given by,

$$u_{m_k} = k_{m_k}\left(k_{p_k} r_{m_k} - g_{m_k} y_{m_k} - z_{s_i}\right),$$
$$u_{s_i} = k_{s_i}\left(z_{m_i} - g_{s_i} z_{s_i}\right).$$

where $z_{s_i} = (f^n_{k_{i=1}}(k_{d_i} y_{s_i}(t - T_i))$ and $z_{m_k} = y_{m_k}(t - T_k)$ are the delayed outputs of the master controller k and slave agent i. Consider a storage function V given by,

$$V(t) = \bar{k}_p \left[\sum_{k=1}^{m} \frac{V_{m_k}(t)}{k_{m_k}} + \sum_{i=1}^{n} \frac{V_{s_i}(t)}{k_{s_i}} + V_d(t,T)\right],$$

where $\bar{k}_p = [\frac{1}{k_{p_1}}, \frac{1}{k_{p_2}}, ...\frac{1}{k_{p_m}}]$ is the element-wise inverse of the matrix k_p, $V_{m_k}(t), V_{s_i}(t)$ are the storage function of master controller k and slave agent i respectively, and $V_d(t,T)$ is the total storage function of the delay channel, and it is given by,

$$V_d(t,T) = \sum_{k=1}^{m}\sum_{i=1}^{n}\left[k_{p_k} b_{ik}\left(\epsilon_{s_i} k_{s_i} + \frac{1}{\epsilon''}\right)\|y_{m_k}(\tau)\|^2 d\tau + k^2_{d_i}\left(\frac{1}{\epsilon'} + \frac{3\epsilon_{m_k}}{2} k_{m_k}\right)\|\zeta(y_{s_i})(\tau)\|^2 d\tau\right]$$

Since the master controller is designed to be passivity-short, the time derivative of $V_{m_k}(t)$ along the system trajectories is the equivalent of (2.5). From Lemma 3, the slave agents follow the relation,

(2.5). Therefore,

$$\begin{aligned}
\dot{V} \leq & \dot{V}_d + \sum_{k=1}^{m} r_{m_k} y_{m_k} + \sum_{k=1}^{m} \frac{3\epsilon_{m_k}}{2} r_{m_k}^T \Big[\bar{k}_{m_k} \bar{k}_{p_k}\Big] r_{m_k} \\
& - \sum_{k=1}^{m} y_{m_k}^T \bar{k}_{p_k} \Big[g_{m_k} - \epsilon' - \frac{3\epsilon_{m_k}}{2} k_{m_k} g_{m_k}^2 + \bar{k}_{m_k} \frac{\varrho_{m_k}}{2}\Big] y_{m_k} \\
& + \sum_{k=1}^{m} \zeta_k(z_s(T))^T \bar{k}_{p_k} [\frac{1}{\epsilon'} + \frac{3\epsilon_{m_k}}{2} k_{m_k}\Big] \zeta_k(z_s(T)) \\
& - \sum_{k=1}^{m} \bar{k}_{p_k} \sum_{i=1}^{n} y_{s_i}^T \Big[g_{s_i} + \frac{\varrho_{s_i}}{2} \bar{k}_{s_i} - \epsilon'' - \epsilon_{s_i} k_{s_i} g_{s_i}^2\Big] y_{s_i} \\
& + \sum_{i=1}^{n} \sum_{k=1}^{m} b_{ik} \left(z_{m_k}(T)^T \bar{k}_{p_k} \Big[\epsilon_{s_i} k_{s_i} + \frac{1}{\epsilon''}\Big] z_{m_k}(T)\right), \\
\leq & r_m^T y_m + r_m^T \frac{\epsilon}{2} r_m - y_m^T \frac{\beta_m}{2} y_m - \zeta(y_s)^T \frac{\beta_s}{2} \zeta(y_s).
\end{aligned} \tag{3.5}$$

where $\beta_m = [\beta_{m_k}], k \in [1, 2...m]$, $\beta_s = [\beta_{s_i}], i \in [1, 2...n]$ and $\epsilon = [\epsilon_k]$ and

$$\begin{aligned}
\beta_{m_k} &= \bar{k}_{p_k} \left[g_{m_k} - \epsilon' - \frac{3\epsilon_{m_k}}{2} k_{m_k} g_{m_k}^2 + \bar{k}_{m_k} \frac{\varrho_{m_k}}{2} \sum_{i=1}^{n} b_{ik} \left(\epsilon_{s_i} k_{s_i} + \frac{1}{\epsilon''} \right) \right], \\
\beta_s &= \bar{k}_{p_k} \left[\sum_{i=1}^{n} \left(g_{s_i} + \frac{\varrho_{s_i}}{2} \bar{k}_{s_i} - \epsilon'' - \epsilon_{s_i} k_{s_i} g_{s_i}^2 \right) + \frac{1}{\epsilon'} + \frac{3\epsilon_{m_k}}{2} k_{m_k} \right], \\
\epsilon_k &= \frac{3\epsilon_m k_p k_m}{2}.
\end{aligned}$$

If the gains are chosen according to (3.4), then the overall system is passivity-short and L_2 stable from input r_m to output y_s. All parameters of the above system are time-varying, but the stability is independent of the value of delays T_1 and T_2.

An application with an illustrative example is presented in the paper [77]. Readers are encouraged to refer to this paper for a comprehensive understanding.

Performance Evaluation from Numerical Simulations

This section provides a numerical assessment for the constant delay case, where there is no rate of change of delay.

To evaluate the performance of the proposed model, it is compared to one of the existing bilateral teleoperation methods called the scattering/wave-variable transformation. In scattering transformation, the input (u_i, u_j) and output signals (y_i, y_j) are converted into wave signals based on the following linear transformation:

$$u_m = \frac{1}{\sqrt{2b}}[u_i + by_i]; \qquad v_m = \frac{1}{\sqrt{2b}}[u_i - by_i];$$
$$u_s = \frac{1}{\sqrt{2b}}[u_j + by_j]; \qquad v_s = \frac{1}{\sqrt{2b}}[u_j - by_j];$$

These transformed wave signals are sent through the communication channels. Application of scattering transformation in bilateral teleoperation is studied in [13, 79, 80].

First, the proposed model is shown to be stable for passivity-short systems. Then, the performance characteristics with passive systems are compared for both the proposed model and the scattering transformation. It is shown that the proposed model has better results in terms of phase-lag and overshoot.

Robotic systems with dynamics (2.10) can be approximated to a linear system with a transfer function of relative degree two, with respect to the joint displacement q, and of relative degree one, with respect to joint velocity $\dot{q}$ [72]. The slave robot is a duplicate of the master robot.

In Figure 3.1, the closed-loop transfer function $\bar{G}_m$ between the output of the master robot (y_m) and the reference input (v) and the closed-loop transfer function $\bar{G}_s$ between the output of the slave

robot (y_s) and the reference input (v) are as follows:

$$\bar{G}_m(s) = \frac{k_m P_m (1 + k_s b_s P_s)}{1 + G + \frac{k_d}{k_p} k_m k_s P_m P_s e^{-sT_1} e^{-sT_2}}, \tag{3.6}$$

$$\bar{G}_s(s) = \frac{k_m k_s P_m P_s e^{-sT_1}}{1 + G + \frac{k_d}{k_p} k_m k_s P_m P_s e^{-sT_1} e^{-sT_2}}, \tag{3.7}$$

where $G = k_m b_m P_m + k_s b_s P_s + k_m k_s b_m b_s P_m P_s$.

For the scattering transformation, the closed-loop transfer function $\bar{H}_m$ between the master output (y_m) and the reference input (v) and the closed-loop transfer function $\bar{H}_s$ between the slave output (y_s) and the reference input (v) are as follows:

$$\bar{H}_m(s) = \frac{P_m \left[(1 + \frac{P_s}{b}) + e^{-sT_1} e^{-sT_2} (1 - \frac{P_s}{b}) \right]}{P_1 + e^{-sT_1} e^{-sT_2} P_2}, \tag{3.8}$$

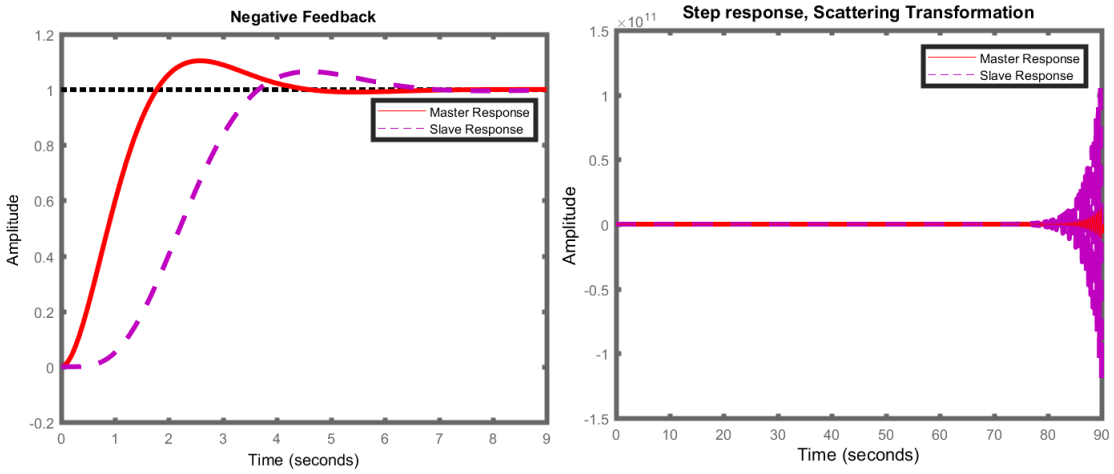
$$\bar{H}_s(s) = \frac{2 P_m P_s e^{-sT_1}}{P_1 + e^{-sT_1} e^{-sT_2} P_2}, \tag{3.9}$$

where $P_1 = (1 + \frac{P_s}{b})(1 + P_m b)$ and $P_2 = (1 - \frac{P_s}{b})(1 - P_m b)$.

Stability and Performance Evaluation for Passivity-Short Systems

This section analyzes the step response of the proposed model with passivity-short master and slave robots, followed by the scattering transformation with the same type of robots. To ensure a fair comparison between the proposed negative feedback and scattering transformation, identical master and slave systems (P_m and P_s) are used. Both systems utilize a linear passivity-short transfer function given by:

$$P_m = P_s = \frac{1}{(s+1)^2}$$



(a) Negative Feedback (b) Scattering Transformation

Figure 3.4: Comparison of step responses for passivity-short systems

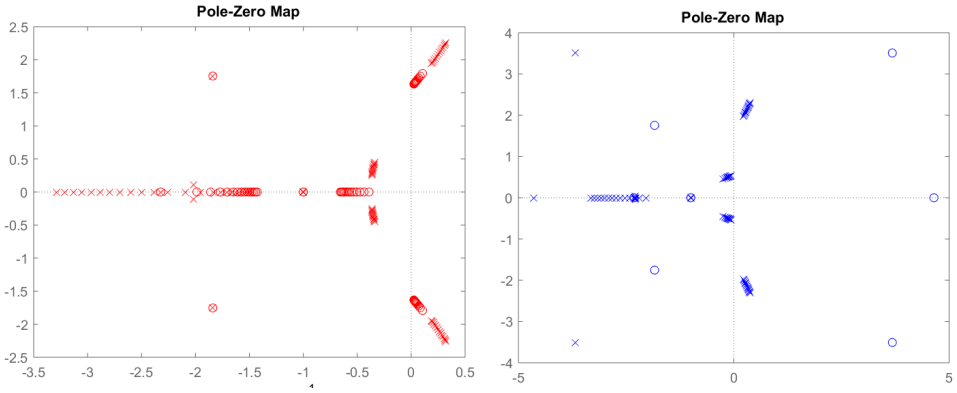
Negative Feedback Results: Based on the data illustrated in Figure 3.4a, it is apparent that the transient response of the negative feedback system is stable. This stability is demonstrated by the closed-loop transfer function in (3.7). To further improve the performance of the system, the gains k_p and k_d can be adjusted to counteract overshoot caused by additional zeros. Additionally, by adjusting the gains b_i and b_j, the poles can be shifted to the left, effectively reducing the settling time. This, in turn, minimizes the phase lag and leads to an overall improved system response.

Scattering Transformation Results: Figure 3.4b illustrates the transient response of the scattering transformation. It should be noted that the closed-loop transfer function of this transformation, as

Table 3.1: Simulation results: settling time and $\%$ overshoot for master and slave outputs

Method/Delay	Settling Time (s)			$\%$ Overshoot		
	100ms	500ms	1000ms	100ms	500ms	1000ms
	Master Robot (Y_m)					
Negative Feedback	1.3176	1.1795	3.2720	0.3248	1.8309	3.6393
Scattering Transformation	4.9212	11.6016	21.9442	35.6547	58.8277	69.6672
	Slave Robot (Y_s)					
Negative Feedback	2.9691	3.0857	3.3931	0.0000	0.1180	1.6185
Scattering Transformation	5.8584	10.3346	20.9085	3.0325	0.1896	0.1129

specified in (3.9), is not stable. It is worth mentioning that the value of the scattering wave variable b does not have any impact on the stability of the scattering transformation. This implies that poor damping is not the reason behind the instability. Furthermore, the pole-zero map of the open loop poles of the scattering transformation, as seen in Figure 3.5, clearly shows that there is always at least one pole present in the right half plane, which ultimately makes the scattering transformation unstable.



(a) Master Output (b) Slave Output

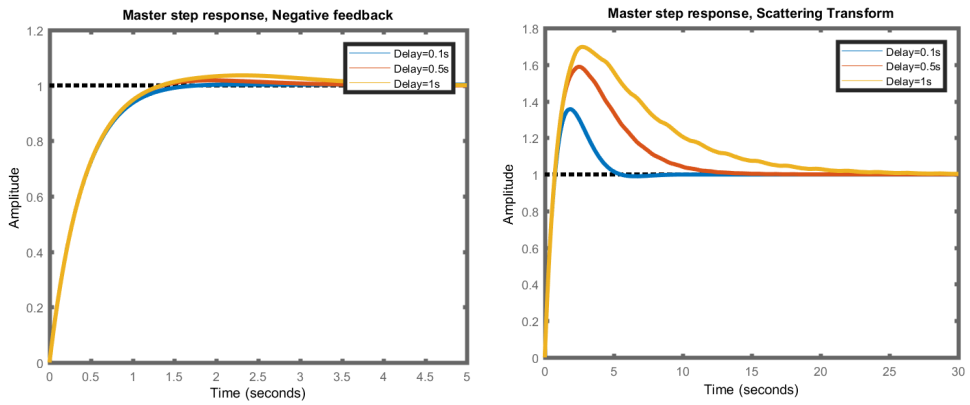
Figure 3.5: Pole-zero map of passivity-short systems under scattering transformation

Table 3.2: Simulation results: % amplitude error and phase lag for master and slave outputs with sinusoidal input

Method/Delay	% Amplitude Error			Phase Lag (rad)		
	100ms	500ms	1000ms	100ms	500ms	1000ms
	Master Robot (Y_m)					
Negative Feedback	1.62	0.02	1.39	-0.38	-0.37	-0.39
Scattering Transformation	46.37	47.31	43.94	-0.61	-0.73	-0.78
	Slave Robot (Y_s)					
Negative Feedback	9.4	17.45	18.15	-1.02	-1.38	-1.92
Scattering Transformation	30.83	73.97	83.57	-1.54	-2.16	-2.29

Performance Evaluation for Passive Systems

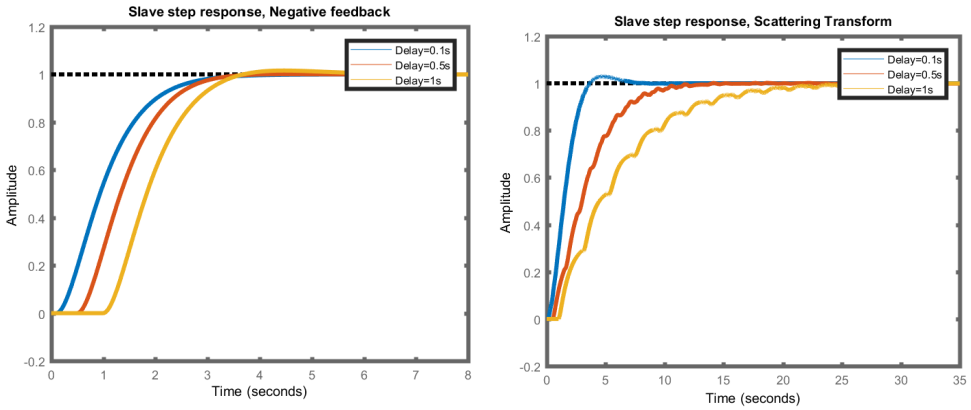
The proposed model is already known to be superior to existing models due to its stability for passivity-short systems. Additionally, the step response characteristics of both the proposed model and the scattering transformation are studied for passive systems to compare their performance characteristics.



(a) Negative Feedback (b) Scattering Transformation

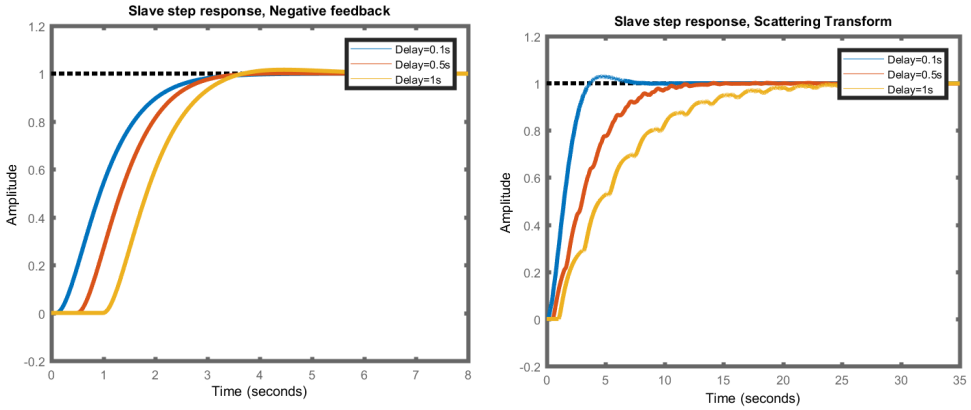
Figure 3.6: Master output vs delay for passive systems

The master and slave systems (P_m and P_s) are considered as linear passivity transfer functions,



(a) Negative Feedback (b) Scattering Transformation

Figure 3.7: Slave output vs delay for passive systems



(a) Master Step Response (b) Slave Step Response

Figure 3.8: Comparison of step responses for passive systems

given by:

$$P_m = P_s = \frac{1}{s+1}$$

Negative Feedback Results: The transient response of $\bar{G}_m$ and $\bar{G}_s$ in (3.7) with passive master and slave robot for different delays is shown in Figure 3.6a and Figure 3.7a, respectively. Overshoot in the master output caused by the additional zeros is compensated by adjusting the gains k_p and k_d.

Scattering Transformation Results: The response of $\bar{H}_m$, given in (3.9), for the passive master and slave robots with different delays is shown in Figure 3.6b. The overshoot in the master response is not compensated and increases as the delay increases. It is noted that the damping for this system cannot be adjusted without affecting the performance of the slave output. The transient response of the slave output ($\bar{H}_s$), given in (3.9), is shown in Figure 3.7b. The slave output exhibits a lesser overshoot, but the settling time is high. As the delay increases, the response becomes oscillatory due to the presence of frequency components. **Comparison:** The master and slave step response of the proposed method and scattering transformation is shown together in Figure 3.8. It can be seen that the proposed method has lesser overshoot and settling time than the scattering transformation. Table 3.1, generated based on the experimentally tuned gain values $k_m = 0.35$, $k_s = 0.6$, $k_p = 0.625$, $k_d = 0.025$, $b_m = 4$, $b_s = 1$ and different delays, compares the master and slave step response characteristics of the negative feedback and the scattering transformation for different delays, numerically. The first half of the table shows the settling time, and the second half shows the $\%$ overshoot, for different values of delays. Table 3.2, generated based on the same gain values, compares the master and slave sinusoidal response characteristics of the negative feedback and the scattering transformation for different delays, for the same system. The first half of the table shows the $\%$ error, and the second half shows the phase lag for different values of delays.

The above evaluations yield the following conclusions:

- The response of $\bar{G}_m$ (proposed model) has less $\%$ overshoot than the response of $\bar{H}_m$ (scattering transformation). This means the proposed model has less error between the sinusoidal master output and the sinusoidal reference input, as seen from Table 3.2.

- The proposed model has a quicker settling time by moving the poles to the desired location. This reduces the phase lag between the sinusoidal master/slave output and the reference input, as seen from Table 3.2.
- Based on the analysis, the suggested model performs effectively for both lower and higher delays. However, upon examining the scattering transformation's master output, it was observed that there was a rising overshoot as the delay increased. Additionally, the slave output of the scattering transformation showed a prolonged settling time and fluctuating response with an increase in delay.

Experimental Results

In this section, the experimental setup is outlined and the comparative results are provided. The setup consists of two Geomagic Touch (previously Phantom Omni) haptic devices as shown in Figure 3.9. They have 3 DOF-actuated joints and a 3 DOF stylus pen. In this experiment, only the first 3 joints are considered and the stylus pen is immobilized. The two devices are connected to the same computer, and master/slave controllers are implemented in MATLAB/Simulink, using the Simulink library PhanSim [81]. Two sets of experiments are conducted: one in the environment where a rigid object in the form of a red box (as shown in Figure 3.9) constrains the slave's first joint, and another in the free space (without any constraint).

The input from the human operator aims to move all the joints to their full ranges of motion. The communication channel is simulated with a variable-time delay, randomly generated under a normal distribution. Unless stated otherwise, the randomly varying time delay is generated with mean $0.6s$ and the maximum rate of change of delay, $\dot{T}_{max} = 0.1$. Since the master and the slave devices have the same structure, their PD parameters set as $k_{pi} = 0.6$, $k_{vi} = 0.2$, for $i = m, s$. The maximum eigenvalue of the inertia matrix used in the experiment is $\lambda_{max}(M_i) = 3.19 * 10^{-4}$ (same

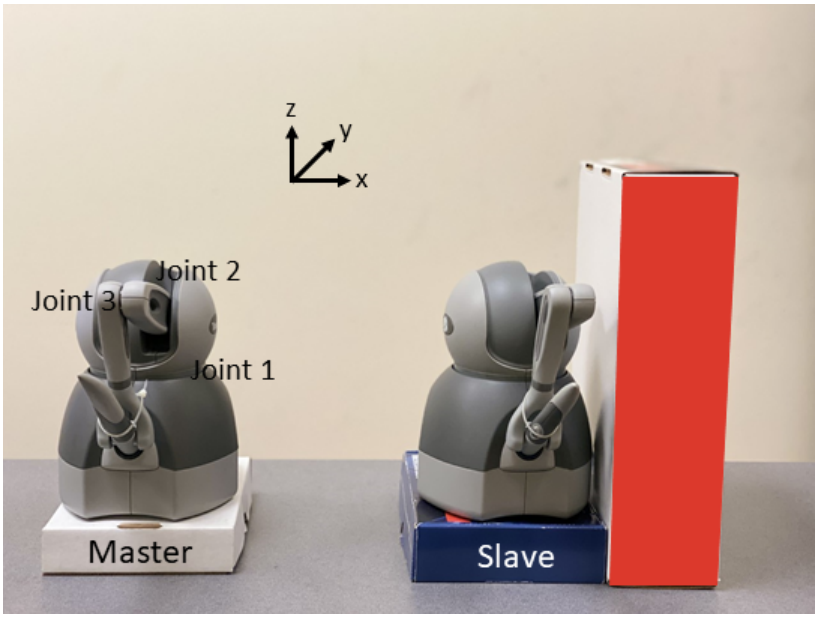


Figure 3.9: Experimental setup

for both master and slave), obtained by a simple system identification of the haptic devices used. Hence, following (3.2), the control gains are chosen as $k_f = 1.6$, $k_m = 0.5$, $k_s = 0.55$, $k_1 = 1$, $k_2 = 1$, $b_m = 1.05$, $b_s = 0.5$.

Free-Space Motion

The results of the free-space experiments are shown in Figure 3.10 in terms of joint position outputs of the master and slave systems, their joint torques, and the instantaneous position errors over time. It can be seen that the position and torque trajectories of the master and slave converge to each other, and hence stability is demonstrated. The position errors are due to the phase lag caused by the delay, which is inevitable; but at the steady state, the position errors become convergent. Their torque response shows that the proposed method has a good force reflection. Their corresponding cartesian trajectories are shown in Figure 3.11. It can be seen that in addition to

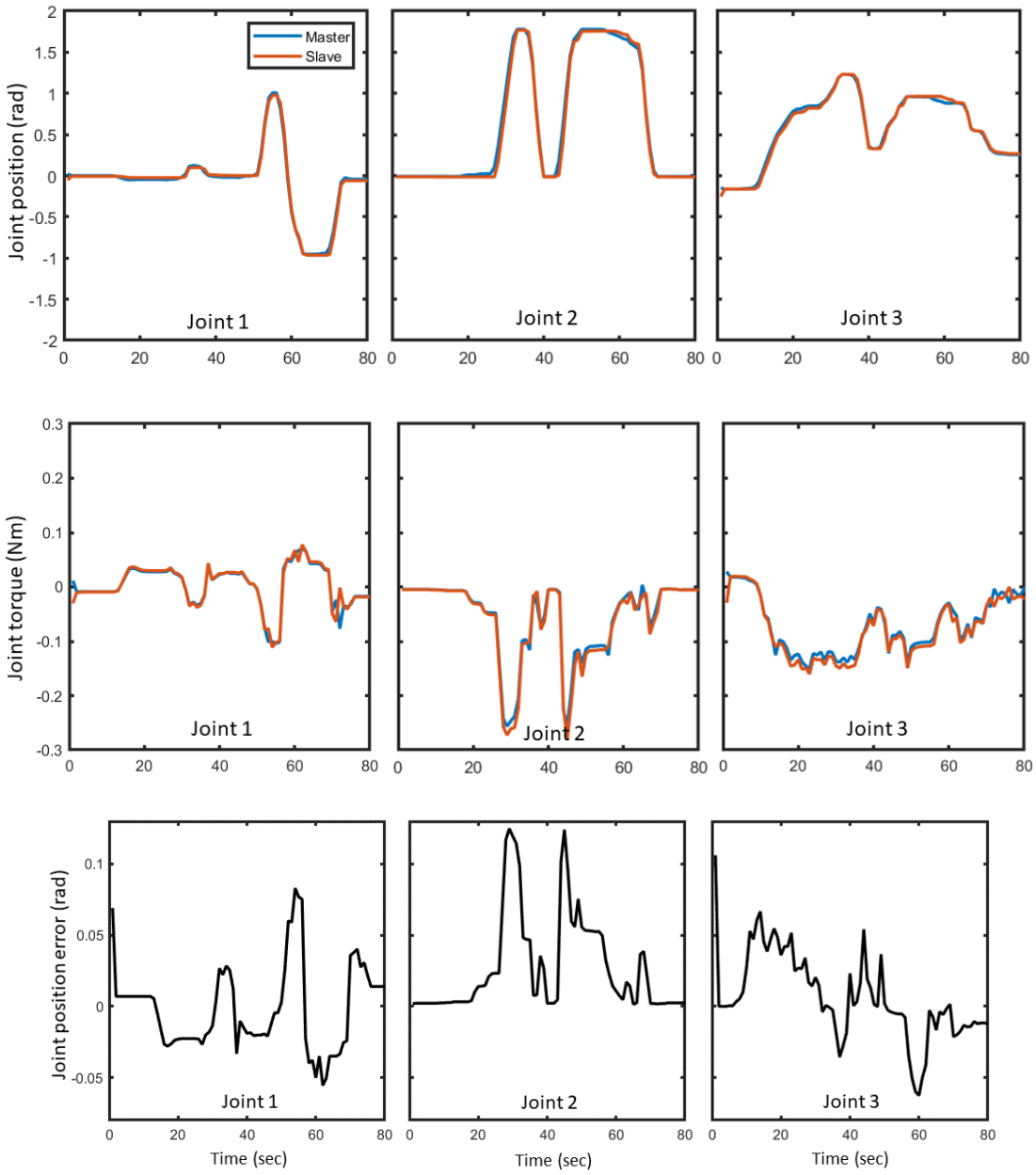


Figure 3.10: Joint position (top), torque (mid) and position error (bottom) of the proposed approach in free space

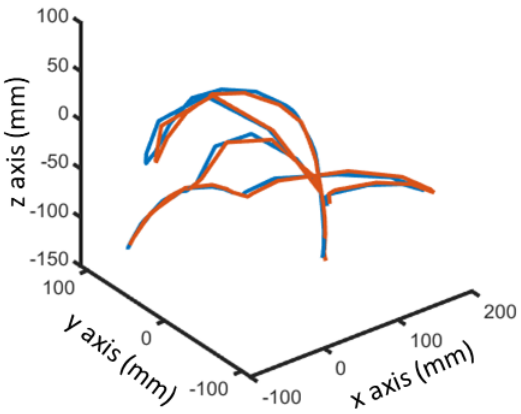


Figure 3.11: Cartesian trajectories under the proposed approach in free space

velocity synchronization (which is used in all passivity-based teleoperation approaches in the literature and [1]), our proposed approach also ensures that position synchronization in the joint space renders position synchronization in the task space.

In addition to stability, there exists a measure of performance known as transparency, generally discussed in the force control literature [82]. In ideally transparent teleoperation, where the master and slave systems are defined by the same dynamics and Denavit-Hartenberg (DH) parameters, with delay-free communication, there exists a kinematic correspondence between the resulting master and slave position and force responses, throughout the teleoperation cycle. During such a correspondence, the impedance perceived by the human operator matches the environment impedance. Transparency of teleoperation can also be measured by the transparency index (μ), which is the ratio of percentage amplitude error (PAE) between the master and slave position responses, and PAE between their torque responses. Such an index of an ideal transparent system would be $\mu_{ideal} = 1$. Given the objective of this research on robust stability and convergence, and with the master and slave systems defined by the same dynamics and DH parameters, the measures of phase lag and PAE are used as performance indices in this paper. The PAE and phase lag between the master

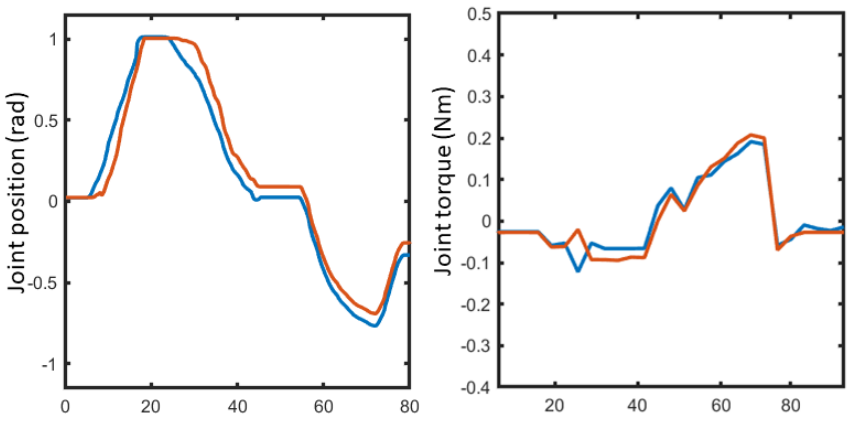


Figure 3.12: Joint position and torque for the proposed approach in free space with large slow-varying delays

and slave position response for the three joints are $[0.65425, -0.0053]$, $[2.55, 0]$, and $[0.99, 0]$, and between their torque responses are $[0.7921, -0.0620]$, $[3.110, 0]$, and $[3.78, 0]$, respectively. The average transparency index of this teleoperation cycle is $\mu_{avg} = 0.6323$ and a phase lag closer to zero indicates that the slave responds very fast to the master system.

The proposed approach is also applied to teleoperation with larger communication delays. Since the performances on different joints are similar, only the results of the first joints of the master and slave are presented here in this paragraph. Performance of the proposed approach in the presence of higher varying-delays (of a mean 5s and maximum rate of change $\dot{T}_{max} = 0.1$) is shown in Figure 3.12, where the position and torque trajectories of the slave track those of the master, with the same amplitudes but with a higher phase lag. The PAE and phase lag of this position and torque responses are $[2.432, -0.1850]$, $[4.0392, -0.1581]$, $\mu_{avg} = 0.6021$. Higher phase lag is expected due to a larger delay, but it is evident that the transparency is not affected due to the large delay.

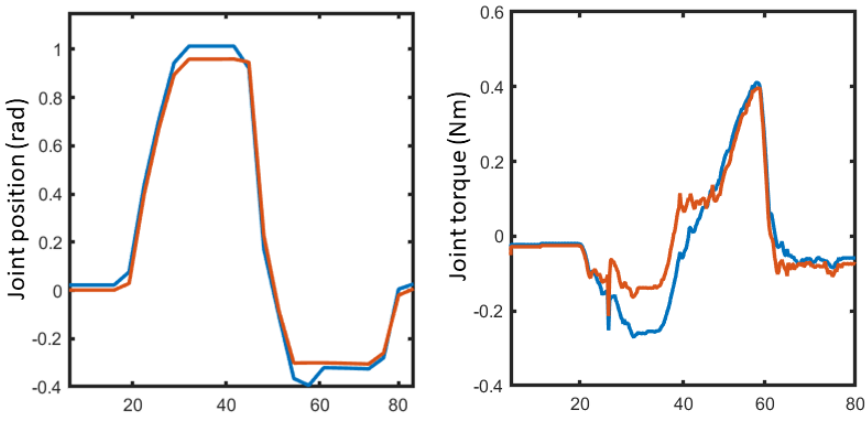


Figure 3.13: Joint position and torque output of the proposed approach in a "rigid" environemnt

Motion in Constrained Environment

The experiment is also conducted in a "rigid" environment where a hard constraint is placed on the first joint (by the red box in Figure 3.9). The position and torque output are shown in Figure 3.13. It can be seen that the environment constrains the movement of the slave system (position trajectory) from 50s to 65s and, during that period, the torque is increased on the slave side. This contact torque is also felt on the master side, even though the operator tries to keep moving along the desired trajectory, and this force feedback increasingly forces the master position back to that corresponding to the slave position.

Comparative Study

To further demonstrate the effectiveness of the proposed approach, a comparative study against [1] is presented. The technique proposed in [1] is an integral quadratic constraint (IQC) framework that uses a Zames-Falb multiplier to model delays and the environment of two-channel teleoperation. The overall system is transformed into a negative feedback interconnection of two blocks: a linear

system block and the uncertainties block. Then, a multiplier is searched such that the uncertainties satisfy the IQC. Even though this approach allows the uncertainties block to accommodate the loss of passivity in master or slave systems, the formulated constraints require strict passivity. As a result, the overall system is subjected to passivity conditions. The result of this approach is shown in comparison to the proposed in Figure3.14, which compares the master and slave joint position errors and torque errors. As used in [1], the gains are chosen as $\mu = 0.8$, and $K_f = 0.6$. The proposed method is far more effective than IQC, in the sense of smaller errors. Besides, it can be seen that the position error during contact with the rigid body (between 50s and 65s) is lesser for the proposed approach, and it is also more responsive to environmental disturbances.

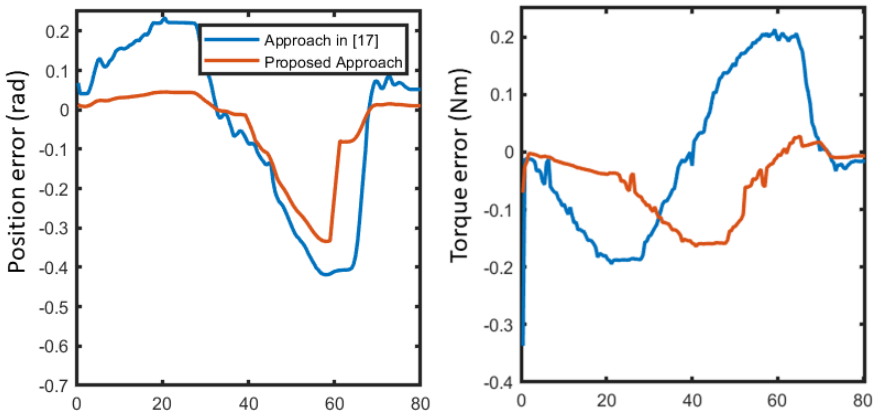


Figure 3.14: Comparision between the proposed approach and [1]

Summary

In this chapter, a comprehensive stability analysis and performance evaluation for teleoperation systems under time-varying delays using the passivity-short framework was conducted. The proposed negative feedback control design was demonstrated to achieve L_2 stability for passivity-short systems, and the necessary conditions for the system parameters to ensure stability and perfor-

mance were derived.

Through numerical simulations, the performance of the proposed model was compared with the existing scattering transformation method. The results showed that the proposed negative feedback approach outperforms the scattering transformation in terms of phase lag and overshoot, demonstrating its superior stability and responsiveness for both passivity-short and passive systems. The experimental results, conducted using Geomagic Touch haptic devices, validated the theoretical claims, showing stable and responsive teleoperation under free-space motion and constrained environments. Comparative studies with existing methods highlighted the superior performance of the proposed approach in terms of lower position and torque errors, even in the presence of time-varying delays.

Overall, the chapter provides a solid foundation for understanding and implementing effective control strategies for teleoperation systems, ensuring stability and high performance in practical applications.

CHAPTER 4: RESILIENT MULTI-AGENT SYSTEMS AGAINST DENIAL OF SERVICE ATTACKS

Introduction

As we transition from the exploration of teleoperation systems and their stability under varying conditions, we now delve into the critical issue of network resilience in multi-agent systems, particularly focusing on their vulnerability to DoS attacks. In both directed and undirected networks, a critical edge—also known as a bridge in undirected networks or a cut arc in directed networks—is an edge whose removal leads to the network becoming disconnected. This identifies it as a significant point of vulnerability within the network's structure. In undirected networks, the critical edge is crucial for linking two components; without it, these components become isolated. In directed networks, a critical edge is essential for maintaining strong connectivity, which is necessary for any node to communicate with any other node. Removing the critical edge disrupts this essential connectivity, preventing the directional flow of information from reaching all parts of the network. This loss of reachability highlights the edge as a critical link whose integrity is vital for sustaining the overall connectivity and operational capacity of the network. [83].

Edge connectivity, on the other hand, is a fundamental metric of network robustness, quantifying the minimum number of edges that must be removed to render the network disconnected. The presence of critical edges directly impacts a network's edge connectivity; specifically, a network with even a single critical edge has an edge connectivity of 1. A network with high edge connectivity is more robust, and capable of withstanding multiple edge failures without losing overall connectivity, whereas networks with critical edges are vulnerable, as the failure of just one such edge can compromise network integrity [84]. Therefore, identifying and reinforcing critical edges not only

enhances a network's robustness but also ensures its resilience to disruptions such as failures or targeted attacks.

This chapter presents a comprehensive framework for enhancing the connectivity and robustness of multi-agent systems through distributed algorithms. Ensuring reliable communication and operation within these systems is paramount, especially in the face of potential failures and attacks. The proposed methods focus on two key aspects: maintaining network connectivity and identifying and eliminating critical edges.

To achieve these objectives, four concurrent algorithms are introduced to operate within the network. Initially, the standard maximum/minimum consensus algorithm is adeptly utilized not only to ascertain network connectivity but also to comprehensively map out each node's neighbors and their minimum distances throughout the network. Subsequently, if the network is originally disconnected, the second algorithm methodically adds edges to enhance connectivity. The third algorithm is a novel distributed method designed to determine whether a specific edge is part of a cycle, aiding in the identification of all critical edges within the network. The final step in this approach involves strategically adding edges to eliminate all critical edges, thereby enhancing network robustness.

The final section extends these algorithms to directed graphs, addressing the unique challenges posed by directed networks and illustrating the scalability of our approach through network reduction techniques. This comprehensive treatment aims to provide a robust framework for improving the resilience and reliability of multi-agent systems, ensuring their continued operation even in adverse conditions. This work is published partially in [85].

Distributed Algorithms for Critical Edge Elimination in Multi-Agent Systems

This approach requires the following 4 algorithms to be performed concurrently.

1. An n-step algorithm to determine the neighbor structure, including the distance from one node to every other node in the network. This has been previously discussed in section 2.

2. An n-step algorithm to ensure the connectivity of the network by adding edges in an ordered fashion.

3. A 2-step algorithm to determine the existence of critical edges/nodes by identifying the absence of cycle(s) containing the specific edge or alternate paths between any pair of two neighboring nodes.

4. If critical edge(s) is(are) identified, an n-step distributed algorithm to add a minimum number of new edges to eliminate critical edges and increase edge connectivity to 2.

Distributed Algorithm to Ensure Connectivity

Consider a modified version of update law (2.27)) for steps $k = n, \cdots, (2n-1)$:

$$\omega_{i,j}(k+1) = \begin{cases} \omega_{i,j}(k) & \text{if } \xi_{i,j}(k+1) \\ & = \xi_{i,j}(k), \\ \omega_{i,i*}(k) + \omega_{j^*,j}(k) & \text{if } \xi_{i,j}(k+1) \\ & > \xi_{i,j}(k). \end{cases} \tag{4.1}$$

where $\xi_{i,j}(k)$ is obtained using update law (2.26).

If the original network is not connected, every node in the network becomes aware of this fact at $k = n$. Specifically, if node i does not have node j in its neighboring structure (in the sense that $\xi_{i,j}(n) = 0$), node i knows that the network is not connected. Then, the node i^* with the highest

index within the connected neighbors is identified, that is

$$i^* = \max\{j \mid \xi_{i,j}(n) = 1\}.$$

Then, i^* identifies its pair j^*, that is not within its connected neighbors as,

$$j^* = \max\{j \mid \xi_{i,j}(n) = 0\}.$$

A new edge from i^* to j^* is added, which upgrades the network towards being connected. The pseudo-code for this is given in Algorithm A for easy reference.

In the worst case that there is no edge in the original network, Algorithm A can take $n-1$ steps to ensure connectivity. For this reason, even when starting with a connected network, (4.1) and (2.26) need to run until $k = 2n$ so that there is no need for central coordination.

To summarize, **Algorithm 2** comprises of update laws (4.1) and (2.26) running for n steps, and for the first $n_c - 1$ steps, where n_c is the number of disjoint connected components, the network is upgraded by Algorithm A. The upgraded network at $k = n + n_c - 1$ becomes connected. By $k = 2n$, every node sees this conclusion in its neighboring structure.

Algorithm 1 Distributed Connection of Non-Connected Network

Require: $\xi_{i,j}(k)$ at $k = n$
1: **while** $\xi_{i,j}(k) = 0$ for any j **do**
2: identify node $i^* = \max(j)$ with the highest index where $\xi_{i,j}(k) = 1$.
3: **if** i is i^* **then**
4: find node $j^* = \max(j)$ with the highest index where $\xi_{i,j}(k) = 0$.
5: establish a link with node j^*.
6: **end if**
7: $k = k + 1$
8: **end while**
Ensure: The network connectivity is established.

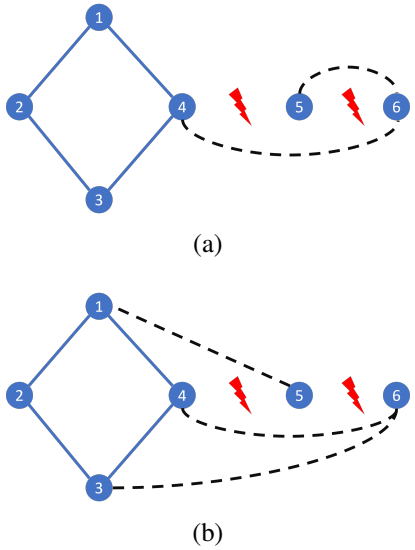


Figure 4.1: Network repair and reconfiguration: a) Highest Index Identification; b) Random Index Identification for J^*. The newly added edges are in dashed lines and the red markers show disconnection from attacks

Effects on Centrality Measures

Centrality measures are metrics used in network theory to identify the most important or influential nodes within a network. There are several types of centrality, including:

- **Degree Centrality**: This measure counts the number of direct connections a node has. Nodes with higher degree centrality are often considered more influential because they can directly connect with many other nodes.
- **Betweenness Centrality**: This measure indicates the number of times a node acts as a bridge along the shortest path between two other nodes. Nodes with high betweenness centrality can control information flow within the network.
- **Closeness Centrality**: This measure assesses how close a node is to all other nodes in the network. Nodes with high closeness centrality can spread information more efficiently

through the network.

The network repair mechanisms utilized following disruptions are illustrated in Figure 4.1, showcasing multiple strategies for identifying the node j^* for new connections. Two distinct approaches, Highest Index Identification and Random Index Identification, are depicted with dashed lines to emphasize the adaptive modifications restoring connectivity, while red markers indicate the locations of the attacks.

The Highest Index Identification strategy in Figure 4.1(a) selects the node with the highest index that lacks connectivity, providing a systematic and hierarchical approach to network repair. This method tends to enhance the degree centrality of specific high-index nodes.

Conversely, the Random Index Identification strategy shown in Figure 4.1(b) selects a random node that lacks connectivity, introducing an element of unpredictability and variability in the repair process. This approach results in a more distributed increase in degree centrality across the network, as the repaired connections are spread more evenly among the nodes.

Different strategies impact the centrality measures differently. The Highest Index Identification approach may lead to higher centralization, with certain nodes becoming more central than others. In contrast, the Random Index Identification approach promotes a more balanced network, potentially reducing the overall centralization and making the network more robust to subsequent attacks by avoiding over-reliance on a few key nodes.

Distributed Determination of Critical Edges

In this section, a distributed 2-step algorithm (a total of $(2n + 2)$ steps when combined with the previous algorithms) is presented for the i^{th} node to determine whether its associated edge e_{il} to its

first neighbor node l is a critical edge. To facilitate this, a measure, $\Delta_i^{(il)}$, is employed, calculated at the $n+1^{\text{th}}$ iteration, to evaluate the presence of alternative pathways between any two directly connected nodes, i, and a neighbor $l \in \mathcal{N}_i^{(1)}$. This measure is defined as follows:

$$\Delta_i^{(il)} = [\Delta_{i,1}^{(il)} \ \cdots \ \Delta_{i,n}^{(il)}]^T, \quad \Delta_{i,j}^{(il)} = \omega_{i,j}(n) - \omega_{l,j}(n), \tag{4.2}$$

where $\omega_{i,j}(n)$ and $\omega_{l,j}(n)$ represent the computed shortest distances from node j to nodes i and l, respectively, obtained after n iterations of (2.27).

In an undirected and connected network, the value of $\Delta_{i,j}^{(il)}$ for each pair of neighboring nodes i and l, with respect to any other node j, can only be -1, 0, or 1. A value of -1 indicates that node j is relatively closer to node i than to node l, and conversely, a value of 1 suggests node j is nearer to node l. A zero value implies that node j is equidistant from both nodes i and l. Through the analysis of these distance increments, a lemma and a theorem are proposed, providing criteria for distributively identifying alternate paths and critical edges within the network, thereby enhancing the understanding of the network's robustness to potential disruptions.

Lemma 6. *Given an undirected and connected network $\mathcal{G}$, consider an edge e_{il} connecting nodes i and l. A node, denoted as k, possesses alternate paths to both i and l, bypassing e_{il}, if and only if at least one node j (potentially including k itself and residing within the relevant cycle) satisfies one of the two conditions below:*

1. *Node j is equidistant to i and l*

$$\Delta_{i,j}^{(il)} = 0, \tag{4.3}$$

2. *There exist nodes $i' \in \mathcal{N}_i$ and $l' \in \mathcal{N}_l$ such that*

$$\Delta_{i,j}^{(il)} \neq 0, \quad \Longrightarrow \quad \Delta_{i,j}^{(ii')} = \Delta_{l,j}^{(ll')} = 1. \tag{4.4}$$

indicating j is on a path that connects i and l through nodes i' and l', thus forming a cycle that includes e_{il}.

Proof: The proof is organized into two main parts: sufficiency and necessity.

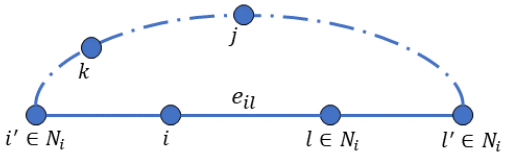
Sufficiency: If there exists a node k with alternate paths to i and l that do not pass through e_{il}, then at least one cycle involving e_{il} and other edges is present in the network. This cycle can be detected by examining distances from nodes in the network to i and l:

1. If $\Delta_{i,j}^{(il)} = 0$ for some node j, it indicates that j is equidistant from both i and l. This condition signifies the presence of a cycle that j is a part of, where j is on an alternate path that circumvents e_{il}, showing the cycle's existence without directly counting nodes.

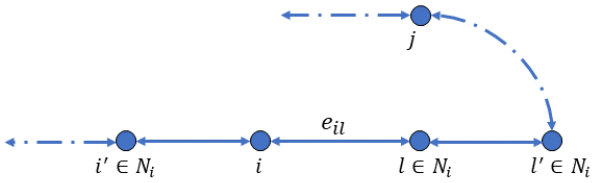
2. Let's consider nodes i' and l', where $i' \in \mathcal{N}_i$ and $l' \in \mathcal{N}_l$ belong to the cycle. There exists a node j that has an equal distance to the cluster consisting of nodes i and l. This implies that departing from either node i or l toward node j results in the distance of the neighbor structure decreasing in both directions, as illustrated in Figure 4.2(a). Consequently, both $\Delta_{i,j}^{(ii')}$ and $\Delta_{l,j}^{(ll')}$ are equal to 1, irrespective of the sign of $\Delta_{i,j}^{(il)}$. This condition confirms that j lies on a cycle that includes both i and l, as well as their neighbors, indicating the existence of alternate paths.

Necessity: Assume no alternate paths exist between i and l except through e_{il}. Removing e_{il} disconnects i and l, showing e_{il} is a critical edge. In such a scenario, for any node j not equal to i or l, it's impossible to have $\Delta_{i,j}^{(il)} = 0$ because j cannot be equidistant to i and l without e_{il}. Additionally, you cannot find nodes $i' \in \mathcal{N}_i$ and $l' \in \mathcal{N}_l$ satisfying $\Delta_{i,j}^{(ii')} = \Delta_{l,j}^{(ll')} = 1$ for any j, as there are no cycles including e_{il} and other edges that could provide such alternate paths. Furthermore, we will have $\Delta_{i,j}^{(ii')} * \Delta_{l,j}^{(ll')} = -1$ (i.e., they must have different signs), since leaving

the cluster of nodes i and l has opposite effects: closer to node j in one direction, and farther in the other, as shown in Figure 4.2(b). □



(a) Edge e_{il} is a part of the cycle formed by alternate paths



(b) Edge e_{il} is not a part of any cycle

Figure 4.2: Illustration of cycle

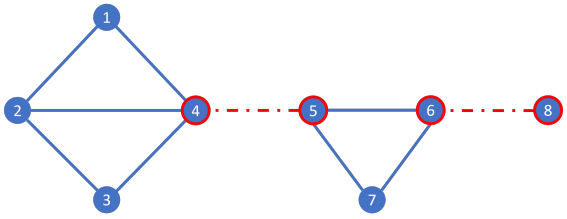
Theorem 5. *Given an undirected and connected network $\mathcal{G}$, suppose each node implements a single-step computation* (4.2)*, in conjunction with the n-step protocols* (2.26) *and* (2.27)*. An edge e_{il} is a critical edge within this framework if and only if, for every node $j \in \mathcal{N}$ and for all adjacent nodes $i' \in \mathcal{N}_i$ and $l' \in \mathcal{N}_l$, the following condition is met:*

$$\Delta_{i,j}^{(il)} \neq 0, \quad \textit{and} \quad \{\Delta_{i,j}^{(ii')},\ \Delta_{l,j}^{(ll')}\} \neq \{1,\ 1\}. \tag{4.5}$$

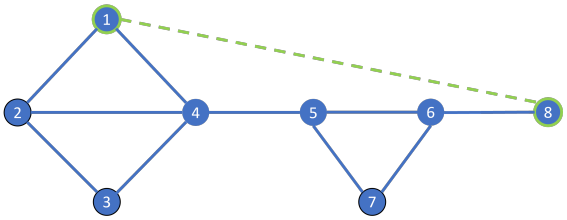
Proof: Condition (4.5) explicitly negates the scenarios described in (4.3) and (4.4) from Lemma 1, indicating e_{il} lacks alternative paths, thus establishing its criticality. Sufficiency arises directly from Lemma 1: if no conditions for non-criticality are met, e_{il} is critical as it is the sole link between i and l. Necessity is self-evident; without e_{il}, connectivity between i and l breaks, signifying its critical. □

This proof conclusively establishes the criteria for determining the critical nature of an edge within the network by linking the absence of alternate paths to the specified unique conditions. □

Algorithm 3 runs for 2 steps where each node can identify the set of all its associated critical edges from (4.5), denoted by $\mathcal{N}_i^c \subset \mathcal{N}_i$ for the i^{th} node in connected graphs.



(a) Undirected network $\mathcal{G}_{ex1}$ with *critical edges* (red dotted lines)



(b) Network $\mathcal{G}_{ex2}$ without critical edge *by adding an edge* (green dot-dashed line)

Figure 4.3: Iluustration of critical edges

Example 3. *For $\mathcal{G}_{ex1}$ in Figure 4.3(ab), it follows that, at $k = 9$, $\Delta_{i,j}^{(il)}$ is as follows*

For edge e_{4-5}:

$$\begin{aligned}\Delta_4^{(45)} = [&\omega_{4,1}(n) - \omega_{5,1}(n), \omega_{4,2}(n) - \omega_{5,2}(n), \omega_{4,3}(n) - \omega_{5,3}(n),\\ &\omega_{4,4}(n) - \omega_{5,4}(n), \omega_{4,5}(n) - \omega_{5,5}(n), \omega_{4,6}(n) - \omega_{5,6}(n),\\ &\omega_{4,7}(n) - \omega_{5,7}(n), \omega_{4,8}(n) - \omega_{5,8}(n)]^T\\ \Delta_4^{(45)} = [&-1, -1, -1, -1, 1, 1, 1, 1]^T,\end{aligned}$$

For edge e_{4-1}:

$$\begin{aligned}\Delta_4^{(41)} = [&\omega_{4,1}(n) - \omega_{1,1}(n), \omega_{4,2}(n) - \omega_{1,2}(n), \omega_{4,3}(n) - \omega_{1,3}(n),\\ &\omega_{4,4}(n) - \omega_{1,4}(n), \omega_{4,5}(n) - \omega_{1,5}(n), \omega_{4,6}(n) - \omega_{1,6}(n),\\ &\omega_{4,7}(n) - \omega_{1,7}(n), \omega_{4,8}(n) - \omega_{1,8}(n)]^T\\ \Delta_4^{(41)} = &[1, 0, -1, -1, -1, -1, -1, -1]^T,\end{aligned}$$

For edge e_{5-6}:

$$\begin{aligned}\Delta_5^{(56)} = [&\omega_{5,1}(n) - \omega_{6,1}(n), \omega_{5,2}(n) - \omega_{6,2}(n), \omega_{5,3}(n) - \omega_{6,3}(n),\\ &\omega_{5,4}(n) - \omega_{6,4}(n), \omega_{5,5}(n) - \omega_{6,5}(n), \omega_{5,6}(n) - \omega_{6,6}(n),\\ &\omega_{5,7}(n) - \omega_{6,7}(n), \omega_{5,8}(n) - \omega_{6,8}(n)]^T\\ \Delta_5^{(56)} = &[-1, -1, -1, -1, -1, 1, 1, 1]^T.\end{aligned}$$

It is straightforward that (4.5) *is satisfied for all i' and l' (e.g., $\Delta_5^{(57)}$) not explicitly shown, thus indicating that edge e_{45} is critical.*

Distributed Edge Addition

This section introduces an algorithm to enhance network robustness by strategically adding new edges to eliminate critical ones, thereby reducing vulnerability to link failures. Our approach focuses on connecting distant nodes within acyclic parts of the network—specifically, those separated by critical edges—to both preserve network connectivity under attack and minimize the additional connections required. By identifying the most remote pairs of nodes adjacent to each critical edge, called *augmentation nodes*, the algorithm ensures these nodes establish new links. Through local

propagation, the augmentation nodes receive the information and complete the planned edge addition, which has a maximum of $n+1$ steps (and, when combined with the previous components, have a total of $3n+3$ steps).

The first step of the n-step edge addition algorithm, that is **Algorithm 4** is executed only by individually each pair of nodes associated with a critical edge, say $\{i,l\}$ with $i \in \mathcal{N}_l^c$ and $l \in \mathcal{N}_i^c$. The goal of this step is to identify the corresponding augmentation nodes, $\{i',l'\} \in \mathcal{N}_i^r$. These augmentation nodes are strategically chosen as the ones farthest (with respect to the neighbor structure) from the critical edge e_{il}. If there exist multiple nodes at the same distance, the ones with the smallest index are chosen, and it reduces the number of edges added.

$$\{i',l'\} \in \mathcal{N}_i^r \text{ if (4.7) or (4.8) or (4.9) is true,} \tag{4.6}$$

where

$$\begin{cases} \mu_i = |\mathcal{N}_i| > 1 \\ \mu_l = |\mathcal{N}_l| > 1 \end{cases} \text{ and } \begin{cases} i \in \mathcal{N}_l^c,\ l \in \mathcal{N}_i^c, \\ \Delta_{i,i'}^{(il)} = \Delta_{l,l'}^{(li)} = -1 \\ \omega_{i,i'} = \max_k \omega_{i,k}, \\ \omega_{l,l'} = \max_k \omega_{l,k}, \end{cases}, \tag{4.7}$$

$$\begin{cases} \mu_i = |\mathcal{N}_i| = 1 \\ \mu_l = |\mathcal{N}_l| > 1 \end{cases} \text{ and } \begin{cases} i \in \mathcal{N}_l^c,\ l \in \mathcal{N}_i^c, \\ \Delta_{l,l'}^{(li)} = -1 \\ i' = i, \\ \omega_{l,l'} = \max_k \omega_{l,k}, \end{cases}, \tag{4.8}$$

$$\begin{cases} \mu_i = |\mathcal{N}_i| > 1 \\ \mu_l = |\mathcal{N}_l| = 1 \end{cases} \text{ and } \begin{cases} i \in \mathcal{N}_l^c,\ l \in \mathcal{N}_i^c, \\ \Delta_{i,i'}^{(il)} = -1 \\ \omega_{i,i'} = \max_k \omega_{i,k}, \\ l' = l, \end{cases}, \tag{4.9}$$

and $\mathcal{N}^r = \cup_{i\in\mathcal{N}}\mathcal{N}_i^r$ is the *augmentation action set* to be found distributively. Note that $\mu_i = |\mathcal{N}_i| = \mu_l = |\mathcal{N}_l| = 1$ is the trivial case of a two-node network and hence is excluded from consideration.

The next $(n-1)$ steps of the Algorithm 4 is to propagate $\mathcal{N}_i^r$ to all the nodes so they all have access to $\mathcal{N}^r$ as follows:

$$\mathcal{N}_i^r(k+1) = \cup_{l\in\mathcal{N}_i\cup\{i\}}\mathcal{N}_l^r(k), \tag{4.10}$$

where $k = (2n+2), \cdots, (3n+1)$, $\cup$ is the union operation of sets containing non-ordering pairs (that is, if $\{i,j\} \subset \mathcal{N}_l^r(k)$, then $\{j,i\} \subset \mathcal{N}_l^r(k)$), and $\mathcal{N}_i^r(3n+2) = \mathcal{N}_i^r$ is given by (4.6).

And, as the final step, edge addition is accomplished by the pairs of nodes identified in $\mathcal{N}^r$ to complete their connection.

Theorem 6. *Consider connected network $\mathcal{G}$ in which each node executes distributed n-step algorithm represented by equations* (4.6) *and* (4.10)*, following the distributed $2n+2$-step algorithms, 1,2 and 3. Then, the resulting network has no critical edge or nodes.*

Proof: The algorithm of (4.6) and (4.10) ensures that there is no critical edge. Hence, the resulting network has no vulnerability under one link failure anywhere in the network. □

Example 4. *Consider the graph in Figure 4.3(a).*

1. ***Identify Critical Edges***

   First, the critical edges are identified using the previously defined algorithm. For this graph, the critical edges are e_{4-5} and e_{6-8}.

2. ***Determine Augmentation Nodes***

   Next, the augmentation nodes for each critical edge are determined. The augmentation nodes are chosen as the ones farthest from the critical edge.

- *For the critical edge e_{4-5}:*

$$\Delta_4^{(45)} = [-1, -1, -1, -1, 1, 1, 1, 1]^T$$

This indicates that nodes 1, 2, and 3 are closer to node 4, while nodes 6, 7, and 8 are closer to node 5. The farthest nodes from the critical edge are nodes 1 and 8. Thus, $i' = 1$ and $l' = 8$.

- *For the critical edge e_{6-8}:*

$$\Delta_6^{(68)} = [-1, -1, -1, -1, -1, -1, -1, 1]^T$$

This indicates that all other nodes 1-7 are closer to node 6. The farthest nodes from the critical edge are nodes 1 and 8. Thus, $i' = 1$ and $l' = 8$.

3. ***Propagate Augmentation Nodes Information***

The augmentation nodes $\mathcal{N}^r$ are propagated to all nodes in the network.

- *For node 4:*

$$\mathcal{N}_4^r = \{(1, 8)\}$$

- *For node 6 (also for 5 and 8):*

$$\mathcal{N}_6^r = \{(1, 8)\}$$

4. ***Add New Edges***

Finally, new edges are added between the augmentation nodes to enhance the network's robustness.

- *Add edge* $(1, 8)$

Resulting Network

By strategically adding the edge $(1, 8)$ *as shown in Figure. 4.3(b) enhances the network's robustness, reducing vulnerability to link failures and ensuring better connectivity.*

To facilitate understanding, $\mathcal{G}_{ex2}$ *is represented as a simplified graph in Figure 4.4 with a group of nodes. As illustrated in the graph, the objective of the proposed approach is to ensure that every node is part of a cycle. The presence of a cycle indicates the absence of a critical edge, ensuring a minimum edge connectivity of 2.*

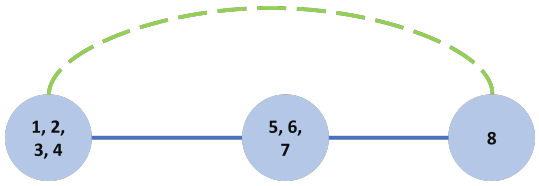


Figure 4.4: Simplified representation of graph $\mathcal{G}_{ex2}$ where each node is a collection of nodes that are originally connected

The pseudo-code of the proposed approach is included below for easy reference in Algorithm B. The algorithm can be implemented as fast as needed to improve network robustness and achieve resilience against link failures.

Complexity and Robustness

All four algorithms are of finite steps proportional to n. For example, Algorithm B operates over $3n + 3$ steps. Hence, the time complexity of the proposed method is $\mathcal{O}(n)$. In addition, each node processes information distributively, their computations involve linear update laws in terms of n-dimensional vectors corresponding to the previous and current time step, and hence memory complexity of the proposed approach is also $\mathcal{O}(n)$.

Algorithm 2 Distributed Edge Addition

Require: Node with unique index i: network size n and neighbor set $\mathcal{N}_i$.

1: Run Algorithm 1 and 2 to ensure connectivity ▷ *For node i, $\omega_i(2n)$ provides its neighbor structure.*
2: For each of its neighbors (i.e., $l \in \mathcal{N}_i$), calculate $\Delta_{i,j}^{(il)}$ using (4.2) and locally identify critical edges using (4.5). ▷ *At the $(2n+2)$nd step local determination of critical edges/nodes.*
3: If i is a critical node, use (4.6) to determine the augmentation nodes for each pair of its critical node neighbors. ▷ *At the $(2n+3)$th step, all the augmentation nodes are locally identified as $\mathcal{N}_i^r$.*
4: Use max-consensus protocol (4.10) to distributively determine set $\mathcal{N}^r$. ▷ *At the $(3n+2)$nd step, every pair of augmentation nodes knows the need to add an edge between each other.*
5: If i is in $\mathcal{N}^r$, reach out to its pair l to make an edge. ▷ *At the $(3n+3)$rd step, edge addition is complete.*

Ensure: Loop the above steps and the resulting network is connected and has no critical node/edge to improve robustness.

The system's robustness can be measured in terms of algebraic connectivity, which is the second lowest eigenvalue (λ_2) of the graph Laplacian matrix. This measure offers insights into the network's connectivity, a higher λ_2 indicates better connectivity, and consequently a more robust network Additionally, edge connectivity provides another critical measure of the system's robustness. This measure denoted as $\kappa'(\mathcal{G})$ for a network $\mathcal{G}$, represents the minimum number of edges that must be removed to disconnect the network.

Example 5. *For $\mathcal{G}_{ex1}$ in Figure 4.3 (a) with critical edges, the $\lambda_2 = 0.5083$ and for $\mathcal{G}_{ex2}$ in Figure 4.3 (b) without critical edges $\lambda_2 = 0.7933$. Similarly $\kappa'(\mathcal{G}_1)$ is 1, while $\kappa'(\mathcal{G}_2)$ is 2.*

Application to Directed Graphs

This section explains the methods used to identify and strengthen critical edges within a directed network to improve its connectivity and robustness. Critical edges are defined by the connectivity of their destination nodes. For example, an edge $e_{j,i}$ becomes critical if a node k relies solely on j to reach i. This identification process requires each agent to trace all potential paths through which

it is connected to other nodes, helping pinpoint any vulnerabilities in the network's structure.

This section details an algorithm that allows each agent to enumerate all distinct paths leading to it from other nodes, aiding in the detection of critical edges. Additionally, the algorithm suggests ways to add new edges to eliminate any critical dependencies, enhancing the network's stability. However, as the size of the network increases, so does the complexity of this task.

Critical Edge Detection and Reinforcement

This subsection has 3 parts:

1. Introduces a distributed algorithm that systematically records all possible paths between nodes, and based on the number of paths node i determines if the edge $e_{k,i}$ from node k is critical.
2. If deemed critical, the strategy to fortify network resilience involves adding new connections, guided by systematic steps

Network Path Detection Algorithm

This subsection introduces a distributed algorithm designed to identify all possible paths between agents in a multi-agent system. The algorithm systematically records the paths by which information or signals travel from one node to another within the network. This is crucial for analyzing network robustness and designing control strategies that can accommodate dynamic network changes.

Algorithm Overview: The algorithm utilizes a data structure $\bar{x}_i$ for each agent i, which contains n sets corresponding to each node in the network. Each set, $\bar{x}_{i,j}(k)$, represents the collection of

paths from node j to node i that can be traversed in k steps. The paths are sequences of nodes that detail the exact route taken from j to i. The cardinality, $\bar{n}_{i,j}(n) = |\bar{x}_{i,j}(n)|$, quantifies the number of distinct paths from j to i after n steps.

The primary goal of this algorithm is to enumerate the paths from any node j to node i:

- **Single Path:** If $\bar{n}_{i,j} = 1$, there exists exactly one path from j to i through a specific in-neighbor, signifying a unique communication route.
- **Multiple Paths:** If $\bar{n}_{i,j} > 1$, it indicates that node i can receive information from node j through multiple in-neighbors, enhancing the redundancy and resilience of the network.

This iterative mapping of network connectivity helps in understanding the network's communication dynamics and potential points of failure or congestion.

Algorithm Description: The algorithm progresses through a series of iterations for each agent i, from $k = 0$ to $n - 1$:

$$\bar{x}_{i,j}(k+1) = \begin{cases} \bar{x}_{i,j}(k) \cup \{p \cup [i]\}, & \text{if } \exists p \in \bar{x}_{l,j}(k) \text{ and } l \in \mathcal{N}_i \text{ such that } i \notin p \\ \bar{x}_{i,j}(k), & \text{otherwise} \end{cases} \tag{4.11}$$

with initial conditions defined as:

$$\bar{x}_{i,j}(0) = \begin{cases} \{[i]\}, & \text{if } j = i \\ \{\}, & \text{otherwise} \end{cases} \tag{4.12}$$

where $\mathcal{N}_i$ denotes the set of in-neighbors of node i, and p is a path from j to an in-neighbor l identified in the previous iteration.

A **path** p from node j to node i in a graph is an ordered sequence of nodes represented as:

$$p = (v_0, v_1, v_2, \ldots, v_k)$$

where:

- $v_0 = j$ (the starting node)
- $v_k = i$ (the destination node)
- $v_m \to v_{m+1}$ is an edge in the graph G for all $0 \leq m < k$

The update rule for the path detection algorithm specifies that for each in-neighbor l of a node i, if there exists a path p from node j to l that does not already pass through i, then this path p is extended to include i by appending l to the set of known paths $\bar{x}_{i,j}(k+1)$. This operation effectively broadens the scope of known connectivity paths by incorporating new information about direct connections as they are discovered. Conversely, if no such paths satisfy the condition—meaning all known paths from j to l pass through i or no paths exist at all—then the set of paths $\bar{x}_{i,j}(k+1)$ remains unchanged from $\bar{x}_{i,j}(k)$. This ensures that each iteration of the algorithm contributes to a progressively more detailed and accurate mapping of the network's pathways, maintaining the integrity and directionality of the discovered paths.

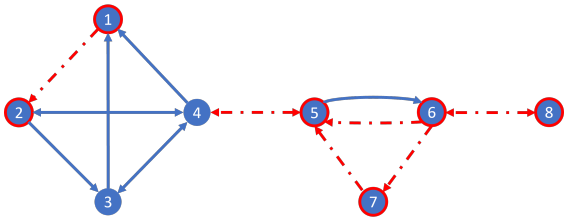
Lemma 7. *Consider a strongly connected digraph $\mathcal{G}$. If every node i runs the update law* (4.11) *for n iterations, each node i can identify all the paths that any node j in the network takes to reach i. Furthermore, it is also guaranteed that no cyclic paths are included in $\bar{x}_{i,k}(n)$, ensuring that all paths identified by i are direct and acyclic.*

Proof: Consider a strongly connected digraph $\mathcal{G}$. If each node i in the network executes the update rule defined in (4.11) for n iterations, the algorithm is designed to map all feasible paths from any node j to i without forming cycles. This ensures that the set $\bar{x}_{i,j}(n)$ only contains acyclic and direct paths. The absence of cyclic paths is enforced by the algorithm's condition that only extends paths to include node i from its in-neighbors l if such extension does not already pass through i. As such, the paths accumulated in $\bar{x}_{i,j}(n)$ provide a comprehensive yet cycle-free depiction of connectivity from j to i, thereby affirming that all paths recognized are direct and acyclic, as stipulated in Lemma 5. ■

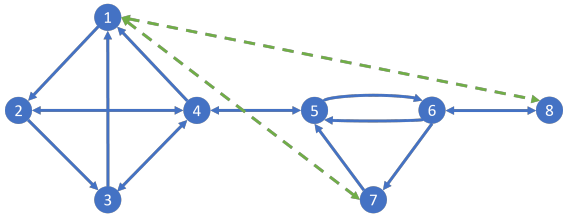
The complexity analysis and properties of the network path detection algorithm, captured in Lemma 5, are crucial for understanding the computational burden and the efficiency of the algorithm. Initially, the complexity of setting up the system is $O(1)$, given that the initialization of path sets for each agent is straightforward. During each iteration $k = 1, \cdots, n$, the complexity depends on multiple factors: the average degree d, representing the number of in-neighbors; the average number of paths per node pair p; and the average path length l. These elements determine the operational complexity at each step, calculated as $O(d \cdot p \cdot l)$. Considering n iterative steps are performed, and assuming an update occurs at each step, the total complexity for a single agent across all steps equates to $O(n)$, leading to an overall complexity of $O(d \cdot p \cdot l \cdot n)$ for the entire process.

Remark: In a fully connected graph scenario—considered as the worst-case scenario for complexity—the average degree d reaches $n-1$, as each node connects to every other node. The algorithm strictly avoids duplicate edges or revisits, hence the number of distinct paths p equals the sum of direct paths and all possible node permutations, i.e., $p = 1+(n-1)!$, and the maximum path length l is $n-1$. This worst-case analysis highlights the potential exponential growth in complexity due to the factorial increase in path permutations as network size increases.

Example 6. *Consider a graph $\mathcal{G}_{ex3}$ as shown in Figure 4.5(a). The path detection algorithm is*



(a) Directed network $\mathcal{G}_{ex3}$ with *critical edges* (red dotted lines)



(b) Network $\mathcal{G}_{ex4}$ without critical edge *by adding edges* (green dot-dashed line)

Figure 4.5: Illustration of critical edge in directed network

illustrated for node 4.

Initialization *Initially, each node only knows the path to itself:*

$$\bar{x}_4(0) = [\{0\}, \{0\}, \{0\}, \{[4]\}, \{0\}, \{0\}, \{0\}, \{0\}]$$

First Iteration *(*$k = 1$*) Nodes exchange path information with their direct neighbors. After the first iteration, the paths are updated as follows:*

$$\bar{x}_{4,2}(1) = \{[2, 4]\},$$

$$\bar{x}_{4,3}(1) = \{[3, 4]\},$$

$$\bar{x}_{4,5}(1) = \{[5, 4]\}$$

Second Iteration *(*$k = 2$*) Nodes continue to exchange and update paths. After the second iteration, the paths for node 4 are updated to:*

$$\bar{x}_{4,1}(2) = \{[1, 2, 4]\},$$
$$\bar{x}_{4,2}(2) = \{[2, 4], [2, 3, 4]\},$$
$$\bar{x}_{4,3}(2) = \{[3, 4]\},$$
$$\bar{x}_{4,4}(2) = \{[4]\},$$
$$\bar{x}_{4,5}(2) = \{[5, 4]\},$$
$$\bar{x}_{4,6}(2) = \{[6, 5, 4]\},$$
$$\bar{x}_{4,7}(2) = \{[7, 5, 4]\}$$

Final Iteration *(*$k = 9$*) In the final iteration, the paths are updated one last time to cover all reachable nodes:*

$$\bar{x}_{4,1}(9) = \{[1, 2, 4], [1, 2, 3, 4]\},$$
$$\bar{x}_{4,2}(9) = \{[2, 4], [2, 3, 4]\},$$
$$\bar{x}_{4,3}(9) = \{[3, 4], [3, 1, 2, 4]\},$$
$$\bar{x}_{4,4}(9) = \{[4]\},$$
$$\bar{x}_{4,5}(9) = \{[5, 4]\},$$
$$\bar{x}_{4,6}(9) = \{[6, 5, 4], [6, 7, 5, 4]\},$$
$$\bar{x}_{4,7}(9) = \{[7, 5, 4]\},$$
$$\bar{x}_{4,8}(9) = \{[8, 6, 5, 4], [8, 6, 7, 5, 4]\},$$
$$\bar{x}_{4,9}(9) = \{[9, 8, 6, 5, 4], [9, 8, 6, 7, 5, 4]\}$$

Result *At the end of the algorithm, node 4 has a comprehensive list of paths from itself to all other reachable nodes without cycles. This ensures that the path detection is accurate and the network connectivity is fully mapped.*

Identification of Critical Edges in Strongly Connected Digraphs

Lemma 6 utilizes the update law described in (4.11) to enable each node i to determine critical edges within n iterations.

Lemma 8. *In a strongly connected directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, after executing the update law for n iterations: 1. An edge $e_{k,i}$ from node k to node i is critical if $\bar{n}_{i,k}(n) = 1$, indicating it is the sole path between k and i. 2. An edge $e_{j,k}$ is critical if all path sequences in $\bar{x}_{i,j}$ consistently show j directly preceding k, confirming j to k as an essential link in the paths to i.*

Proof: 1. For edge $e_{k,i}$, its criticality is confirmed if it is the only path from k to i, demonstrated by $\bar{n}_{i,k}(n) = 1$. This uniqueness marks it as critical since its removal would disconnect k from i. 2. For edge $e_{j,k}$, if every path from j to i must pass through k, indicated by all paths in $\bar{x}_{i,j}$ showing j followed by k, then removing $e_{j,k}$ disrupts the connectivity from j to i, establishing its criticality.

The verification process is computationally efficient for determining if $\bar{n}_{i,k}(n) = 1$, a simple lookup operation with a complexity of $O(1)$ per in-neighbor, summed over d, the number of in-neighbors, resulting in a total complexity of $O(d)$. The check for criticality in the second case involves iterating through path sequences, scaling the complexity to $O(p \cdot l \cdot n)$, where p denotes the number of paths, l the path length, and n the number of iterations, reflecting a thorough yet efficient examination of network paths and their critical components.

Eliminating Critical Edges

Each node i identifies the subset of agents $\bar{\mathcal{N}}_i^k$ that rely exclusively on node k to connect to i. This subset comprises nodes j for which every path in $\bar{x}_{i,j}$ includes the sequence $[k, i]$. Based on the size of this subset, $\bar{n}_i^k$, the approach to establishing new connections varies:

If $\bar{n}_i^k = 0$, implying no direct dependency on k by any nodes to reach i, node i instructs the farthest in-neighbor in the network, denoted as j' (which does not belong to $\bar{\mathcal{N}}_i^k$), to establish a direct connection to k, thus forming the edge $e_{j',k}$. Conversely, if $\bar{n}_i^k = n - 2$, indicating that nearly all nodes depend on the critical edge $e_{k,i}$, then i requests its farthest in-neighbor, denoted as j'' (which is part of $\bar{\mathcal{N}}_i^k$), to send information directly to i, leading to the creation of the edge $e_{i,j''}$. For intermediate cases where $1 < \bar{n}_i^k < n - 2$, node i seeks to enhance connectivity by having the farthest in-neighbor within $\bar{\mathcal{N}}_i^k$, named j'', connect to the farthest in-neighbor not within $\bar{\mathcal{N}}_i^k$, named j'. This action results in the establishment of the edge $e_{j',j''}$.

This strategic modification of network connections not only mitigates the risk posed by critical edges but also bolsters the overall robustness of the communication framework within $\mathcal{G}$.

Example 7. *Graph $\mathcal{G}_{ex3}$ in Figure 4.5(a) shows all the identified critical edges in red and the graph $\mathcal{G}_{ex4}$ in Figure 4.5(b) shows the addition of edges to remove the criticality in the edges.*

To facilitate understanding, $\mathcal{G}_{ex4}$ is represented as a simplified graph in Figure 4.6 where each node represents a group of nodes that were originally strongly connected in the original graph. In this simplified representation, the structure of the graph is easier to analyze.

Initially, without the additional edges, the simplified graph $\mathcal{G}_{ex4}$ does not contain any cycles. This means that removing a single edge could disrupt the connectivity between these groups.

However, once the additional edges are introduced, the structure of the graph changes significantly.

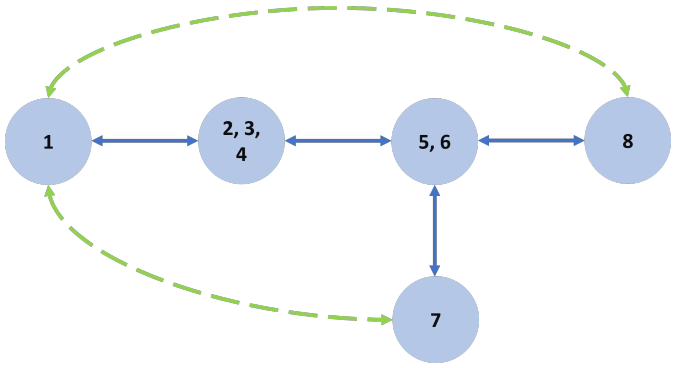


Figure 4.6: Simplified representation of graph $\mathcal{G}_{ex4}$ where each node is a collection of nodes that are originally connected

These added edges create cycles in the graph, ensuring that every node (or group of nodes) is part of at least one cycle. This transformation increases the robustness of the graph by ensuring that there are multiple paths between any two groups of nodes.

By ensuring that every node becomes part of a cycle, a minimum edge connectivity of 2 is achieved. This means that at least two edges need to be removed to disconnect any part of the graph. As a result, the criticality of individual edges between the groups of nodes is greatly reduced. The graph becomes more resilient to edge failures because the presence of cycles provides alternative paths for communication or flow between the nodes.

Illustrative Example

The following section presents an example and simulation-based demonstration of the proposed algorithms. Consider a network with 15 nodes, connected in a graph, say $\mathcal{G}_e$ as shown in 4.7, with a total of 20 edges.

The Algorithm 1 executes for 15 steps. By the 15^{th} step, each node i determines its neighbor

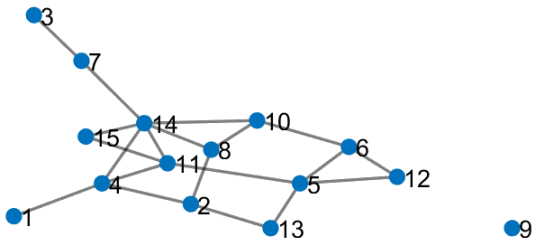


Figure 4.7: A random network $\mathcal{G}_e$, of 15 nodes and 20 edges

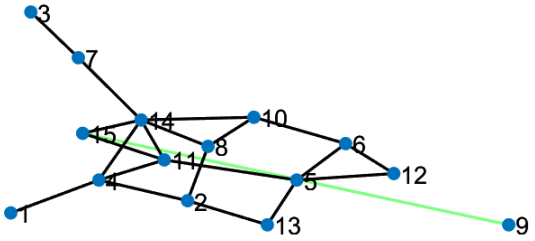
structure of the network $\mathcal{G}_e$ through equation (2.27). It also identified that $\mathcal{G}_e$ is not connected. Then within the next 15 steps Algorithm 2 adds a new edge to establish connectivity, as shown in Figure 4.8(a).

In the next 2 steps, each node i identifies the critical edges connected to i. The critical edges identified by each node i are as follows:

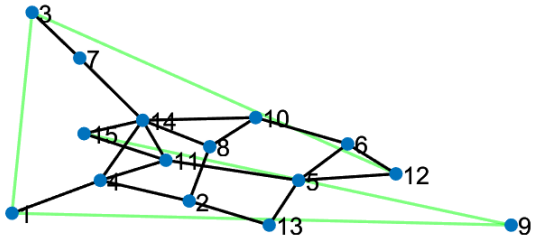
$$\mathcal{N}_1^c = \{(1,4)\} \qquad \mathcal{N}_3^c = \{(3,7)\} \qquad \mathcal{N}_4^c = \{(4,1)\}$$
$$\mathcal{N}_7^c = \{(7,3)\} \qquad \mathcal{N}_7^c = \{(7,14)\} \qquad \mathcal{N}_9^c = \{(9,15)\}$$
$$\mathcal{N}_{14}^c = \{(14,7)\} \qquad \mathcal{N}_{15}^c = \{(15,19)\}$$

and for the rest of the nodes $\mathcal{N}_i^c = \{\}$

The next 1 step (33^{rd} step) is to identify the corresponding augmentation nodes, which is performed only by the nodes associated with the critical edges. That is, nodes $\{1,3,4,7,9,14,15\}$ and it



(a) From Algorithm 2



(b) From Algorithm 4

Figure 4.8: Augmented network $\mathcal{G}'_e$

follows:

$$\mathcal{N}_1^r = \{(1,3)\} \qquad \mathcal{N}_3^r = \{(3,12)\} \qquad \mathcal{N}_4^r = \{(3,1)\}$$

$$\mathcal{N}_7^r = \{(3,12) \qquad \mathcal{N}_9^r = \{(9,1)\} \qquad \mathcal{N}_{14}^r = \{(12,3)\}$$

$$(12,3)\} \qquad \mathcal{N}_{15}^r = \{(1,9)\}$$

In the following 14 steps, the information is propagated to all the other nodes, and the augmentation node set for each node is $\mathcal{N}_i^r = \{(1,3),(3,12),(3,1),(12,3),(9,1),(1,9)\}$.

In the final 48^{th} step, every node present in $\mathcal{N}^r$ reaches out to its pair and forms the edge. The augmented network is shown in Figure 4.8.

Eigenvalue Analysis

For a multi-agent system represented by a graph $\mathcal{G}$ with Laplacian matrix L, the eigenvalues λ_i of L provide crucial insights into the network's properties. The second smallest eigenvalue λ_2, known as the algebraic connectivity, is particularly important.

$$\lambda_2 = \min_{\mathbf{v} \perp \mathbf{1}} \frac{\mathbf{v}^T L \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \tag{4.13}$$

A higher λ_2 indicates a more robust and connected network. The distributed algorithms aim to maximize λ_2 by eliminating critical edges and adding redundant paths, enhancing network resilience.

Summary

In this chapter, the identification and reinforcement of critical edges in both directed and undirected networks were explored to enhance their resilience against disruptions. Two distinct approaches were introduced: one tailored for undirected networks and the other for directed networks. The first approach involved a series of distributed algorithms to ensure connectivity, identify critical edges, and augment the network to eliminate these critical vulnerabilities. The second approach extended these principles to directed networks, focusing on maintaining strong connectivity and robustness.

The proposed solutions were validated through numerical examples and simulations, highlighting their effectiveness in maintaining network connectivity and robustness. In summary, this chapter provides a comprehensive framework for enhancing network resilience through distributed algo-

rithms, offering robust solutions to maintain connectivity and consensus in the face of adversarial attacks and dynamic network conditions.

CHAPTER 5: ADVANCED APPROACHES FOR LARGE-SCALE NETWORKS IN PRACTICAL APPLICATIONS

Introduction

This chapter explores advanced methods for improving the robustness and efficiency of large-scale networked systems using graph theory-based connectivity analysis and cooperative control. The focus is on simplifying large graphs into smaller, manageable components to identify critical edges and reduce computational complexity. This is crucial for enhancing the resilience of cyber-physical systems (CPS) in applications like communication networks, power grids, and transportation systems.

The chapter includes a case study that uses directed graphs (digraphs) to model interactions within a CPS. By decomposing a large, strongly connected digraph into multiple strongly connected components (SCCs), critical edges that affect network connectivity can be efficiently identified and analyzed. This method is applicable to various domains, including:

- **Smart Grids**: Achieving consensus among distributed energy resources for stable power distribution.
- **Multi-Robot Systems**: Coordinating movements and tasks to meet collective goals [86].
- **Financial Networks**: Ensuring secure and validated transactions [87].
- **Autonomous Vehicle Networks**: Preventing collisions and optimizing routes [88].

The process of breaking down large graphs into SCCs, identifying in-neighbors, calculating information numbers, and mapping connectivity is detailed. These methods enhance network robustness

and provide practical insights for real-world applications.

By linking theory to practice, this chapter shows how advanced control strategies and graph theory-based methods can improve the performance and resilience of teleoperation and multi-agent systems in various scenarios.

Reducing Large Graphs

Identifying critical edges through the algorithm can become computationally intensive when managing large graphs due to increased time and operational complexities. It is often advantageous to simplify the large graph into a smaller, equivalent representation to enhance efficiency. This approach reduces the computational load while preserving the essential properties of the original network, thereby enabling more effective and faster analysis of critical edges.

The process of converting a strongly connected graph into multiple SCCs involves four key steps. Initially, a node j is selected to stop sending information, causing the graph to break into multiple SCCs. This initial step simplifies the graph, which typically reduces to a minimal form consisting of a single node without edges. Next, the in-neighbors of each node are identified through an n-step algorithm, previously detailed, that determines each node's awareness of its network connections. Following this, each node calculates its information numbers to identify its respective SCC and elect virtual leaders, simplifying the network architecture into fewer, distinctive SCCs. Finally, the connectivity within and between SCCs is mapped and analyzed using an iterative update protocol that determines direct connections and the relative structure between SCCs, enhancing the understanding of the network's overall structural dynamics and interactions.

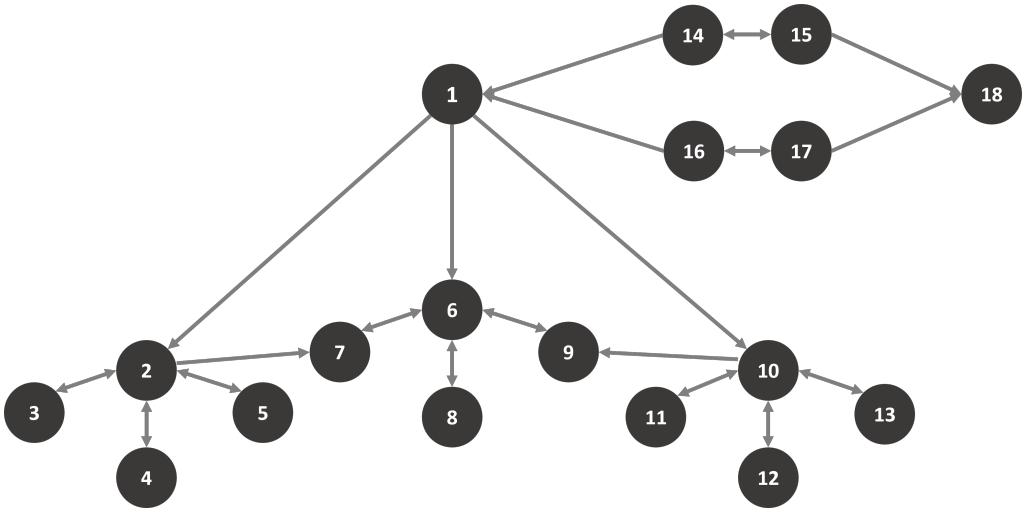


Figure 5.1: Example of a digraph representation ($\mathcal{G}_{cps}$)

Algorithm to Identify the SCCs

In this subsection, an algorithm designed to calculate the information numbers for agents within a network is introduced, enabling each agent to identify its associated SCC. The state of each agent i is represented as a vector η_i, which evolves over discrete time steps k ranging from n to $2n-1$:

$$\eta_i(k+1) = \begin{bmatrix} \eta_{i,1}(k+1) & \cdots & \eta_{i,n}(k+1) \end{bmatrix}.$$

The distributed protocol for updating these information numbers is defined as follows:

$$\eta_{i,j}(k+1) = \max_{l \in \mathcal{N}_i^c \cup \{i\}} \eta_{l,j}(k), \tag{5.1}$$

where $\mathcal{N}_i^c$ represents the in-neighbor set of node i, including node i itself. The initial condition for

this recursive update is:

$$\eta_{i,j}(n) = \begin{cases} \mathbf{1}_n^T \xi_i(n), & \text{if } j = i \\ 0, & \text{otherwise} \end{cases},$$

which allows every node to determine its own SCC as given by the matrix $\mathcal{SCC} = \begin{bmatrix} \mathcal{SCC}_1 & \cdots & \mathcal{SCC}_n \end{bmatrix}$. Each SCC is composed of nodes j such that:

$$\mathcal{SCC}_l = \{j \in \mathcal{N} : \ \zeta_{l,j}(n) = 1 \text{ and } \eta_{l,j}(2n) = \eta_{l,l}(2n)\}. \tag{5.2}$$

and are identified by a virtual leader, defined as the node with the maximum index of all nodes in that SCC, $V_i^* = \max_j\{j : \forall j \in SCC_i\}$. This virtual leadership structure evolves iteratively, building a unique set of leaders $V^* = V^*(n)$ of elements $V_{i^*}^* \triangleq V^*(SCC_i)$, where $V^*(i+1)$ includes a new leader V_{i+1}^* only if it is not already present in $V^*(i)$. That is,

$$V^*(i+1) = \begin{cases} V^*(i) \cup V_{i+1}^* & \text{if } V_{i+1}^* \notin V^*(i) \\ V^*(i) & \text{else} \end{cases}, \quad i = 1, \cdots, n.$$

The process starts from an empty set $V^*(0) = \emptyset$ and progresses similarly for the set of distinctive SCCs, $\mathcal{SCC}^* \triangleq \mathcal{SCC}^*(n)$, with $\mathcal{SCC}^*(i+1)$ including a new SCC only if it's not already present in $\mathcal{SCC}^*(i)$, that is $\mathcal{SCC}^*(0) = \emptyset$. Then,

$$\mathcal{SCC}^*(i+1) = \begin{cases} \mathcal{SCC}^*(i) \cup SCC_{i+1} & \text{if } SCC_{i+1} \notin \mathcal{SCC}^*(i) \\ \mathcal{SCC}^*(i) & \text{else} \end{cases}, \quad i = 1, \cdots, n.$$

Generally, the overall composition of SCC^* or V^* may not be fully known to individual SCCs or their constituent nodes. In many cases, the cardinality of these sets, $|SCC^*|$ or $|V^*|$, is significantly smaller than n, indicating a more compact network structure than originally perceived. This model

enables agents to gain insights into not only their immediate network environment but also how they integrate into the broader network topology through their identified SCCs.

By recognizing and understanding the hierarchical structure and interconnections of SCCs, agents can better navigate and optimize the network's overall functionality. This compact representation of the network reduces complexity and enhances the ability to manage and analyze the network efficiently.

Example 8. *Consider the graph $\mathcal{G}_{cps}$ in Figure 5.1.*

1. *Identify in-neighbors (18 steps) The process of identifying all in-neighbors ξ_i takes n steps. In this example, it requires 18 steps.*

   For Node 1:

   First in-neighbors are $[14, 16]$ and final state

$$\xi_1(18) = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0]$$

   For Node 6:

   First in-neighbors are $[1, 7, 8, 9]$ and final state

$$\xi_6(18) = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0]$$

2. *Finding number of connected nodes **Initialization** ($k = 18$) At the 18th step, the state η_i is initialized as follows for example nodes 1 and 6:*

$$\eta_1(18) = [5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$\eta_6(18) = [0, 0, 0, 0, 0, 17, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

After 1 iteration *(*$k = 19$*) Nodes* 1 *and* 18 *receive corresponding values from their in-neighbors and update the states as follows:*

$$\eta_1(19) = [5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0]$$
$$\eta_6(19) = [5, 0, 0, 0, 0, 0, 17, 17, 17, 17, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

Final state, after n ***iterations*** *(*$k = 36$*) After completing* n *iterations, nodes* 4 *and* 5 *receive values from all their in-neighbors and update their states as follows:*

$$\eta_1(36) = [5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0]$$
$$\eta_6(36) = [5, 9, 9, 9, 9, 9, 17, 17, 17, 17, 9, 9, 9, 9, 2, 2, 2, 2, 5]$$

The states for the rest of the nodes are updated similarly:

3. *SCC grouping*

   The SCCs are then identified by each node as

$$\mathcal{SCC}_1 = [1]$$
$$\mathcal{SCC}_2, \mathcal{SCC}_3, \mathcal{SCC}_4, \mathcal{SCC}_5 = [2, 3, 4, 5]$$
$$\mathcal{SCC}_6, \mathcal{SCC}_7, \mathcal{SCC}_8, \mathcal{SCC}_9 = [6, 7, 8, 9]$$
$$\mathcal{SCC}_{10}, \mathcal{SCC}_{11}, \mathcal{SCC}_{12}, \mathcal{SCC}_{13} = [10, 11, 12, 13]$$
$$\mathcal{SCC}_{14}, \mathcal{SCC}_{15} = [14, 15]$$
$$\mathcal{SCC}_{16}, \mathcal{SCC}_{17} = [16, 17]$$
$$\mathcal{SCC}_{18} = [18]$$

4. *Assigning virtual leaders*

The virtual leaders are nodes with the maximum index in each SCC as follows:

$$V^* = \{1, 5, 9, 13, 15, 17, 18\}$$

5. *Identifying SCCs*

   The SCCs are identified as

$$\mathcal{SCC}^* = \{[1], [2, 3, 4, 5], [6, 7, 8, 9], [10, 11, 12, 13], [14, 15], [16, 17], [18]\}$$

Finding SCC structure

Next, an algorithm that employs state representations θ_i, θ_i^*, and ν_i^* to systematically identify and analyze the in-neighbor structure of SCC within a network is described. This algorithm enhances the understanding of SCC interactions and the broader network topology by iteratively calculating connectivity metrics.

State Representation: Each agent i in the network maintains the following state vectors, where $k = 2n, \cdots, 3n-1$

$$\begin{aligned}
&\theta_i(k+1) = \begin{bmatrix} \theta_{i,1}(k+1) & \cdots & \theta_{i,n}(k+1) \end{bmatrix}, \\
&\theta_i^*(k+1) = \text{a row vector of size equal to \# of SCCs}, \\
&\nu_i^*(k+1) = \text{a row vector of size equal to \# of SCCs}.
\end{aligned}$$

to map the connectivity to other SCCs, with θ_i^* focusing specifically on interactions between SCCs.

To analyze inter-SCC connectivity, θ_i and θ_i^* are updated using:

$$\theta_{i,j}(k+1) = \max_{l \in \mathcal{N}_i^c \cup \{i\}} \theta_{l,j}(k), \quad \theta_{i,j}(2n) = \begin{cases} 1, & \text{if } j \in \mathcal{SCC}_i \\ 0, & \text{otherwise} \end{cases}, \tag{5.3}$$

and

$$\theta^*_{V_i^*,V_j^*}(k+1) = \max_{\substack{l \in [(\mathcal{N}_{i^*}^c \cup \{i^*\}) \cap (\mathcal{N} - \mathcal{SCC}^*_{i^*})] \\ m \in [(\mathcal{N}_{j^*}^c \cup \{j^*\}) \cap (\mathcal{N} - \mathcal{SCC}^*_{j^*})]}} \theta_{l,m}(k) \tag{5.4}$$

to establish whether virtual leaders of different SCCs can reach each other through their respective members.

Depth Determination between SCCs: Finally, $\nu^*_{V_i^*,V_j^*}(k+1)$ calculates the 'depth' or minimum number of connections between different SCCs, modifying based on direct and discovered paths:

$$\nu^*_{V_i^*,V_j^*}(k+1) = \begin{cases} \nu^*_{V_p^*,V_j^*}(k) + 1, & \text{if } \theta^*_{V_i^*,V_p^*} = 1 \text{ for some } p^* \neq i^* \text{ and} \\ & \quad \theta_{l,m}(k+1) - \theta_{q,m}(k) = 1 \\ & \quad \text{for some } l \in \mathcal{SCC}^*_{V_i^*}, q \in \mathcal{SCC}^*_{V_p^*}, \text{ and} \\ & \quad m \in \mathcal{SCC}^*_{V_j^*}, \\ \nu^*_{V_i^*,V_j^*}(k), & \text{otherwise} \end{cases} \tag{5.5}$$

Example 9. *Consider the graph $\mathcal{G}_{cps}$ in Figure 5.1.*

For each virtual leader, the states θ^* and ν^* after 18 iterations are as follows:

$$\theta_4^* = [1, 0] \quad \nu_4^* = [0, 0]$$

$$\theta_8^* = [1, 1] \quad \nu_8^* = [1, 0]$$

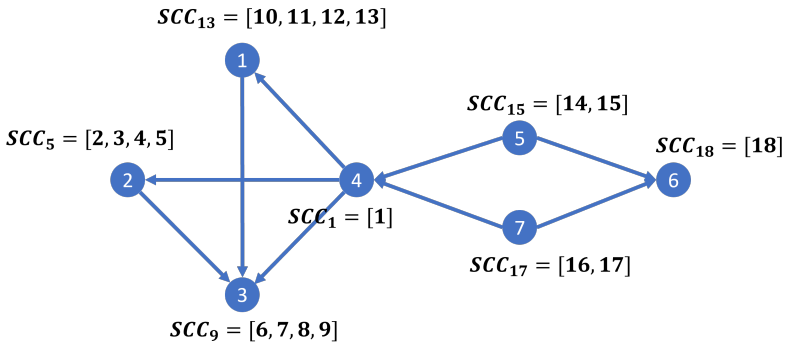


Figure 5.2: Reduced graph representation of the cyber-physical system $\mathcal{G}_{cps}$

The overall graph $\mathcal{G}_{ex5}$ is now reduced to the graph in Figure 5.2.

Enabling multiple SCCs

The connectivity of a digraph $\mathcal{G}$ can be determined by its structure of the SCCs. A digraph $\mathcal{G}$ is strongly connected if it comprises only one SCC. If $\mathcal{G}$ is not strongly connected, its general neighboring structure and the relationships between its SCCs can be described using specific metrics.

For a given station i in a digraph $\mathcal{G}$, the results $\mathcal{SCC}_i$ and $\nu_i(3n)$ from (5.1) and (5.5) enable station i to identify all other stations within its SCC, its in-neighboring SCCs, and the depth(s) to its source SCC(s). If station i is part of block E'_{ll} in a canonical block decomposition, it can access all matrices on the j-th block row and those diagonal blocks E'_{mm} that correspond to nonzero E'_{ml} entries.

Lemma 9. *Assume $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is a strongly connected directed graph. Consider a scenario where $\mathcal{G}_j = (\mathcal{N}, \mathcal{E}_j)$ represents a virtual copy of $\mathcal{G}$ in which an agent, say j, does not transmit information to other agents. The reduced-order graph generated using the algorithm outlined in* (2.26)-(2.28) *and* (5.1)-(5.5) *namely $\mathcal{G}_j^* = (\mathcal{N}^*, \mathcal{E}_j^*)$ exhibits the following properties:*

- *$\mathcal{G}_j^*$ contains more than one SCC, demonstrating a lack of strong connectivity.*
- *The virtual leader of the SCC to which j belongs, denoted $\mathcal{SCC}_{V_j^*}^*$, acts as a sink node.*

Proof: Given that node j refrains from sending information, $\mathcal{G}_j$ loses its strong connectivity and naturally divides into multiple SCCs. Despite the original graph $\mathcal{G}$ being strongly connected, the specific SCC containing node j cannot exist in isolation due to the interconnected nature of the graph. Thus, it must at least act as a sink within the reduced graph $\mathcal{G}_j^*$.

Properties of Reduced Graphs

The properties of the reduced graph are as follows:

- The algorithm transforms any graph $\mathcal{G}$ into a reduced graph $\mathcal{G}^* = (\mathcal{N}^*, \mathcal{E}^*)$, which includes fewer nodes and edges, specifically $n^* \leq n$ nodes and $m^* \leq m$ edges. This simplification maintains essential connectivity properties while reducing complexity.
- Each node in $\mathcal{G}^*$ acts as a virtual leader for one of the SCCs in $\mathcal{G}$. These virtual leaders, represented as $\mathcal{SCC}_{V_i^*}^*$, encompass all nodes from a specific SCC, where V_i^* is the highest index within that SCC.
- Nodes in $\mathcal{G}^*$ may not adhere to numerical order due to their organization based on SCCs, reflecting the hierarchical structure rather than numerical sequencing.

- Edges in $\mathcal{G}^*$, denoted as $e^*_{(\mathcal{SCC}^*_{V_i^*}, \mathcal{SCC}^*_{V_k^*})}$, establish communication links between the SCCs, indicating the directional flow of information or influence between components.

- If $\mathcal{G}$ is strongly connected, the reduction results in a simplified representation with a single node $\mathcal{SCC}^*_{V_i^*}$ and no edges, showcasing the graph's strong connectivity in a unified form.

While it seems like the overall process takes multiple discreet time steps, it can be completed instantaneously in continuous time domain

Case Study

This dissertation represents a cyber-physical system (CPS) using a directed graph to model the interactions and dependencies between various components. Each node in the digraph represents a physical or computational entity, such as sensors, actuators, or processing units, while directed edges illustrate the communication pathways and control flows between these entities. This digraph representation is versatile and can be applied to a range of real-life applications, such as distributed energy management in smart grids, multi-robot systems for coordinated task execution, financial network systems, autonomous vehicle coordination, sensor networks for environmental monitoring, load balancing in distributed computing, and collaborative filtering in recommendation systems. In each of these applications, consensus plays a crucial role by ensuring that all nodes in the network agree on critical parameters, thereby enabling coordinated and efficient operation. For instance, in a smart grid, consensus algorithms help distributed energy resources agree on power generation levels to maintain grid stability. In multi-robot systems, consensus ensures coordinated movements and task allocation to achieve mission objectives. Similarly, in financial networks, consensus is used to validate transactions and update distributed ledgers, while in autonomous vehicle networks, it prevents collisions by coordinating vehicle routes. In sensor networks, con-

sensus ensures accurate environmental data by aggregating measurements from multiple sensors. In distributed computing, it balances the load across servers to optimize performance, and in recommendation systems, it aggregates user preferences to provide personalized suggestions. Thus, the digraph representation not only models the structural complexity of a CPS but also highlights the importance of consensus in achieving reliable and efficient system operation.

Smart Grid

A smart grid is an electrical grid system that uses digital communication technology to monitor and manage the generation, distribution, and consumption of electricity more efficiently and reliably. It integrates various renewable energy sources, traditional power plants, and energy storage systems with consumers and smart devices.

Representation as a Graph

- **Nodes (Vertices)**: Each node in the graph represents a component of the smart grid. This includes power plants, substations, transformers, smart meters, and consumer appliances.
- **Edges (Links)**: The edges represent the communication and power lines connecting these components. They can indicate the flow of electricity, data, or control signals between nodes.

Figure 5.3 illustrates a smart grid as an example of a cyber-physical system, highlighting the integration of various energy sources, communication links, and real-time management for efficient electricity distribution.

Importance of Consensus

In the context of a smart grid, consensus is crucial for several reasons:

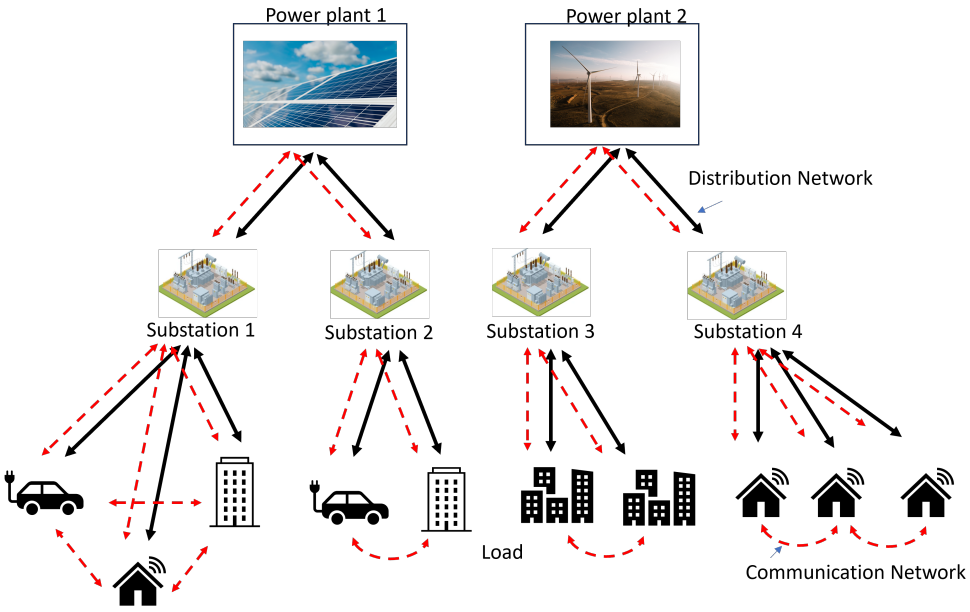


Figure 5.3: A smart grid as an example of a cyber-physical system.

1. *Load Balancing*: To ensure the grid operates efficiently, it is essential to balance the load across various power generation sources and consumers. Consensus algorithms help in coordinating the distributed energy resources to match supply with demand dynamically.

2. *Frequency Regulation*: Maintaining the frequency of the power grid within a specific range is vital for stability. Consensus mechanisms enable distributed components to agree on adjustments needed to keep the frequency stable.

3. *Fault Tolerance and Recovery*: In the event of faults or outages, consensus algorithms help in reconfiguring the grid to isolate the affected areas and restore service without central control, improving resilience.

4. *Demand Response*: Consensus is used to implement demand response strategies where consumers adjust their energy usage in response to supply conditions, price signals, or incentives. This requires agreement among distributed entities to reduce or shift consumption.

Example Scenario Consider a scenario where several distributed energy resources (DERs), such as solar panels and wind turbines, are connected to a smart grid. The output from these DERs can fluctuate based on weather conditions. To maintain grid stability, the system needs to achieve consensus on how to adjust the output of conventional power plants and how much energy to draw from or store in batteries. This requires continuous communication and agreement among the DERs, power plants, and control systems.

Now, imagine the system is under a DoS attack, which disrupts the communication channels between these components. Such an attack can cause delays or complete loss of communication, leading to instability in the grid as the DERs, power plants, and control systems can no longer coordinate effectively. This lack of coordination can result in power imbalances, overloading certain parts of the grid while underutilizing others, potentially leading to blackouts or damage to infrastructure.

To mitigate the effects of DoS attacks, the proposed algorithms are designed to ensure resilience and stability in the network. By implementing these algorithms, the smart grid can maintain stable and efficient operation even under the stress of a DoS attack. The network identification algorithm identifies disruptions, while the dynamic distributed consensus algorithm work together to reroute communications and ensure all components remain synchronized. This resilience ensures the grid can continue to balance load, regulate frequency, and respond to demand dynamically, maintaining overall stability and efficiency.

Impact of Malicious Nodes on Neighbor Identification Algorithm

The distributed algorithms presented in this paper are designed to identify DoS attacks and provide a resilient approach to handle such disruptions. However, if a malicious node is present, these mechanisms might not be sufficient to ensure accurate network structure identification.

In general, the presence of malicious nodes can significantly impact the accuracy and reliability of the network structure identification in the proposed distributed algorithms. A malicious node can misreport its state values, potentially leading to incorrect estimation of in-neighbor structures and shortest path distances. If a malicious node $m \in \mathcal{N}$ reports incorrect $\xi_{m,j}(k)$ or $\omega_{m,j}(k)$ values, it can cause its neighboring nodes to propagate false information throughout the network. This misinformation can result in:

- Incorrect identification of in-neighbor relationships, where legitimate nodes are falsely identified or omitted as in-neighbors.
- Incorrect calculation of shortest path distances, leading to suboptimal routing and communication delays.
- Potential network partitioning, where certain nodes may be isolated due to the propagation of false connectivity information.

To mitigate the impact of malicious nodes, additional mechanisms such as redundant checks, trust-based algorithms, or consensus protocols can be incorporated to validate the integrity of the reported state values. These mechanisms can help ensure that the network's connectivity and in-neighbor structures are accurately identified, even in the presence of adversarial behavior.

For example, consider the neighbor identification algorithm from Example 1.

Numerical Example with Malicious Node

To illustrate the impact of a malicious node, let's assume node 5 in the given network acts maliciously by misreporting its state values.

Initial Setup with Malicious Node

For each node $i \in \mathcal{N}$, initialize the following states as before:

$$\xi_i(0) = [\xi_{i,1}(0) \ \cdots \ \xi_{i,n}(0)]^T \in \mathbb{R}^n$$

$$\omega_i(0) = [\omega_{i,1}(0) \ \cdots \ \omega_{i,n}(0)]^T \in \mathbb{R}^n$$

However, node 5 will now report incorrect values during the iterations.

Iterative Steps with Malicious Node

During each iteration, node 5 will misreport its $\xi_{5,j}(k)$ and $\omega_{5,j}(k)$ values to disrupt the algorithm. For example, node 5 may report a higher or lower value than the actual distance.

Impact on Node 4

Initial State:

- $\xi_4(0) = [0, 0, 0, 1, 0, 0, 0, 0]$
- $\omega_4(0) = [\infty, \infty, \infty, 0, \infty, \infty, \infty, \infty]$

After 1st Iteration:

$$\xi_4(1) = [\max(0,1), \max(0,1), \max(0,1), 1, \max(0,1), 0, 0, 0]$$

$$\xi_4(1) = [1, 1, 1, 1, 1, 0, 0, 0]$$

$$\omega_4(1) = [1, 1, 1, 0, 1, \infty, \infty, \infty]$$

State after 2nd Iteration with Malicious Node 5:

$$\xi_4(2) = [\max(1, \max(1, 0)), \cdots, \cdots, 1, \cdots, \max(1, 0), \max(1, 0), 0]$$

$$\xi_4(2) = [1, 1, 1, 1, 1, 1, 1, 0]$$

$$\omega_4(2) = [1, 1, 1, 0, \mathbf{2}, 2, 2, \infty] \quad \text{(Incorrect due to malicious node)}$$

Due to the malicious behavior of node 5, the shortest path calculations for node 4 and potentially other nodes are affected, leading to incorrect $\omega_{i,j}(k)$ values. This demonstrates how a malicious node can disrupt the network's connectivity information.

To mitigate such issues, additional mechanisms such as redundant checks, trust-based algorithms, or consensus protocols can be incorporated to validate the integrity of the reported state values. These mechanisms can help ensure that the network's connectivity and in-neighbor structures are accurately identified, even in the presence of adversarial behavior.

Summary

This chapter has explored advanced methods for enhancing the robustness and efficiency of large-scale networked systems through graph theory-based connectivity analysis and cooperative control. By simplifying large graphs into smaller, manageable components, the identification of critical edges and reduction of computational complexity were demonstrated, thereby improving the resilience of CPS in various applications, such as smart grids, multi-robot systems, financial networks, and autonomous vehicle networks.

A detailed case study was provided, illustrating the decomposition of a large, strongly connected digraph into multiple SCCs, and emphasizing the importance of consensus in maintaining coordi-

nated and efficient operation. This approach is crucial in real-life scenarios where system stability and reliability are paramount, especially under the threat of DoS attacks.

The proposed algorithms, including those for instantaneous detection and dynamic activation of communication layers, ensure network resilience and stability, even under adverse conditions. By applying these methods, systems can maintain critical functionalities and avoid catastrophic failures, ensuring continuous operation and optimal performance.

Overall, this chapter bridges the gap between theoretical advancements and practical implementations, showcasing how sophisticated control strategies and graph theory-based methods can significantly enhance the performance and resilience of teleoperation and multi-agent systems in various real-world scenarios.

CHAPTER 6: CONCLUSION

Summary of Contributions

This dissertation investigated robust and resilient control strategies for networked systems, focusing on teleoperation and multi-agent systems affected by significant time delays and DoS attacks. The primary objective was to develop methods ensuring stability and performance despite these adversities, contributing novel frameworks and algorithms that enhance the robustness of these systems.

A framework was introduced for designing control systems to maintain stability and transparency despite time-varying communication delays and environmental disturbances. By developing a novel passivity-shortage framework, the limitations of traditional passivity-based approaches were addressed. The proposed strategies were evaluated through simulations, demonstrating their effectiveness in achieving consensus while maintaining performance and stability.

Additionally, a suite of distributed algorithms aimed at ensuring network connectivity and eliminating critical edges in multi-agent systems was introduced. These algorithms operate concurrently to determine neighbor structures, add necessary edges, and identify and address critical nodes and edges that could compromise network robustness. By implementing these methods, the overall resilience of multi-agent systems can be enhanced, ensuring their reliable operation even in the face of potential failures and attacks. The effectiveness of these algorithms was demonstrated through illustrative examples and simulations, providing a solid foundation for further research and application in various domains. The extension of these algorithms to directed graphs further underscores their versatility and scalability, making them suitable for a wide range of networked systems.

Practical Applications and Effectiveness

The practical application and effectiveness of the methodologies developed in this dissertation were demonstrated through experimental setups and detailed case studies. The bilateral teleoperation experiment using Phantom Omni haptic devices validated the stability and performance improvements under variable-time delays, while the case study on cyber-physical systems showcased the robustness and resilience of multi-agent systems against DoS attacks. Additionally, the reduction of large graphs into smaller, manageable components highlighted the efficiency of the proposed algorithms in analyzing network robustness. These practical evaluations affirm the theoretical contributions of this work, offering robust solutions for enhancing the performance and stability of teleoperation systems and networked multi-agent systems in real-world applications.

Relating Resilient Multi-Agent Systems to Teleoperation Systems

The strategies and algorithms for enhancing the resilience of multi-agent systems against DoS attacks are closely related to passivity-shortage-based control used in teleoperation systems. Both approaches focus on maintaining stability and robust performance despite disruptions. In teleoperation systems, passivity-shortage-based control ensures stable interactions between a human operator and a remote environment, balancing input and output energies to manage disturbances. Similarly, distributed algorithms in multi-agent systems identify and eliminate critical edges to maintain network connectivity, allowing the system to reconfigure itself during attacks. Both methodologies emphasize maintaining system integrity under adverse conditions, with stability in teleoperation analyzed using passivity-shortage and eigenvalue stability, and robustness in multi-agent systems analyzed using the network's Laplacian matrix eigenvalues. Redundancy and adaptability are key in both areas, achieved through multiple control paths in teleoperation and additional communi-

cation links in multi-agent systems. This dissertation contributes to a broader framework for designing resilient systems that can adapt to and recover from disruptions, bridging the gap between control theory and network resilience.

Future Work

Future work should explore the simultaneous mitigation of False Data Injection (FDI) attacks and DoS attacks, as these pose significant threats to the stability and security of networked control systems. Investigating the interplay between FDI and DoS attacks and their combined impact on multi-agent systems will be crucial for developing comprehensive defense mechanisms. Research could focus on designing robust detection and mitigation algorithms that can identify and neutralize both types of attacks in real-time, ensuring continuous operation and accurate consensus among agents. Additionally, exploring advanced machine learning techniques to predict and adapt to potential attack patterns can enhance the resilience of these systems. Integrating these strategies into practical applications, such as smart grids and autonomous vehicle networks, will be essential to validate their effectiveness and address real-world challenges. By tackling both FDI and DoS attacks concurrently, future research can significantly advance the security and robustness of critical networked control systems.

Conclusion

This dissertation addresses critical challenges in achieving stable and robust control of teleoperation systems with time-varying delays and multi-agent systems under Denial of Service (DoS) attacks. Through the development of novel control strategies and resilient algorithms, the research successfully mitigates the destabilizing effects of communication delays and enhances system ro-

bustness against cyber threats. By ensuring input-to-state stability, minimizing lag and errors, and maintaining L_2 stability, the proposed solutions significantly improve the performance and reliability of teleoperation systems. Additionally, the implementation of scalable, distributed algorithms enhances the resilience of multi-agent systems, safeguarding their functionality in the face of DoS attacks. The findings contribute to advancing the field of networked control systems, providing a comprehensive framework for addressing both structural vulnerabilities and cyber threats, thereby ensuring continuous and reliable operation in dynamic environments.

LIST OF PUBLICATIONS

1. D. Babu Venkateswaran and Z. Qu, ”Passivity-short bilateral teleoperation with communication delays,” 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 2018, pp. 1275-1281 [33].

2. D. Babu Venkateswaran and Z. Qu, ”A passivity-shortage based control design for teleoperation with time-varying delays,” in IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 4070-4077, July 2020 [76].

3. D. Babu Venkateswaran and Z. Qu, ”Distributed multilateral teleoperation framework using passivity-shortage,” in IFAC-PapersOnLine, 2019 Jan 1;52(20):181-6 [77].

4. A. Gusrialdi, D. B. Venkateswaran, and Z. Qu, “Enhancing resilience in cooperative systems against cyber-attacks: A defense framework through adaptive network reconfiguration and digital twin,” in Latest Adaptive Control Systems (D. P. Ioannou, ed.), Rijeka: IntechOpen, 2024 [85].

5. D. Babu Venkateswaran and Z. Qu, ”Resilient multi-agent systems against denial-of-service attacks via adaptively activatable network layers,” submitted in a special section: resilient and safe control in multi-agent systems, in IEEE Open Journal of Control Systems (OJ-CSYS) [89].

6. D. Babu Venkateswaran, Z. Qu and A. Gusrialdi, ”A distributed method for detecting critical edges and increasing edge connectivity in undirected networks,” submitted in 63rd IEEE Conference on Decision and Control [90].

7. D. Babu Venkateswaran and Z. Qu, ”Enhancing directed network robustness through critical edge detection and graph simplification,” under preparation [91].

LIST OF REFERENCES

[1] H. Tugal, J. Carrasco, P. Falcon, and A. Barreiro, "Stability analysis of bilateral teleoperation with bounded and monotone environments via zames–falb multipliers," *IEEE Transactions on Control Systems Technology*, vol. 25, pp. 1331–1344, July 2017.

[2] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006.

[3] M. Shahbazi, S. F. Atashzar, and R. V. Patel, "A systematic review of multilateral teleoperation systems," *IEEE Transactions on Haptics*, vol. 11, no. 3, pp. 338–356, 2018.

[4] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, Jan 2007.

[5] T. B. Sheridan and W. R. Ferrell, "Remote manipulative control with transmission delay," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-4, pp. 25–29, Sept 1963.

[6] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic Control*, vol. 34, pp. 494–501, May 1989.

[7] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *IEEE Transactions on Robotics*, vol. 22, pp. 269–281, April 2006.

[8] N. Chopra, "Passivity results for interconnected systems with time delay," in *47th IEEE Conference on Decision and Control*, pp. 4620–4625, Dec 2008.

[9] E. Nuño, L. Basañez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, vol. 47, no. 3, pp. 485 – 495, 2011.

[10] J. Yamauchi, M. W. S. Atman, T. Hatanaka, N. Chopra, and M. Fujita, "Passivity-based control of human-robotic networks with inter-robot communication delays and experimental verification," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 628–633, July 2017.

[11] D. Sun, F. Naghdy, and H. Du, "Wave-variable-based passivity control of four-channel nonlinear bilateral teleoperation system under time delays," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 238–253, Feb 2016.

[12] R. J. Anderson and M. W. Spong, "Asymptotic stability for force reflecting teleoperators with time delays," in *International Conference on Robotics and Automation*, pp. 1618–1625 vol.3, May 1989.

[13] T. Matiakis, S. Hirche, and M. Buss, "Control of networked systems using the scattering transformation," *IEEE Transactions on Control Systems Technology*, vol. 17, pp. 60–67, Jan 2009.

[14] G. Niemeyer and J.-J. E. Slotine, "Stable adaptive teleoperation," in *1990 American Control Conference*, pp. 1186–1191, May 1990.

[15] G. Niemeyer and J.-J. E. Slotine, "Towards force-reflecting teleoperation over the internet," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 3, pp. 1909–1915 vol.3, May 1998.

[16] R. Lozano, N. Chopra, and M. Spong, "Passivation of force reflecting bilateral teleoperators with time varying delay," in *in Proceedings of the 8. Mechatronics Forum*, pp. 24–26, 2002.

[17] F. Hashemzadeh, I. Hassanzadeh, M. Tavakoli, and G. Alizadeh, "A new method for bilateral teleoperation passivity under varying time delays," *Mathematical Problems in Engineering*, vol. 2012, 2012. Article ID 792057.

[18] Z. Chen, F. Huang, W. Sun, and W. Song, "An improved wave-variable based four-channel control design in bilateral teleoperation system for time-delay compensation," *IEEE Access*, vol. 6, pp. 12848–12857, 2018.

[19] Y. Yuan, Y. Wang, and L. Guo, "Force reflecting control for bilateral teleoperation system under time-varying delays," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 1162–1172, Feb 2019.

[20] D. Heck, A. Saccon, R. Beerens, and H. Nijmeijer, "Direct force-reflecting two-layer approach for passive bilateral teleoperation with time delays," *IEEE Transactions on Robotics*, vol. 34, pp. 194–206, Feb 2018.

[21] Z. Li, L. Ding, H. Gao, G. Duan, and C. Su, "Trilateral teleoperation of adaptive fuzzy force/motion control for nonlinear teleoperators with communication random delays," *IEEE Transactions on Fuzzy Systems*, vol. 21, pp. 610–624, Aug 2013.

[22] E. Nuño, I. Sarras, and L. Basañez, "An adaptive controller for bilateral teleoperators: Variable time-delays case," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 9341 – 9346, 2014. 19th IFAC World Congress.

[23] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812 – 823, 2010. Special Issue on Design and Control Methodologies in Telerobotics.

[24] D. Sun, F. Naghdy, and H. Du, "Neural network-based passivity control of teleoperation system under time-varying delays," *IEEE Transactions on Cybernetics*, vol. 47, pp. 1666–1680, July 2017.

[25] J. Sheng and M. Spong, "Model predictive control for bilateral teleoperation systems with time delays," in *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)*, vol. 4, pp. 1877–1880 Vol.4, 2004.

[26] L. Chan, Y. Liu, Q. Huang, and P. Wang, "Robust adaptive observer-based predictive control for a non-linear delayed bilateral teleoperation system," *IEEE Access*, vol. 10, pp. 52294–52305, 2022.

[27] J. I. Lipton, A. J. Fay, and D. Rus, "Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing," *IEEE Robotics and Automation Letters*, vol. 3, pp. 179–186, Jan 2018.

[28] X. Xu, A. Song, D. Ni, H. Li, P. Xiong, and C. Zhu, "Visual-haptic aid teleoperation based on 3-d environment modeling and updating," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 6419–6428, Oct 2016.

[29] L. Huijun and S. Aiguo, "Virtual-environment modeling and correction for force-reflecting teleoperation with time delay," *IEEE Transactions on Industrial Electronics*, vol. 54, pp. 1227–1233, April 2007.

[30] T. Abut and S. Soygüder, "Haptic industrial robot control and bilateral teleoperation by using a virtual visual interface," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, May 2018.

[31] V. Chawda and M. K. O'Malley, "Position synchronization in bilateral teleoperation under time-varying communication delays," *IEEE/ASME Transactions on Mechatronics*, vol. 20, pp. 245–253, Feb 2015.

[32] M. Shahbazi, S. F. Atashzar, M. Tavakoli, and R. V. Patel, "Position-force domain passivity of the human arm in telerobotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 23, pp. 552–562, April 2018.

[33] D. Babu Venkateswaran and Z. Qu, "Passivity-short bilateral teleoperation with communication delays," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1275–1281, Oct 2018.

[34] M. W. S. Atman, K. Noda, R. Funada, J. Yamauchi, T. Hatanaka, and M. Fujita, "On passivity-shortage of human operators for a class of semi-autonomous robotic swarms," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 21 – 27, 2019. 2nd IFAC Conference on Cyber-Physical and Human Systems.

[35] $\dot{I}$. Polat and C. W. Scherer, "Stability analysis for bilateral teleoperation: An IQC formulation," *IEEE Transactions on Robotics*, vol. 28, pp. 1294–1308, Dec 2012.

[36] C. A. Lopez Martínez, R. van de Molengraft, S. Weiland, and M. Steinbuch, "Switching robust control for bilateral teleoperation," *IEEE Transactions on Control Systems Technology*, vol. 24, pp. 172–188, Jan 2016.

[37] B. H. Jafari and M. W. Spong, "Passivity-based switching control in teleoperation systems with time-varying communication delay," in *2017 American Control Conference (ACC)*, pp. 5469–5475, May 2017.

[38] I. G. Polushin, A. Tayebi, and H. J. Marquez, "Control schemes for stable teleoperation with communication delay based on IOS small gain theorem," *Automatica*, vol. 42, no. 6, pp. 905 – 915, 2006.

[39] C. Hua and X. P. Liu, "Teleoperation over the internet with/without velocity signal," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp. 4–13, Jan 2011.

[40] D.-H. Zhai and Y. Xia, "Robust saturation-based control of bilateral teleoperation without velocity measurements," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 15, pp. 2582–2607, 2015.

[41] C. Hua, Y. Yang, and P. X. Liu, "Output-feedback adaptive control of networked teleoperation system with time-varying delay and bounded inputs," *IEEE/ASME Transactions on Mechatronics*, vol. 20, pp. 2009–2020, Oct 2015.

[42] C. Passenberg, A. Peer, and M. Buss, "Model-mediated teleoperation for multi-operator multi-robot systems," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4263–4268, Oct 2010.

[43] Z. Chen, Y.-J. Pan, and J. Gu, "Integrated adaptive robust control for multilateral teleoperation systems under arbitrary time delays," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 12, pp. 2708–2728, 2016.

[44] U. Ahmad and Y. Pan, "A time domain passivity approach for asymmetric multilateral teleoperation system," *IEEE Access*, vol. 6, pp. 519–531, 2018.

[45] Z. Chen, F. Huang, W. Song, and S. Zhu, "A novel wave-variable based time-delay compensated four-channel control design for multilateral teleoperation system," *IEEE Access*, vol. 6, pp. 25506–25516, 2018.

[46] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, 2017.

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

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

[49] C. L. P. Chen, G.-X. Wen, Y.-J. Liu, and F.-Y. Wang, "Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1217–1226, 2014.

[50] W. He, B. Xu, Q.-L. Han, and F. Qian, "Adaptive consensus control of linear multiagent systems with dynamic event-triggered strategies," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 2996–3008, 2020.

[51] D. Zhang, G. Feng, Y. Shi, and D. Srinivasan, "Physical safety and cyber security analysis of multi-agent systems: A survey of recent advances," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 319–333, 2021.

[52] Z. Feng and G. Hu, "Distributed secure average consensus for linear multi-agent systems under dos attacks," in *2017 American Control Conference (ACC)*, pp. 2261–2266, 2017.

[53] B. Chang, X. Mu, Z. Yang, and J. Fang, "Event-based secure consensus of muti-agent systems under asynchronous dos attacks," *Applied Mathematics and Computation*, vol. 401, p. 126120, 2021.

[54] Y. Yang, Y. Li, and D. Yue, "Event-trigger-based consensus secure control of linear multi-agent systems under dos attacks over multiple transmission channels," *Science China Information Sciences*, vol. 63, pp. 1–14, 2020.

[55] V. S. Dolk, P. Tesi, C. De Persis, and W. P. M. H. Heemels, "Event-triggered control systems under denial-of-service attacks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 93–105, 2017.

[56] T.-Y. Zhang and D. Ye, "Distributed event-triggered control for multi-agent systems under intermittently random denial-of-service attacks," *Information Sciences*, vol. 542, pp. 380–390, 2021.

[57] C. Peng and H. Sun, "Switching-like event-triggered control for networked control systems under malicious denial of service attacks," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3943–3949, 2020.

[58] N. Zhao, P. Shi, W. Xing, and J. Chambers, "Observer-based event-triggered approach for stochastic networked control systems under denial of service attacks," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 158–167, 2021.

[59] G. Wen, P. Wang, Y. Lv, G. Chen, and J. Zhou, "Secure consensus of multi-agent systems under denial-of-service attacks," *Asian Journal of Control*, vol. 25, no. 2, pp. 695–709, 2023.

[60] L. Zhao and G.-H. Yang, "Adaptive fault-tolerant control for nonlinear multi-agent systems with dos attacks," *Information Sciences*, vol. 526, pp. 39–53, 2020.

[61] T. Wang, J. Feng, J.-A. Wang, J. Zhang, and X. Wen, "Observer-based distributed event-triggered secure consensus of multi-agent system with dos attack," *IEEE Access*, vol. 11, pp. 34736–34745, 2023.

[62] Z. Zuo, X. Cao, Y. Wang, and W. Zhang, "Resilient consensus of multiagent systems against denial-of-service attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2664–2675, 2022.

[63] A. Amini, A. Asif, and A. Mohammadi, "A unified optimization for resilient dynamic event-triggering consensus under denial of service," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 2872–2884, 2022.

[64] A.-Y. Lu and G.-H. Yang, "Stability analysis for cyber-physical systems under denial-of-service attacks," *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5304–5313, 2021.

[65] C.-L. Zhang, G.-H. Yang, and A.-Y. Lu, "Resilient observer-based control for cyber-physical systems under denial-of-service attacks," *Information Sciences*, vol. 545, pp. 102–117, 2021.

[66] R.-Z. Chen, Y.-X. Li, and Z.-S. Hou, "Distributed model-free adaptive control for multi-agent systems with external disturbances and dos attacks," *Information Sciences*, vol. 613, pp. 309–323, 2022.

[67] Y.-S. Ma, W.-W. Che, C. Deng, and Z.-G. Wu, "Distributed model-free adaptive control for learning nonlinear mass under dos attacks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1146–1155, 2023.

[68] Z. Liu and L. Wang, "Leveraging network topology optimization to strengthen power grid resilience against cyber-physical attacks," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1552–1564, 2021.

[69] K. Jalilpoor, M. T. Ameli, S. Azad, and Z. Sayadi, "Resilient energy management incorporating energy storage system and network reconfiguration: A framework of cyber-physical system," *IET Generation, Transmission & Distribution*, vol. 17, no. 8, pp. 1734–1749, 2023.

[70] J. C. Willems, "Dissipative dynamical systems part i: General theory," *Archive for Rational Mechanics and Analysis*, vol. 45, pp. 321–351, Jan 1972.

[71] Z. Qu and M. A. Simaan, "Modularized design for cooperative control and plug-and-play operation of networked heterogeneous systems," *Automatica*, vol. 50, pp. 2405–2414, Sept. 2014.

[72] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice-Hall, third ed., 2002.

[73] Y. Joo, R. Harvey, and Z. Qu, "Cooperative control of heterogeneous multi-agent systems in a sampled-data setting," in *2016 IEEE 55th Conference on Decision and Control*, pp. 2683–2688, Dec 2016.

[74] R. Harvey and Z. Qu, "Cooperative control and networked operation of passivity-short systems," in *Control of Complex Systems* (K. G. Vamvoudakis and S. Jagannathan, eds.), pp. 499 – 518, Butterworth-Heinemann, 2016.

[75] Z. Qu and D. M. Dawson, *Robust Tracking Control of Robot Manipulators*. Piscataway, NJ, USA: IEEE Press, 1st ed., 1995.

[76] D. B. Venkateswaran and Z. Qu, "A passivity-shortage based control design for teleoperation with time-varying delays," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4070–4077, 2020.

[77] D. B. Venkateswaran and Z. Qu, "Distributed multilateral teleoperation framework using passivity-shortage," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 181–186, 2019. 8th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2019.

[78] W. Ni and D. Cheng, "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Systems and Control Letters*, vol. 59, no. 3, pp. 209 – 217, 2010.

[79] E. Nuño, "Bilateral teleoperation experiments: Scattering transformation and passive output synchronization re," pp. 12697–12702, 07 2008.

[80] T. Hatanaka, N. Chopra, M. Fujita, and M. Spong, *Passivity-Based Control and Estimation in Networked Robotics*. Springer Publishing Company, Incorporated, 2015.

[81] A. Mohammadi, M. Tavakoli, and A. Jazayeri, "Phansim: A simulink toolkit for the sensable phantom haptic devices," in *Proceedings of the 23rd CANCAM*, pp. 787–790, June 2011.

[82] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 624–637, Oct 1993.

[83] A. K. Wu, L. Tian, and Y.-Y. Liu, "Bridges in complex networks," *Phys. Rev. E*, vol. 97, p. 012307, Jan 2018.

[84] D. Naor, D. Gusfield, and C. Martel, "A fast algorithm for optimally increasing the edge-connectivity," in *31st Annual Symposium on Foundations of Computer Science*, pp. 698–707 vol.2, 1990.

[85] A. Gusrialdi, D. B. Venkateswaran, and Z. Qu, "Enhancing resilience in cooperative systems against cyber-attacks: A defense framework through adaptive network reconfiguration and digital twin," in *Latest Adaptive Control Systems* (D. P. Ioannou, ed.), ch. 0, Rijeka: IntechOpen, 2024.

[86] E. Latif and R. Parasuraman, "Dgorl: Distributed graph optimization based relative localization of multi-robot systems," in *Distributed Autonomous Robotic Systems* (J. Bourgeois, J. Paik, B. Piranda, J. Werfel, S. Hauert, A. Pierson, H. Hamann, T. L. Lam, F. Matsuno, N. Mehr, and A. Makhoul, eds.), (Cham), pp. 243–256, Springer Nature Switzerland, 2024.

[87] S. Battiston, G. Caldarelli, R. M. May, T. Roukny, and J. E. Stiglitz, "The price of complexity in financial networks," *Proceedings of the National Academy of Sciences*, vol. 113, no. 36, pp. 10031–10036, 2016.

[88] M. Maghenem, A. Loría, E. Nuño, and E. Panteley, "Consensus-based formation control of networked nonholonomic vehicles with delayed communications," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2242–2249, 2021.

[89] D. B. Venkateswaran and Z. Qu, "Resilient multi-agent systems against denial of service attacks via adaptively activatable network layers." Submitted in a Special Section: Resilient

and Safe Control in Multi-Agent Systems, in IEEE Open Journal of Control Systems (OJ-CSYS), 2024.

[90] D. B. Venkateswaran, Z. Qu, and A. Gusrialdi, "A distributed method for detecting critical edges and increasing edge connectivity in undirected networks." Submitted to the 63rd IEEE Conference on Decision and Control, 2024.

[91] D. B. Venkateswaran and Z. Qu, "Enhancing directed network robustness through critical edge detection and graph simplification." under preparation, 2024.