ANALYSIS OF LARGE-SCALE ASYNCHRONOUS SWITCHED DYNAMICAL SYSTEMS

A Dissertation

by

KOOKTAE LEE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Raktim Bhattacharya
Co-Chair of Committee,	Diego Donzis
Committee Members,	Aniruddha Datta
	P. R. Kumar
Head of Department,	Rodney D. W. Bowersox

August 2015

Major Subject: Aerospace Engineering

ABSTRACT

This dissertation addresses research problems related to the switched system as well as its application to large-scale asynchronous dynamical systems. For decades, this switched system has been widely studied in depth, owing to the broad applicability of the switched system framework. For example, the switched system can be adopted for modeling the dynamics of numerous systems including power systems, manufacturing systems, aerospace systems, networked control systems, etc. Despite considerable research works that have been developed during last several decades, there are still remaining yet important and unsolved problems for the switched systems. In the first part of this dissertation, new methods are developed for uncertainty propagation of stochastic switched systems in the presence of the state uncertainty, represented by probability density functions (PDFs). The main difficulty of this problem is that the number of PDF components in the state increases exponentially under the stochastic switching, incurring the curse of dimensionality. This dissertation provides a novel method that circumvents the issue regarding the curse of dimensionality. As an extension of this research, the new method for the switching synthesis is presented in the second part, to achieve the optimal performance of the switched system. This research is relevant to developing the switching synthesis on how to switch between different switching modes.

In the following chapters, some interesting applications that emerges as today's leading-edge technology in high-performance computing (HPC) will be introduced. Generally, the massive parallel computing entails idle process time in multi-core processors or distributed computing devices as up to 80% of total computation time, owing to the synchronization of the data. Thus, there is a trend toward relaxing

such a restriction on synchronization penalty to overcome this bottleneck problem. This dissertation presents asynchronous computing algorithms as a key solution to leverage the computing performance to the maximum capabilities. The price to pay for adopting the asynchronous computing algorithms is, however, unpredictability of the solution due to the randomness in the behavior of asynchrony. In this dissertation, the switched system is employed to model the characteristics of the asynchrony in parallel computing, enabling analysis of the asynchronous algorithm. Particularly, the analysis will be performed for massively parallel asynchronous numerical algorithms implemented on 1D heat equation and large-scale asynchronous distributed quadratic programming problems. As another case study, this switched system is also implemented on the stability analysis of large-scale distributed networked control systems (DNCS) having random communication delays. For these problems, the convergence or stability analysis is carried out by the switched system framework. One of major concerns when adopting the switched system framework for analysis of these systems is the scalability issues associated with extremely large switching mode numbers. Due to the massive parallelism or large-scale distributed nodes, the switching mode numbers are beyond counting, leading to the computational intractability. The proposed methods are developed targeting the settlement of this scalability issue, which inevitably takes place in adopting the switched system framework. Thus, the primary emphasis of this dissertation is placed on the mathematical development of computationally efficient tools, particularly for analysis of the large-scale asynchronous switched dynamical system, which has broad applications including massively parallel asynchronous numerical algorithms to solve ODE/PDE problems, distributed optimization problems, and large-scale DNCS with random communication delays.

DEDICATION

To my lovely wife Youjin and my dearest daughter Serene for their unconditional love and endless support they have shown all the way!

ACKNOWLEDGEMENTS

I cannot help mentioning those people who have had influence on me both intellectually and physically while pursuing my degree in Doctor of Philosophy. I admit that without their great support and help, I would not finish this long journey to obtain my Ph.D. degree. First of all, I am very grateful to Dr. Raktim Bhattacharya who has guided me during my Ph.D. as a research and academic advisor. His brilliant ideas have always inspired me whenever I could not find a way to look at the problems in broader vision. Occasionally, I was really into the problem itself rather than finding out how to implement the proposed research methods into wider context. Dr. Bhattacharya's great inspirations have always illuminated me while I couldn't see the forest for the trees. Also, I appreciate his great support, which made me purely concentrate on the research problems as a graduate student.

I also thank to Dr. Aniruddha Datta, Dr. Diego Donzis, and Dr. P. R. Kumar who are my committee members. I took Dr. Datta's class in 2011, which helped me to understand the basic concepts and fundamentals of adaptive control. I had research collaboration with Dr. Donzis, to analyze the behavior of asynchrony in massively parallel numerical algorithms. It was always my great pleasure to have discussions on the analysis of asynchronous parallel computing with Dr. Donzis and this research collaboration led to motivation for my doctoral dissertation. I also thank to Dr. Kumar whose deep insight on research made me better understand the problems in different perspective. I appreciate Dr. Kumar's considerable comments on my research during my preliminary examination.

In summer 2014, I visited University of Notre Dame as a visiting scholar, having research collaboration with Dr. Vijay Gupta. Dr. Gupta's exceptional ideas and

advice helped me to solve very challenging research problems, and this research collaboration made me grow up one more step academically. I never forget Dr. Gupta's kindness and great hospitality while I was staying at Notre Dame.

I also would like to express my thankful mind to Karen, Michelle, and Rose in the Aerospace department. Karen always helped me whenever I got in trouble due to the lack of information. Michelle assisted me to process the work as a research assistant. Especially, I really thank to Rose who facilitated my travel for the conference throughout my Ph.D.

Last but never least, I heartily appreciate my family's unlimited support and unconditional love they always showed at my side. My wife, Youjin, has encouraged me and made me feel alive. Her vivid characteristics has made me cheer up whenever I felt stressed or depressed due to a long road to my degree, which seemed to be endless. My dearest daughter, Serene, has always been meaning and purpose of my life. Her smile has lifted my spirit, and has brightened me. Without my family's enormous amount of love and trust, I would not end up with Ph.D. I dedicate this work to them.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Overview

Typically, the *switched system* (or jump linear system, interchangeably) refers to the dynamical system consists of a family of subsystem dynamics and a switching logic that orchestrates the switching between subsystem dynamics [10], [17], [19], [31], [66], [68], [34], [22], [62], [63], [64]. Since this switched system has broad applications ranging from power systems, manufacturing systems, aerospace systems to even networked control systems, there have been considerable research works on the analysis of the switched system during past decades. One of the enthralling aspects of this switched system is that the system can be unstable by the switching even in the case that all subsystem dynamics are known to be stable. Thus, it has attracted wide research interests regarding the stability analysis of the switched system. For example, Kozin [55] surveyed basic ideas for the stability of stochastic systems and defined various concepts of stochastic stability, which has served as a stepping stone in the stability analysis of stochastic switched systems. Feng *et.al.* [34] showed equivalence of three different notions of the stability (e.g., second moment stability, stochastic stability and exponential mean square stability) for the switched systems by inquiring about the stochastic properties of the transition matrix. In [52], Ji *et.al.* studied the stability of the discrete-time jump linear systems in the mean square sense, and introduced less conservative stability notion – almost sure stability.

Later, the classification of the stochastic switched system is more concretized by the embodiment of the switching process such as independent, identically distributed (i.i.d.) process, Markovian process, and semi-Markovian process. Among different notions for stochastic switched systems, Markov jump linear system struc-

ture has been widely employed due to its practicality for formulating randomness. For instance, the Markov jump linear system framework can be utilized for modeling communication delays [97] or packet losses [99] in networked systems, abrupt environmental disturbances or changes in subsystems [74], systems with parametric uncertainties [105], etc. Since the system stability is of concern under Markovian switching, it has led to a variety of research areas [105], [23], [86], [9]. For the general non-Markovian stochastic jump linear systems, however, only few results [34], [74] can be found. In most cases, the mean square stability conditions are obtained from the Lyapunov's method, where finding Lyapunov function satisfying certain conditions is sometimes troublesome. Moreover, most literature has focused on the stability itself, meaning that it has remained as an unexplored area to investigate the performance of the switched system in transient time, which requires uncertainty propagation under the switching process.

This dissertation addresses a new method that enables uncertainty propagation for the switched system of which state is represented by the probability density function (PDF). Particularly, this research will investigate uncertainty propagation for any arbitrary stochastic switching process, including Markovian jump process. The major difficulty in uncertainty propagation problem is that the number of component PDFs grows exponentially under the stochastic switching, leading to the curse of dimensionality problems. To avoid this dimensionality issue, this study will develop a novel approach, "split-and-merge" algorithm, which enables the performance and robustness analysis for the stochastic switched system in a computationally efficient manner. Uncertainty is quantified via the Wasserstein metric to measure the distance between PDFs. Through the convergence of the state PDF, system stability is guaranteed in the mean square sense, which will be proven with a particular choice of the reference PDF as Dirac distribution. Thus, the proposed methods can be

used to measure the system performance as well as robustness with respect to the stochastic switching both in transient and steady-state time. Then, a new approach for the optimal switching synthesis will be presented for general switched systems. In conjunction with the receding horizon control framework, this optimal switching synthesis guarantees the optimal performance of the switched systems, which provides the information regarding how and when to switch. The proposed optimal switching synthesis method can be adopted to the controller switching for optimal performance of the system.

In the following chapters, this dissertation introduces some interesting case studies as applications for the large-scale asynchronous switched dynamical systems [62], [60], [61]. These case studies include massively parallel asynchronous numerical algorithm, large-scale asynchronous distribute quadratic programming problems, and large-scale distributed networked control systems (DNCSs) with random communication delays. The first two case studies are mostly relevant to the high-performance computing to achieve the maximum computing capability in multi-core processors or distributed computing devices. With a rapid advancement of computing technology in distributed and parallel computing, one can handle complicated and computationally intensive applications. These applications range from science and engineering problems to even big data and information processing, for which solutions can be approximately obtained by numerical methods. In solving such large-scale problems, the purpose of implementing distributed and parallel computing is glaringly obvious – to speedup the computation. However, distributed and parallel computing entails synchronization at each iteration step due to the interdependency of the data, which may result in a large amount of waiting time, particularly for extreme-scale parallel computation. In fact, it has been reported [13] that this synchronization latency can reach up to 50% or even more in the total computation time. Therefore, the syn-

chronization latency may become dominant as compared to the pure computation time, and thus it may severely degrade the performance of distributed and parallel computing. To mitigate this restriction on synchronization penalty, this dissertation investigates asynchronous parallel computing algorithm, where asynchronous algorithm is a benefit as well as a key point to solving massively computation intensive problems. The major concern when adopting the asynchronous computing algorithm is that it is not completely and thoroughly revealed yet what is the effect of asynchrony on the solution. Due to the randomness of asynchrony, the solution obtained from asynchronous computing algorithms becomes unpredictable, incurring the tradeoff between speedup and accuracy. Motivated by the utilization of the stochastic switched system that is broadly used in the analysis of systems with random delays, this dissertation employs the switched system framework to characterize the behavior of such an asynchrony. Thus, the effect of asynchrony is analyzed by means of this switched system framework. Particularly, this dissertation aims at developing new mathematical tools based on the switched system framework for analysis of asynchronous parallel computing algorithms, implemented on asynchronous parallel numerical algorithms to solve ODE/PDE problems and asynchronous distributed quadratic programming problems. As another example, the stability of large-scale DNCSs having random communication delays will be discussed. For this system, each agent has communication with its neighbors, where the communication delays may take place as described in Fig. 1.1. In such cases, system stability is one of the major concerns because the communication delays may incur system instability. To describe the randomness of communication delays in DNCS, the switched system framework will be employed again. Then, it will be shown that the dynamics of large-scale DNCS with random communication delays can be successfully modeled by the switched system. Therefore, the stability analysis will be performed under

Figure 1.1: The schematic of DNCS with random communication delays.

this switched system framework.

Note that we are particularly interested in large-scale problems of which network map, for example, appears as in Fig. 1.2. In this type of excessively large-scale network connection, adopting the switched system framework for the purpose of system analysis will cause scalability problems associated with an extremely large amount of switching mode numbers. Therefore, the primal emphasis of dissertation lies in the development of computationally efficient tools for the stability or convergence analysis of large-scale asynchronous switched dynamical systems. Further explanations regarding the proposed research approaches with chapter organizations are described in the following section.

1.2 Research Approaches with Chapter Organization

The chapter organization for this dissertation is described in Fig. 1.3. In chapter 2, a new method for performance analysis of the stochastic switched system will be presented, followed by the optimal switching synthesis method in chapter 3. These two chapters deal with analysis and synthesis for general switched systems. Then, the remaining three chapters introduce some case studies for large-scale asynchronous switched dynamical systems. Especially, chapter 4 and 5 are related to the

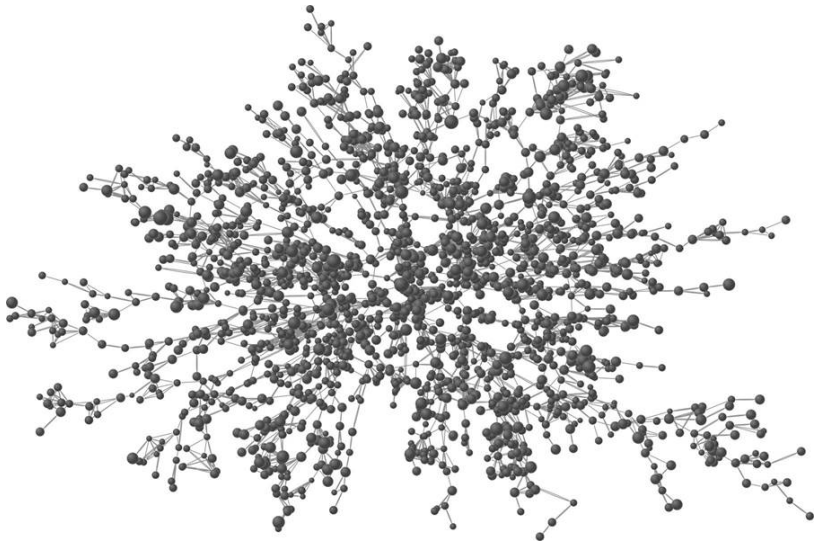


Figure 1.2: Example of large-scale network map.

high-performance computing with an implementation of the asynchronous computing algorithms. In these chapters, massively asynchronous parallel numerical algorithms and asynchronous distributed quadratic programming problems will be analyzed in the switched system framework. Finally, analysis of large-scale DNCS with random communication delays will be introduced in chapter 6, and the stability condition will be established via the switched system framework.

More details about problem descriptions and research approaches are addressed below.

• **Performance Analysis of Switched Systems with Stochastic Jumping Parameters:**

This research focuses on the performance and the robustness analysis of stochastic jump linear systems. In the presence of stochastic jumps, state variables evolve as random process, with associated time varying probability density functions. Consequently, system analysis is performed at the density level and a proper metric is

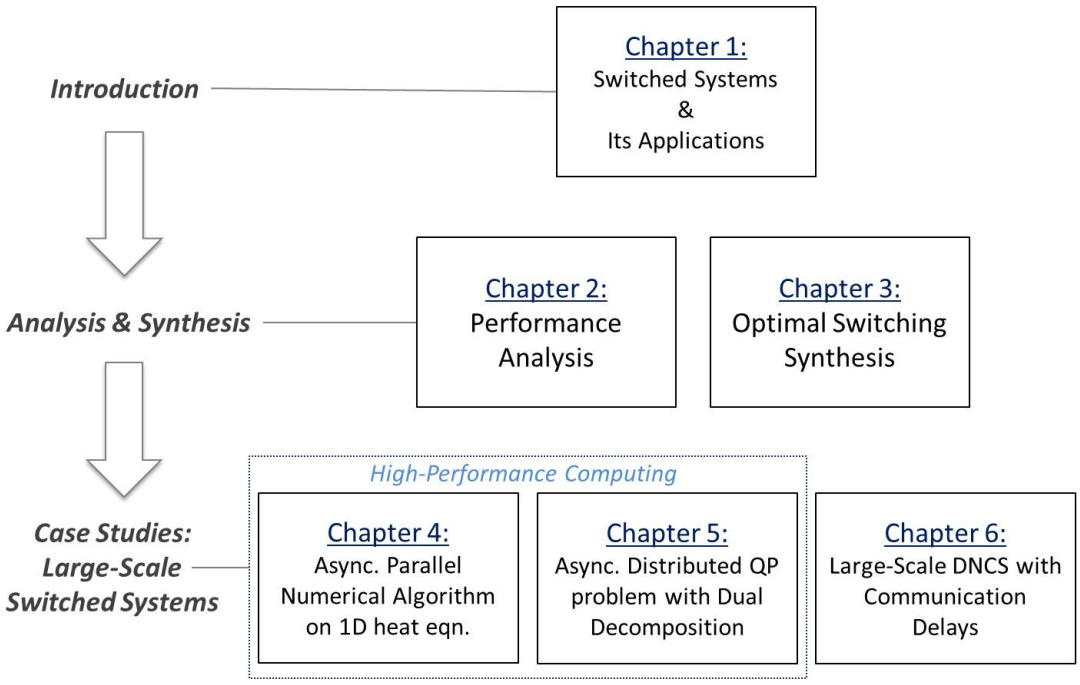


Figure 1.3: Chapter organization and flows.

necessary to quantify the system performance. In this research, Wasserstein metric between probability density functions is employed to develop new results for the performance analysis of stochastic jump linear systems. Both transient and steady-state performance of the systems, with given initial state uncertainties, can be analyzed in this framework. Also, we prove that the convergence of the Wasserstein metric implies the mean square stability. In addition, we present a novel "Split-and-Merge" algorithm for propagation of state uncertainty for such systems. Overall, this study provides a unifying framework for the performance and the robustness analysis of general stochastic jump linear systems, and not necessarily Markovian that is commonly assumed.

• **Optimal Switching Synthesis for Switched Systems with Gaussian Initial State Uncertainty:**

This study provides a method to design an optimal switching sequence for jump linear systems with given Gaussian initial state uncertainty. In the practical perspective, the initial state contains some uncertainties that come from measurement errors or sensor inaccuracies and we assume that the type of this uncertainty has the form of Gaussian distribution. In order to cope with Gaussian initial state uncertainty and to measure the system performance, Wasserstein metric that defines the distance between probability density functions is used. Combining with the receding horizon framework, an optimal switching sequence for jump linear systems can be obtained by minimizing the objective function that is expressed in terms of Wasserstein distance. The proposed optimal switching synthesis also guarantees the mean square stability for jump linear systems.

• **Analysis of Massively Parallel Asynchronous Numerical Algorithms:**

In the near future, massively parallel computing systems will be necessary to solve computation intensive applications. The key bottleneck in massively parallel implementation of numerical algorithms is the synchronization of data across processing elements (PEs) after each iteration, which results in significant idle time. Thus, there is a trend towards relaxing the synchronization and adopting an asynchronous model of computation to reduce idle time. However, it is not clear what is the effect of this relaxation on the stability and accuracy of the numerical algorithm.

In this study we present a new framework to analyze such algorithms. We treat the computation in each PE as a dynamical system and model the asynchrony as stochastic switching. The overall system is then analyzed as a switched dynamical system. However, modeling of massively parallel numerical algorithms as switched dynamical systems results in a very large number of modes, which makes current analysis tools available for such systems computationally intractable. We develop new techniques that circumvent this scalability issue. The framework is presented

on a one-dimensional heat equation and the proposed analysis framework is verified by solving the partial differential equation (PDE) in a nVIDIA Tesla™ GPU machine, with asynchronous communication between cores.

• **On the Convergence of Asynchronous Distributed Quadratic Programming via Dual Decomposition:**

In this research, we analyze the convergence as well as the rate of convergence of asynchronous distributed quadratic programming (QP) with dual decomposition technique. In general, distributed optimization requires synchronization of data at each iteration step due to the interdependency of data. This synchronization latency may incur a large amount of waiting time caused by an idle process during computation. We aim to attack this synchronization penalty in distributed QP problems by implementing asynchronous update of dual variable. The price to pay for adopting asynchronous computing algorithms is unpredictability of the solution, resulting in a tradeoff between speedup and accuracy. Thus, the convergence to an optimal solution is not guaranteed owing to the stochastic behavior of asynchrony. In this study, we employ the switched system framework as an analysis tool to investigate the convergence of asynchronous distributed QP. This switched system will facilitate analysis on asynchronous distributed QP with dual decomposition, providing necessary and sufficient conditions for the mean square convergence. Also, we provide an analytic expression for the rate of convergence through the switched system, which enables performance analysis of asynchronous algorithms compared to the synchronous case. To verify the validity of the proposed methods, numerical examples are presented with an implementation of asynchronous parallel QP using OpenMP.

• **Stability of Large-scale Distributed Networked Control Systems with Random Communication Delays:**

In this research, we consider the stability analysis of large-scale distributed net-

worked control systems with random communication delays. The stability analysis is performed in the switched system framework, particularly as the Markov jump linear system. There have been considerable research on stability analysis of the Markov jump systems. However, these methods are not applicable to large-scale systems because large numbers of subsystems result in extremely large number of switching modes. To circumvent this scalability issue, we propose a new reduced mode model for stability analysis, which is computationally scalable. We also consider the case in which the transition probabilities for the Markov jump process contain uncertainties. We provide a new method that estimates bounds for uncertain Markov transition probability matrix to guarantee the system stability. Numerical example verifies the computational efficiency of the proposed methods.

1.3 Key Contribution and Research Impacts

This dissertation will mainly focus on the analysis of stochastic switched systems and its applications to large-scale asynchronous switched dynamical systems. Key contributions of this dissertation include new techniques for:

1) A novel method for uncertainty propagation of the switched system in conjunction with nonlinear, non-Gaussian state PDFs (Chapter 2)

2) Optimal switching synthesis for the switched system with Gaussian state uncertainty (Chapter 3)

3) Application I: Analysis of massively parallel asynchronous computing algorithms implemented on parallel numerical algorithms (Chapter 4)

4) Application II: Analysis of large-scale distributed quadratic programming (Chapter 5)

5) Application III: Stability analysis of large-scale distributed networked control systems with random communication delays (Chapter 6)

The above lists 1) and 2) are developed for general switched systems. On the other hands, lists from 3) to 5) are targeting the analysis of asynchronous algorithms and large-scale systems with random communication delays by employing the switched system framework.

Regarding the uncertainty propagation for the stochastic switched systems, it is expected that the proposed method will be utilized to measure the system uncertainty, caused by both state uncertainty and stochastic jump process. The major impact of this research is that it enables the uncertainty quantification both transient and steady-state time, whereas previous literature only focused on the stability in the steady state. Thus, the performance and robustness analysis can be carried out by the proposed methods while avoiding the aforementioned curse of dimensionality issue. Also, a new approach is developed for the optimal switching synthesis problem. This research provides information on how to switch for the optimal performance of the switched system considering the state uncertainty in the form of Gaussian noise. This will benefit the synthesis problems including controller switching synthesis for the optimal performance of the system.

For the analysis of asynchronous computing algorithms, it is expected to bring transformative potential in scientific computing at extreme scale. At these scales, the idle time for synchronization across multi-core or distributed devices may incur a huge amount of waiting time, which can be avoided by the asynchronous algorithms. This capability is highly relevant to the scientific computing community involved in developing mathematical methods and algorithms to accurately and efficiently describe the behavior of complex systems. Ultimately, this will enable both accurate

and efficient analysis on massively parallel asynchronous computing algorithms that will be necessary as well as ubiquitous in the near future to further understanding the complicated and critical scientific inquiry.

2. PERFORMANCE ANALYSIS OF SWITCHED SYSTEMS WITH STOCHASTIC JUMPING PARAMETERS

2.1 Introduction

A jump linear system is defined as a dynamical system constructed with a set of linear subsystem dynamics and a switching logic that conduct a switching between linear subsystems. Over decades, a jump linear system has attracted a wide range of researches due to its practical implementations. For instance, jump linear systems are used for power systems, manufacturing systems, aerospace systems, networked control systems, etc.

In general, a jump linear system can be divided into two different categories depending on the switching logic. One branch is a deterministic jump linear system where the jump process is deterministically driven by a certain switching logic. The utilization of such systems stems from plant stabilization [75], adaptive control [76], system performance [68], and resource-constrained scheduling [8]. In most cases, the system stability has been one of the major issues to investigate since even stable subsystems make the system unstable by the switching. Hence, numerous results have been established for the stability analysis and the recent literature regarding the stability of deterministic jump linear systems can be found in [68]. In [68], necessary and sufficient conditions for the system stability are shown via a finite tuple, satisfying a certain condition.

Unlike the deterministic jump linear system, a stochastic jump linear system (SJLS), which is another category of jump linear systems, refers to systems with a stochastic switching process. This type of jump linear systems is commonly used to represent the randomness in the switching such as communication delays or packet

losses in the networked control systems [47], [97]. In [47], the networked control system with packet losses was modeled as an asynchronous dynamical system incorporating both discrete and continuous dynamics, and its stability was analysed through Lyapunov techniques. Since then, this problem has been formulated in a more general setting by representing the various aspects of communication uncertainties as Markov chains [103], [18], [98], [99], [73]. Stability analysis in the presence of such uncertainty, has been performed in the Markov jump linear systems (MJLSs) framework [97], [104], [96], [53], [58], [106]. Especially, [106] analysed the stability of MJLS without requiring any knowledge of the structures in partially unknown Markov transition probabilities. Further, the stochastic stability for a class of nonlinear stochastic systems with semi-Markovian jump parameters is introduced in [51], [65]. Most previous literatures, however, have only dealt with steady-state analysis in terms of the system stability.

Beyond the current literature, this chapter has a key contribution for the analysis of a SJLS as follows. Based on the theory of optimal transport [94], we propose new probabilistic tools for analysing the performance and the robustness of SJLSs. Compared to the current literatures that only guarantees asymptotic performance with a deterministic arbitrary initial state condition, our contribution is to develop a unifying framework enabling both transient and asymptotic performance analysis with uncertain initial state conditions. The main difficulty dealing with analysis of SJLSs is that the system trajectories differ from every run due to the random switching. Moreover, the system state for SJLSs becomes random variables with a corresponding probability density function (PDF) even with a deterministic initial state condition. Therefore, we need to adopt a proper metric to measure the performance and the robustness of SJLSs in the distributional sense. In this chapter, the Wasserstein metric that enables uncertainty quantification by evaluating a distance

between PDFs is employed to measure the performance of SJLSs. We also prove that the convergence of this metric implies the mean square stability. To sum up, this chapter provides the performance and the robustness analysis tools for SJLSs with given initial state uncertainties in the absence of any restriction on the underlying jump processes.

The remainder of this chapter is organized as follows. In Section 2.2, we provide a brief review of the preliminaries. Section 2.3 deals with the performance and the robustness analysis of SJLSs and introduces computationally efficient tools for that purpose. Numerical examples are provided in Section 2.4, to illustrate the performance and the robustness analysis results developed in this work. Section 2.5 concludes the chapter.

Notation: The set of real and natural numbers are denoted by $\mathbb{R}$ and $\mathbb{N}$, respectively. Further, $\mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}$. The symbols $\mathrm{tr}\,(\cdot)$, $\otimes$, and $vec\,(\cdot)$ denote the trace of a square matrix, Kronecker product, and vectorization operators, respectively. The abbreviation m.s. stands for the convergence in mean-square sense. The notations $\mathbb{P}(\cdot)$ and $X \sim \rho\,(x)$ denote the probability and the random variable X with PDF $\rho\,(x)$, respectively. The symbol $\mathcal{N}\,(\mu, \Sigma)$ is used to denote the PDF of a Gaussian random vector with mean μ and covariance Σ.

2.2 Preliminaries

Consider a discrete-time jump linear system as follows.

$$x(k+1) = A_{\sigma_k} x(k), \quad k \in \mathbb{N}_0, \tag{2.1}$$

where k is a discrete-time index, $x(k)$ is the state vector, and A_{σ_k} denotes the system matrices. $\sigma_k \in \mathcal{M} \triangleq \{1, 2, \ldots, m\}$ stands for the stochastic jump process, governing the switching among m different modes of (2.1). In this chapter, we will consider

general stochastic jump processes σ_k, and hence σ_k can be any arbitrary random process. Then, the resulting dynamics becomes a SJLS as defined next.

Definition 2.1 *(**Stochastic jump linear system**) Tuple of the form* $(\{\pi(k)\}_{k=1}^{\infty}, \{A_1, \ldots, A_m\})$ *is termed as a SJLS, provided the mode dynamics are given by (2.1);* $\pi(k)$ *denotes the occupation probability vector at time* k *for the prescribed stochastic process* σ_k*.*

Remark 2.1 *A SJLS, as defined above, is a collection of modal vector fields and a sequence of mode-occupation probability vectors. If the jump processes* σ_k *is deterministic, then at each time,* $\pi(k)$ *will have integral co-ordinates (single 1 and remaining* $m-1$ *zeroes), resulting in a* deterministic switching sequence*. If, however,* σ_k *is stochastic jump processes, then* $\pi(k)$ *will contain proper fractional co-ordinates, resulting in a* randomized switching sequence *where at each time, exactly one out of* m *modes will be chosen according to probability* $\pi(k)$*. Thus, starting from a deterministic initial condition, each execution of the SJLS may result in different switching sequences corresponding to random sample paths of* σ_k *over* $\mathcal{M}$*. Every realization of these random switching sequences results in a trajectory realization on the state space, and hence repeated the SJLS executions, even with a fixed initial condition, yields a spatio-temporal evolution of joint state PDF* $\rho(x(k))$*.*

Based on the structure that governs the temporal evolution of $\pi(k)$, SJLSs can be categorized into several subsets according to inherent jump processes as follows.

1) i.i.d. jump process:

   A SJLS switching sequence is called stationary, if the occupation probability vector $\pi(k)$ remains stationary in time. In particular, a stationary *deterministic* switching sequence implies execution of a single mode (no switching). A

stationary randomized switching sequence implies i.i.d. jump process.

2) Markov jump process:

Consider a discrete-time discrete state Markov chain with mode transition probabilities given by

$$p_{ij} = \mathbb{P}\left(\sigma_{k+1} = j \mid \sigma_k = i\right),$$

where $p_{ij} \geq 0$, $\forall i, j \in \mathcal{M}$. Hence, for $k \geq 0$, the probability distribution $\pi\left(k\right) \in \mathbb{R}^m$ of the modes of (2.1), is governed by

$$\pi(k+1) = \pi(k)P, \quad \pi(0) = [\pi_1(0) \ \cdots \ \pi_m(0)],$$

where the *transition probability matrix* $P \in \mathbb{R}^{m\times m}$ is a right stochastic matrix with row sum $\sum_{j=1}^{m} p_{ij} = 1$, $\forall i \in \mathcal{M}$.

3) semi-Markov jump process:

For a homogeneous and discrete-time semi-Markov chain, semi-Markov kernel q is defined by

$$q_{ij}(k) = \mathbb{P}(\sigma_{n+1} = j, X_{n+1} = k | \sigma_n = i),$$

where X_n denotes the sojourn time in state $\sigma_n = i$. Note that the transition probability p_{ij} in Markov chain can be expressed in terms of the semi-Markov kernel by $p_{ij} = \sum_{k=0}^{\infty} q_{ij}(k)$.

2.3 Performance and Robustness Analysis using Wasserstein Metric

Uncertainties in a SJLS appear at the execution level due to random switching sequence. Additional uncertainties may stem from imprecise setting of initial conditions and parameter values. These uncertainties manifest as the evolution of the state PDF $\rho(x(k))$. Thus, a natural way to quantify the uncertainty for the performance of a SJLS, is to compute the "distance" of the instantaneous state PDF from a reference measure. In particular, if we fix the reference PDF as Dirac delta function at the origin, denoted as $\delta(x)$, then the time-history of this "distance" would reveal the rate of convergence (divergence) for the stable (unstable) SJLS in the distributional sense.

For meaningful inference, the notion of "distance" must define a metric, and should be computationally tractable. The choice of the metric is very important as it must be able to highlight properties of density functions that are important from a dynamical system point of view. We propose that the shape of the density functions characterizes the dynamics of the system. Regions of high probability density correspond to high likelihood of finding the state there, which corresponds to higher concentration of trajectories. Higher concentration occurs in regions with low time scale dynamics or time invariance. For example, for a stable system, all trajectories accumulate at the origin and the corresponding PDF is the Dirac delta function at the origin. Similarly, low concentration areas indicate fast-scale dynamics or instability, and the corresponding steady-state density function is zero in the unstable manifold. Therefore, behaviors of two dynamical systems are identical in the distribution sense if their state PDFs have identical shapes. In order to properly capture the above aspects in dynamical systems, we adopt Wasserstein distance and details are introduced in the following subsection.

2.3.1 Wasserstein Distance

Definition 2.2 *(**Wasserstein distance**) Consider the vectors $x_1, x_2 \in \mathbb{R}^n$. Let $\mathcal{P}_2(\varsigma_1, \varsigma_2)$ denote the collection of all probability measures ς supported on the product space $\mathbb{R}^{2n}$, having finite second moment, with first marginal ς_1 and second marginal ς_2. Then the Wasserstein distance of order 2, denoted as $\mathcal{W}$, between two probability measures ς_1, ς_2, is defined as*

$$\mathcal{W}(\varsigma_1, \varsigma_2) \triangleq \left(\inf_{\varsigma \in \mathcal{P}_2(\varsigma_1, \varsigma_2)} \int_{\mathbb{R}^{2n}} \| x_1 - x_2 \|^2_{\ell_2(\mathbb{R}^n)} \; d\varsigma(x_1, x_2) \right)^{\frac{1}{2}}. \tag{2.2}$$

Remark 2.2 *Intuitively, Wasserstein distance equals the* least amount of work *needed to morph one distributional shape to the other, and can be interpreted as the cost for Monge-Kantorovich optimal transportation plan [93]. Further, one can prove (p. 208, [93]) that $\mathcal{W}$ defines a metric on the manifold of PDFs.*

Next, we present new results for system stability in terms of $\mathcal{W}$ and simplifications in its computation.

Proposition 2.1 *If we fix Dirac distribution as a reference measure, then distributional convergence in Wasserstein metric is* necessary and sufficient *for convergence in m.s. sense.*

Proof Consider a sequence of n-dimensional joint PDFs $\{\rho_j(x)\}_{j=1}^{\infty}$, that converges to $\delta(x)$ in distribution, i.e., $\lim_{j \to \infty} \mathcal{W}(\rho_j(x), \delta(x)) = 0$. From (2.2), we have

$$\begin{aligned} \mathcal{W}^2(\rho_j(x), \delta(x)) &= \inf_{\varsigma \in \mathcal{P}_2(\rho_j(x), \delta(x))} \mathbb{E}\left[\| X_j - 0 \|^2_{\ell_2(\mathbb{R}^n)} \right] \\ &= \mathbb{E}\left[\| X_j \|^2_{\ell_2(\mathbb{R}^n)} \right], \end{aligned} \tag{2.3}$$

where the random variable $X_j \sim \rho_j(x)$. The last equality follows from the fact that $\mathcal{P}_2(\rho_j(x), \delta(x)) = \{\rho_j(x)\}\ \forall\, j$, thus infimum is obviated.

From (2.3), $\lim_{j\to\infty} \mathcal{W}(\rho_j(x), \delta(x)) = 0 \Rightarrow \lim_{j\to\infty} \mathbb{E}\left[\| X_j \|^2_{\ell_2}\right] = 0$, establishing distributional convergence to $\delta(x) \Rightarrow$ m.s. convergence. Conversely, m.s. convergence $\Rightarrow$ distributional convergence, is well-known [43] and unlike the other direction, holds for arbitrary reference measure. □

Proposition 2.2 *($\mathcal{W}$ **between multivariate Gaussians** [40]) The Wasserstein distance between two multivariate Gaussians supported on $\mathbb{R}^n$, with respective joint PDFs $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$, is given by*

$$\mathcal{W}(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) = \sqrt{\| \mu_1 - \mu_2 \|^2_{\ell_2(\mathbb{R}^n)} + tr\left(\Sigma_1 + \Sigma_2 - 2\left[\sqrt{\Sigma_1}\Sigma_2\sqrt{\Sigma_1}\right]^{\frac{1}{2}}\right)}. \tag{2.4}$$

Corollary 2.1 *($\mathcal{W}$ **between Gaussian and Dirac PDF**) Since we can write $\delta(x) = \lim_{\mu,\Sigma\to 0} \mathcal{N}(\mu, \Sigma)$ (see e.g., p. 160-161, [46]), it follows from (2.4) that*

$$\mathcal{W}(\mathcal{N}(\mu, \Sigma), \delta(x)) = \sqrt{\| \mu \|^2_{\ell_2(\mathbb{R}^n)} + tr(\Sigma)}. \tag{2.5}$$

2.3.2 Performance and Robustness Analysis for SJLSs

The performance and the robustness analysis problem for SJLSs is stated as follows: given a SJLS $(\{\pi(k)\}_{k=1}^{\infty}, \{A_1, \ldots, A_m\})$, compute and analyse the performance history, quantified by $W(k) \triangleq \mathcal{W}(\rho(x(k)), \delta(x))$. Comparison of $W(k)$ of uncertain systems with that of a nominal system, quantifies the degradation in system performance due to system uncertainty.

2.3.2.1 Uncertainty propagation in SJLSs

The key difficulty here is the propagation of state PDFs under the stochastic switching and we present a new algorithm for such computations.

Proposition 2.3 *Given m absolutely continuous random variables $X_1, \ldots, X_m$, with respective cumulative distribution functions (CDFs) $F_i(x)$, and PDFs $\rho_i(x)$, $\forall i \in \mathcal{M}$. Let $X \triangleq X_i$, with probability $\alpha_i \in [0,1]$, $\sum_{i=1}^{m} \alpha_i = 1$. Then, the CDF and the PDF of X are given by*

$$F(x) = \sum_{i=1}^{m} \alpha_i F_i(x), \quad \rho(x) = \sum_{i=1}^{m} \alpha_i \rho_i(x). \tag{2.6}$$

Proof

$$\begin{aligned} F(x) &\triangleq \mathbb{P}(X \leq x) \\ &= \sum_{i=1}^{m} \mathbb{P}(X = X_i)\, \mathbb{P}(X_i \leq x) \\ &= \sum_{i=1}^{m} \alpha_i F_i(x), \end{aligned}$$

where we have used the law of total probability. Since each X_i and hence X, is absolutely continuous, we have $\rho(x) = \sum_{i=1}^{m} \alpha_i \rho_i(x)$. □

Note that any continuous PDF can be approximated by a Gaussian mixture PDF in weak sense [83, 91]. Therefore, we assume the initial PDF for the SJLS to be m_0 components mixture of Gaussian (MoG), given by

$$\rho_0 = \sum_{j_0=1}^{m_0} \alpha_{j_0} \mathcal{N}(\mu_{j_0}, \Sigma_{j_0}), \quad \sum_{j_0=1}^{m_0} \alpha_{j_0} = 1.$$

Then, we have the following results.

Theorem 2.1 *(**A SJLS preserves MoG**) Consider a SJLS* $(\{\pi(k)\}_{k=1}^{\infty}, \{A_1, \ldots, A_m\})$ *with initial PDF* $\rho_0 = \sum_{j_0=1}^{m_0} \alpha_{j_0} \mathcal{N}(\mu_{j_0}, \Sigma_{j_0})$. *Then the state PDF at time* k, *denoted by* $\rho(x(k))$, *is given by*

$$\rho(x(k)) = \sum_{j_k=1}^{m} \sum_{j_{k-1}=1}^{m} \ldots \sum_{j_1=1}^{m} \sum_{j_0=1}^{m_0} \left(\prod_{r=1}^{k} \pi_{j_r}(r) \right) \alpha_{j_0} \mathcal{N}(\mu_{j_k}, \Sigma_{j_k}), \tag{2.7}$$

where

$$\begin{aligned} \mu_{j_k} &= A^*_{j_k} \mu_{j_0}, \\ \Sigma_{j_k} &= A^*_{j_k} \Sigma_{j_0} A^{*\top}_{j_k}, \\ A^*_{j_k} &\triangleq \prod_{r=k}^{1} A_{j_r} = A_{j_k} A_{j_{k-1}} \ldots A_{j_2} A_{j_1}. \end{aligned}$$

Proof Starting from ρ_0 at $k = 0$, the modal PDF at time $k = 1$, is given by

$$\rho_{j_1}(x(1)) = \sum_{j_0=1}^{m_0} \alpha_{j_0} \mathcal{N}(\mu_{j_1}, \Sigma_{j_1}), \tag{2.8}$$

where $j_1 = 1, \ldots, m$, $\mu_{j_1} = A_{j_1} \mu_{j_0}$, and $\Sigma_{j_1} = A_{j_1} \Sigma_{j_0} A^{\top}_{j_1}$, which follows from the fact that linear transformation of an MoG is an equal component MoG with linearly transformed component means and congruently transformed component covariances (see Theorem 6 and Corollary 7 in [2]). From Proposition 2.3, it follows that the state PDF at $k = 1$, is

$$\rho(x(1)) = \sum_{j_1=1}^{m} \sum_{j_0=1}^{m_0} \pi_{j_1}(1) \alpha_{j_0} \mathcal{N}(\mu_{j_1}, \Sigma_{j_1}), \tag{2.9}$$

where $\pi_{j_1}(1)$ is the occupation probability for mode j_1 at time $k = 1$. Notice that (2.9) is an MoG with mm_0 component Gaussians. Proceeding likewise from this

$\rho(x(1))$, at time $k = 2$, we obtain

$$\rho_{j_2}(x(2)) = \sum_{j_1=1}^{m} \sum_{j_0=1}^{m_0} \pi_{j_1}(1)\alpha_{j_0}\, \mathcal{N}\big(\mu_{j_2}, \Sigma_{j_2}\big), \tag{2.10}$$

$$\text{where } j_2 = 1, \ldots, m,\ \mu_{j_2} = (A_{j_2}A_{j_1})\mu_{j_0},$$

$$\Sigma_{j_2} = (A_{j_2}A_{j_1})\Sigma_{j_0}(A_{j_2}A_{j_1})^\top,$$

$$\rho(x(2)) = \sum_{j_2=1}^{m} \sum_{j_1=1}^{m} \sum_{j_0=1}^{m_0} \pi_{j_2}(2)\pi_{j_1}(1)\alpha_{j_0}\, \mathcal{N}\big(\mu_{j_2}, \Sigma_{j_2}\big), \tag{2.11}$$

Continuing with this recursion till time k, we arrive at (2.7), which is an MoG with $m^k m_0$ components. We comment that the expression is simplified when $m_0 = 1$, i.e. when the initial PDF is Gaussian. □

Remark 2.3 *(**Computational complexity**) Given an initial MoG and a SJLS, from Theorem 2.1, one can in principle compute the state PDF at any finite time, in closed form (i.e., an analytical form with a finite number of well-defined functions). However, since the number of component Gaussians grows exponentially in time, the computational complexity in evaluating (2.7), grows exponentially, and hence the computation becomes intractable. In the following, we show that the Wasserstein based performance analysis can still be performed in closed form while keeping the computational complexity constant in time.*

2.3.2.2 Wasserstein computation in SJLSs

For a SJLS, there are no known results to represent the $\mathcal{W}$ distance in closed form. The main computational issue is that even with Gaussian initial PDF, the instantaneous state PDF remains no longer Gaussian but rather MoG, as shown in Theorem 2.1. This brings forth concerns for the exponential growth of computational complexity to obtain $\rho(x(k))$. To address these concerns, we firstly introduce a

following theorem that enables the Wasserstein computation in an analytical form. Then, we further show that the exponential growth can be avoided by the *core idea* introduced in this subsection.

Theorem 2.2 *($\mathcal{W}$ **for an** m**-mode SJLS with Dirac reference PDF**) At any given time k, let the state PDF for a SJLS be $\rho(x) = \sum_{j=1}^{m} \alpha_j \rho_j(x)$, $x \in \mathbb{R}^n$, where $\rho_j(x)$, α_j, and m are the instantaneous modal PDF, occupation probability of mode j, and the number of individual mixture components, respectively. If we define $W \triangleq \mathcal{W}\big(\rho(x), \delta(x)\big)$, and $W_j \triangleq \mathcal{W}\big(\rho_j(x), \delta(x)\big)$, then*

$$W = \left(\sum_{j=1}^{m} \alpha_j W_j^2 \right)^{1/2}. \tag{2.12}$$

Proof From (2.2) and Proposition 2.3, we have

$$\begin{aligned} W^2 &= \int_{\mathbb{R}^n} \| x \|^2_{\ell_2(\mathbb{R}^n)} \, \rho(x) dx \\ &= \int_{\mathbb{R}^n} \| x \|^2_{\ell_2(\mathbb{R}^n)} \sum_{j=1}^{m} \alpha_j \rho_j(x) dx \\ &= \sum_{j=1}^{m} \alpha_j \int_{\mathbb{R}^n} \| x \|^2_{\ell_2(\mathbb{R}^n)} \, \rho_j(x) dx \\ &= \sum_{j=1}^{m} \alpha_j W_j^2. \end{aligned} \tag{2.13}$$

$$\Rightarrow \quad W = \left(\sum_{j=1}^{m} \alpha_j W_j^2 \right)^{1/2}. \tag{2.14}$$

□

Theorem 2.2 provides an analytical solution to compute the performance and the robustness of the SJLS in terms of Wasserstein distance. However, expression in

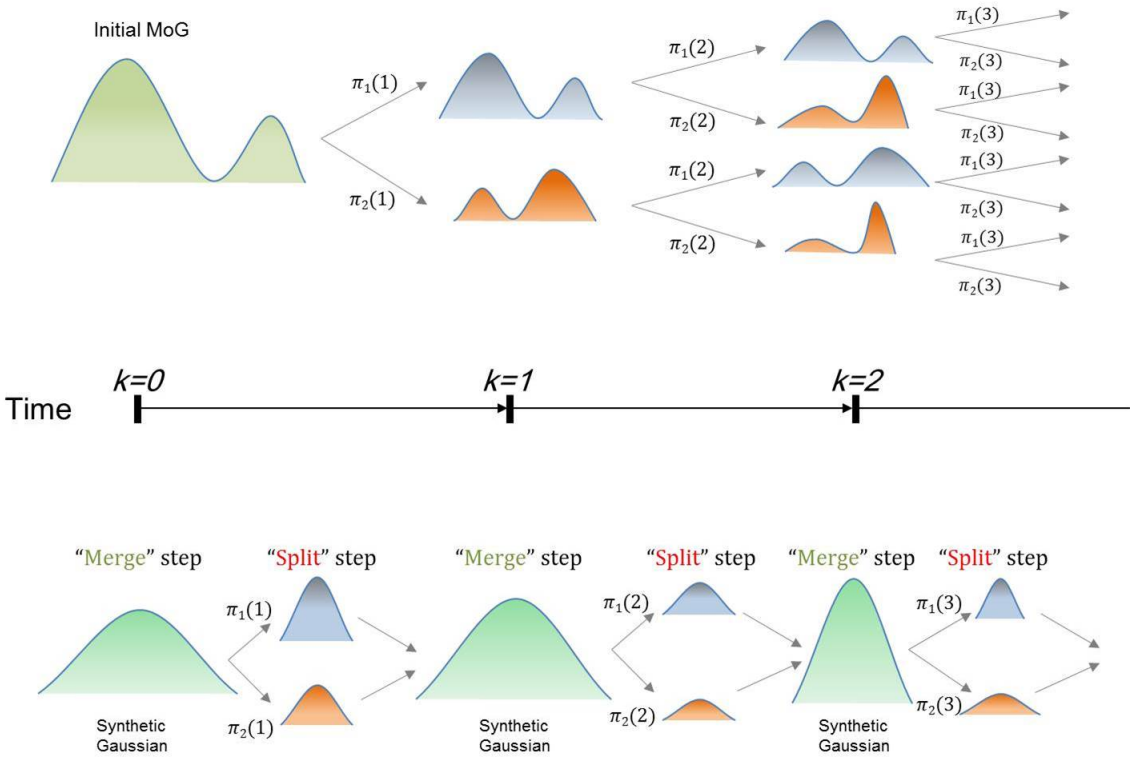


Figure 2.1: Schematic of PDF propagation under stochastic switching. Initially, an MoG PDF was given; Upper one shows the exponential growth of MoG components; Bottom one shows "Split-and-Merge" algorithm where the number of Gaussian components remains constatnt, which is m modes at most. In this figure, $m = 2$.

(2.12) still includes the component-wise $\mathcal{W}$ computation, and hence the computation becomes intractable shortly due to the exponential growth of Gaussian components in the state PDF $\rho(x)$. In order to cope with this problem, we introduce a **"Split-and-Merge"** algorithm as follows.

1) Merge Step:

For a given MoG $\rho(x)$ at any time k, we can compute the mean $\widehat{\mu}$ and the covariance $\widehat{\Sigma}$ of an MoG by the following lemma.

Lemma 2.1 *(**Mean and covariance of a mixture PDF**) Consider any mixture PDF* $\rho(x) = \sum_{j=1}^{m} \alpha_j \rho_j(x)$, *with component mean-covariance pairs* (μ_j, Σ_j), $j = 1, \ldots, m$. *Then, the mean-covariance pair* $\left(\widehat{\mu}, \widehat{\Sigma}\right)$ *for the mixture PDF* $\rho(x)$, *is given by*

$$\begin{aligned} \widehat{\mu} &= \sum_{j=1}^{m} \alpha_j \mu_j, \\ \widehat{\Sigma} &= \sum_{j=1}^{m} \alpha_j \left(\Sigma_j + (\mu_j - \widehat{\mu})(\mu_j - \widehat{\mu})^\top\right). \end{aligned} \tag{2.15}$$

Proof We have

$$\begin{aligned} \widehat{\mu} &\triangleq \int_{\mathbb{R}^n} x\rho(x)dx \\ &= \int_{\mathbb{R}^n} x \sum_{j=1}^{m} \alpha_j \rho_j(x)dx \\ &= \sum_{j=1}^{m} \alpha_j \int_{\mathbb{R}^n} x\rho_j(x)dx \\ &= \sum_{j=1}^{m} \alpha_j \mu_j. \end{aligned}$$

On the other hand,

$$\begin{aligned}
\widehat{\Sigma} &\triangleq \mathbb{E}\left[(x-\widehat{\mu})(x-\widehat{\mu})^{\top}\right] \\
&= \mathbb{E}\left[xx^{\top}\right] - \widehat{\mu}\widehat{\mu}^{\top} \\
&= \int_{\mathbb{R}^n} xx^{\top} \sum_{j=1}^{m} \alpha_j \rho_j(x) dx - \widehat{\mu}\widehat{\mu}^{\top} \\
&= \sum_{j=1}^{m} \alpha_j \int_{\mathbb{R}^n} (x-\widehat{\mu}+\widehat{\mu})(x-\widehat{\mu}+\widehat{\mu})^{\top} \rho_j(x)\, dx - \widehat{\mu}\widehat{\mu}^{\top} \\
&= \sum_{j=1}^{m} \alpha_j \left(\Sigma_j + (\mu_j-\widehat{\mu})(\mu_j-\widehat{\mu})^{\top}\right).
\end{aligned}$$

□

Lemma 2.1 proves that for any mixture PDF, we can compute the mean $\widehat{\mu}$ and the covariance $\widehat{\Sigma}$. Then, from this information, a synthetic Gaussian $\mathcal{N}(\widehat{\mu}(k), \widehat{\Sigma}(k))$, which is a Gaussian PDF, can be constructed.

2) Split Step:

Once the synthetic Gaussian $\mathcal{N}(\widehat{\mu}(k), \widehat{\Sigma}(k))$ is obtained at time k at "Merge step", the modal PDFs, propagated from $\mathcal{N}(\widehat{\mu}(k), \widehat{\Sigma}(k))$ along each modal dynamics $\{A_j\}_{j=1}^{m}$, are computed at "Split step." At this stage, we have m numbers of Gaussian components $\mathcal{N}(A_j\widehat{\mu}(k), A_j\widehat{\Sigma}(k)A_j^{\top})$, $j = 1, 2, \ldots, m$ with a switching probability $\pi(k+1)$. Accordingly, an MoG PDF, which has m-mode numbers in Gaussian components, is constructed by Proposition 2.3 at time $k+1$.

Repeating "Split-and-Merge" algorithm at every time step as depicted in Fig. 2.1, linear modal dynamics results in m modal Gaussian PDFs ("Split step"). Then, instead of computing the non-Gaussian SJLS state PDF in an MoG form, one would construct a synthetic Gaussian $\mathcal{N}(\widehat{\mu}, \widehat{\Sigma})$ ("Merge step") corresponding to the actual state PDF $\rho(x)$, and repeat thereafter.

Although the "Split-and-Merge" algorithm obviates the need to compute the state PDF $\rho(x)$ where Gaussian components grow exponentially, it does not imply that $\mathcal{N}(\widehat{\mu}, \widehat{\Sigma})$ substitutes $\rho(x)$. Since the state PDF $\rho(x)$, expressed in an MoG form, have higher moments other than first and second, $\rho(x)$ is a non-Gaussian PDF and differs from $\mathcal{N}(\widehat{\mu}, \widehat{\Sigma})$ that is a Gaussian PDF. Most importantly, however, we address that both $\rho(x)$ and $\mathcal{N}(\widehat{\mu}, \widehat{\Sigma})$ are equidistant to $\delta(x)$ in terms of the Wasserstein distance as proved in the following theorem. (i.e., $\mathcal{W}(\rho(x), \delta(x)) = \mathcal{W}(\mathcal{N}(\widehat{\mu}, \widehat{\Sigma}), \delta(x)))$.

Theorem 2.3 *(**Equidistance between** W **and** $\widehat{W}$) At any given time k, let the state PDF for an m-mode SJLS $\rho(x(k))$, be of the form (2.7), which we rewrite as* $\rho\left(x(k)\right) = \sum_{j_k=1}^{m} \sum_{j_0=1}^{m_0} \alpha_{j_0} \beta_{j_k} \mathcal{N}\left(\mu_{j_k}, \Sigma_{j_k}\right)$, *where* $\beta_{j_k} \triangleq \sum_{j_{k-1}=1}^{m} \cdots \sum_{j_1=1}^{m} \left(\prod_{r=1}^{k} \pi_{j_r}(r) \right)$, $\mu_{j_k} = A^*_{j_k} \mu_{j_0}$, $\Sigma_{j_k} = A^*_{j_k} \Sigma_{j_0} A^{*\top}_{j_k}$, *and* $A^*_{j_k} \triangleq \prod_{r=k}^{1} A_{j_r}$. *Let the instantaneous mean and covariance of the mixture PDF $\rho(x(k))$ be $\widehat{\mu}(k)$ and $\widehat{\Sigma}(k)$, respectively. Then, we have*

$$\widehat{W}(k) = W(k) = \left(\sum_{j_k=1}^{m} \sum_{j_0=1}^{m_0} \alpha_{j_0} \beta_{j_k} W^2_{j_k}(k) \right)^{1/2}, \forall k \in \mathbb{N}_0, \tag{2.16}$$

where

$$\widehat{W}(k) \triangleq \mathcal{W}\left(\mathcal{N}\left(\widehat{\mu}(k), \widehat{\Sigma}(k) \right), \delta(x) \right),$$

$$W(k) \triangleq \mathcal{W}\left(\rho\left(x(k)\right), \delta(x) \right),$$

$$W_{j_k}(k) \triangleq \mathcal{W}\left(\mathcal{N}\left(\mu_{j_k}, \Sigma_{j_k}\right), \delta(x) \right),$$

$$\mu_{j_k} = A^*_{j_k} \mu_{j_0},\ \Sigma_{j_k} = A^*_{j_k} \Sigma_{j_0} A^{*\top}_{j_k}, \quad \forall k \geq 1.$$

Proof The rightmost equality in (2.16), follows directly from Theorem 2.2. Thus,

it suffices to prove that $\widehat{W}(k) = \left(\sum_{j_k=1}^{m}\sum_{j_0=1}^{m_0}\alpha_{j_0}\beta_{j_k}W_{j_k}^2(k)\right)^{1/2}$.

At time $k=0$, from the "Merge step", the mean and the covariance pair $(\widehat{\mu}_0, \widehat{\Sigma}_0)$ of an initial MoG can be computed by $(\widehat{\mu}_0, \widehat{\Sigma}_0) = \big(\sum_{j_0=1}^{m_0}\alpha_{j_0}\mu_{j_0}, \sum_{j_0=1}^{m_0}(\Sigma_{j_0} + (\mu_{j_0} - \widehat{\mu}_0)(\mu_{j_0}-\widehat{\mu}_0)^\top)\big)$ from Lemma 2.1. If we construct a synthetic Gaussian $\mathcal{N}(\widehat{\mu}_0, \widehat{\Sigma}_0)$, the Wasserstein distance $\widehat{W}$ at time $k=0$ can be computed by (3.6) as follows.

$$\begin{aligned}\widehat{W}^2(0) &\overset{(3.6)}{=} \| \widehat{\mu}_0 \|_{\ell_2(\mathbb{R}^n)}^2 + \mathrm{tr}(\widehat{\Sigma}_0) \\ &\overset{(2.15)}{=} \widehat{\mu}_0^\top\widehat{\mu}_0 + \mathrm{tr}\left(\sum_{j_0=1}^{m_0}\alpha_{j_0}\Big(\Sigma_{j_0} + (\mu_{j_0}-\widehat{\mu}_0)(\mu_{j_0}-\widehat{\mu}_0)^\top\Big)\right). \end{aligned} \tag{2.17}$$

Since $\mathrm{tr}(\cdot)$ is a linear operator, we can expand (2.17) as

$$\begin{aligned}\widehat{W}^2(0) = \widehat{\mu}_0^\top\widehat{\mu}_0 &+ \sum_{j_0=1}^{m_0}\alpha_{j_0}\mathrm{tr}\left(\Sigma_{j_0}\right) + \mathrm{tr}\left(\sum_{j_0=1}^{m_0}\alpha_{j_0}\mu_{j_0}\mu_{j_0}^\top\right) \\ &- \mathrm{tr}\left(\left(\sum_{j_0=1}^{m_0}\alpha_{j_0}\mu_{j_0}\right)\widehat{\mu}_0^\top\right) - \mathrm{tr}\left(\widehat{\mu}_0\left(\sum_{j_0=1}^{m_0}\alpha_{j_0}\mu_{j_0}\right)^\top\right) \\ &+ \mathrm{tr}\left(\widehat{\mu}_0\widehat{\mu}_0^\top\right). \end{aligned} \tag{2.18}$$

Recalling that $\widehat{\mu}_0 = \sum_{j_0=1}^{m_0}\alpha_{j_0}\mu_{j_0}$ and $\widehat{\mu}_0^\top\widehat{\mu}_0 = \mathrm{tr}\left(\widehat{\mu}_0^\top\widehat{\mu}_0\right) = \mathrm{tr}\left(\widehat{\mu}_0\widehat{\mu}_0^\top\right)$, the first, fourth, fifth and sixth term in the right-hand-side of (2.18) cancel out, resulting in

$$\begin{aligned}\widehat{W}^2(0) &= \sum_{j_0=1}^{m_0}\alpha_{j_0}\,\mathrm{tr}\left(\mu_{j_0}\mu_{j_0}^\top\right) + \sum_{j_0=1}^{m_0}\alpha_{j_0}\mathrm{tr}\left(\Sigma_{j_0}\right) \\ &= \sum_{j_0=1}^{m_0}\alpha_{j_0}\left(\| \mu_{j_0} \|_{\ell_2(\mathbb{R}^n)}^2 + \mathrm{tr}\left(\Sigma_{j_0}\right)\right) \\ &= \sum_{j_0=1}^{m_0}\alpha_{j_0}\mathcal{W}^2\Big(\mathcal{N}(\mu_{j_0},\Sigma_{j_0}), \delta(x)\Big) \\ &= \sum_{j_0=1}^{m_0}\alpha_{j_0}W_{j_0}^2(0) \overset{(2.12)}{=} W^2(0). \end{aligned} \tag{2.19}$$

Hence, $\widehat{W}(0)$ is equivalent to $W(0)$.

At time $k=1$, we propagate the modal PDFs from a synthetic Gaussian $\mathcal{N}(\widehat{\mu}_0, \widehat{\Sigma}_0)$, which results in m modal Gaussians $\mathcal{N}(A_{j_1}\widehat{\mu}_0, A_{j_1}\widehat{\Sigma}_0 A_{j_1}^\top)$, $j_1 = 1, 2, \ldots, m$ at "Split step". After that, a synthetic Gaussian $\mathcal{N}(\widehat{\mu}_1, \widehat{\Sigma}_1)$, where $\widehat{\mu}_1 = \sum_{j_1=1}^m \pi_{j_1}(1) A_{j_1}\widehat{\mu}_0$ and $\widehat{\Sigma}_1 = \sum_{j_1=1}^m \pi_{j_1}(1)\Big(A_{j_1}\widehat{\Sigma}_0 A_{j_1}^\top + (A_{j_1}\widehat{\mu}_0 - \widehat{\mu}_1)(A_{j_1}\widehat{\mu}_0 - \widehat{\mu}_1)^\top\Big)$ is constructed at "Merge step". Then, $\widehat{W}(1)$ computation is carried out as follows.

$$\begin{aligned}
\widehat{W}^2(1) &\overset{(3.6)}{=} \| \widehat{\mu}_1 \|^2_{\ell_2(\mathbb{R}^n)} + \mathrm{tr}\big(\widehat{\Sigma}_1\big) \\
&= \widehat{\mu}_1^\top \widehat{\mu}_1 + \mathrm{tr}\left(\sum_{j_1=1}^m \pi_{j_1}(1)\Big(A_{j_1}\widehat{\Sigma}_0 A_{j_1}^\top + \big(A_{j_1}\widehat{\mu}_0 - \widehat{\mu}_1\big)\big(A_{j_1}\widehat{\mu}_0 - \widehat{\mu}_1\big)^\top\Big)\right). \quad (2.20)
\end{aligned}$$

By exactly same procedures in (2.18), term cancellations result in

$$\begin{aligned}
\widehat{W}^2(1) &= \sum_{j_1=1}^m \pi_{j_1}(1)\left(\mathrm{tr}\Big(A_{j_1}\widehat{\mu}_0\widehat{\mu}_0^\top A_{j_1}^\top + A_{j_1}\widehat{\Sigma}_0 A_{j_1}^\top\Big)\right) \\
&\overset{(2.15)}{=} \sum_{j_1=1}^m \pi_{j_1}(1)\left(\mathrm{tr}\Big(A_{j_1}\Big(\sum_{j_0=1}^{m_0}\alpha_{j_0}\big(\mu_{j_0}\mu_{j_0}^\top + \Sigma_{j_0}\big)\Big)A_{j_1}^\top\Big)\right) \\
&= \sum_{j_1=1}^m \sum_{j_0=1}^{m_0} \pi_{j_1}(1)\alpha_{j_0}\Big(\ \| \mu_{j_1} \|^2_{\ell_2(\mathbb{R}^n)} + \mathrm{tr}\big(\Sigma_{j_1}\big)\Big) \\
&= \sum_{j_1=1}^m \sum_{j_0=1}^{m_0} \pi_{j_1}(1)\alpha_{j_0} W_{j_1}^2(1) \overset{(2.12)}{=} W^2(1), \quad (2.21)
\end{aligned}$$

where $\mu_{j_1} = A_{j_1}\mu_{j_0}$ and $\Sigma_{j_1} = A_{j_1}\Sigma_{j_0}A_{j_1}^\top$.

Continuing in this manner, finally we obtain the following result for any time k.

$$\begin{aligned}\widehat{W}^2(k) &= \sum_{j_k=1}^{m}\cdots\sum_{j_1=1}^{m}\sum_{j_0=1}^{m_0}\left(\prod_{r=1}^{k}\pi_{j_r}(r)\right)\alpha_{j_0}\Big(\parallel\mu_{j_k}\parallel^2_{\ell_2(\mathbb{R}^n)}+\mathrm{tr}\big(\Sigma_{j_k}\big)\Big)\\ &= \sum_{j_k=1}^{m}\cdots\sum_{j_1=1}^{m}\sum_{j_0=1}^{m_0}\left(\prod_{r=1}^{k}\pi_{j_r}(r)\right)\alpha_{j_0}W^2_{j_k}(k)\\ &\overset{(2.12)}{=} W^2(k), \end{aligned} \tag{2.22}$$

where $\mu_{j_k}=A_{j_k}A_{j_{k-1}}\cdots A_{j_1}\mu_{j_0}=A^*_{j_k}\mu_{j_0}$,
$\Sigma_{j_k}=\big(A_{j_k}A_{j_{k-1}}\cdots A_{j_1}\big)\Sigma_{j_0}\big(A_{j_k}A_{j_{k-1}}\cdots A_{j_1}\big)^\top=A^*_{j_k}\Sigma_{j_0}A^{*\top}_{j_k}$. □

According to Theorem 2.3, it is unnecessary to propagate the state PDF $\rho(x)$ and to compute W, which is intractable due to the exponential growth of Gaussian components. Instead, we can analyse the performance of the SJLS through $\widehat{W}$, since $\widehat{W}$ is equal to W at all time k.

The major advantages of the "Split-and-Merge" algorithm with $\widehat{W}$ computation for the performance and the robustness analysis can be summarized in the following sense. $\widehat{W}$ computation using (3.6) provides an analytical solution, which is computationally efficient. In addition, at any time step, we only have m mean vectors and covariance matrices to work with, and hence the scalability problem with an exponential growth can be avoided.

Remark 2.4 ***(Applicability of the performance and the robustness measure to general SJLSs)*** *Since the switching probability $\pi(k)$ is an independent variable with regard to $\widehat{W}(k)$ as described in Theorem 2.3, we can compute $\widehat{W}(k)$ for any SJLSs regardless of the updating rule for $\pi(k)$. Once $\pi(k)$ is computed at time k by governing recursion equation (e.g., i.i.d., Markov, or semi-Markov jump process, etc.), the performance and the robustness for SJLSs are measured by $\widehat{W}(k)$. Hence,*

the proposed method for the performance and the robustness analysis can be applied to generic SJLSs.

2.3.3 Alternative Proof for Stability of Switched Systems via Wasserstein Metric

Proposition 2.4 *(**MoG state PDF and a synthetic Gaussian are equidistant from Dirac**) Starting from an initial MoG joint PDF* $\varsigma_0(k) = \sum_{j_0=1}^{m_0} \alpha_{j_0} \mathcal{N}(\mu_{j_0}, \Sigma_{j_0})$, *let* $\varsigma(k)$ *be the joint state PDF at time* k, *for stochastic jump linear systems with arbitrary switching probability* $\pi(k)$. *Further, let the mean and covariance for* $\varsigma(k)$, *be denoted as* $\hat{\mu}(k)$ *and* $\hat{\Sigma}(k)$, *respectively. Let* $W(k) \triangleq W(\varsigma(k), \delta(x))$, *and* $\hat{W}(k) \triangleq W\left(\mathcal{N}\left(\hat{\mu}(k), \hat{\Sigma}(k)\right), \delta(x)\right)$. *Then*

$$W^2(k) = \hat{W}^2(k) = vec(I_n)^\top \Gamma(k) vec(\hat{\mu}(0)\hat{\mu}(0)^\top + \hat{\Sigma}(0)), \tag{2.23}$$

where I_n *denotes the* $n \times n$ *identity matrix, and* $\hat{\mu}(0) = \sum_{j_0=1}^{m_0} \alpha_{j_0}\mu_{j_0}$, $\hat{\Sigma}(0) = \sum_{j=1}^{m_0} \alpha_{j_0}(\Sigma_{j_0} + (\mu_{j_0} - \hat{\mu}(0))(\mu_{j_0} - \hat{\mu}(0))^\top)$ *are the mean and covariance of* ς_0, *respectively. The matrix* $\Gamma(k)$ *is defined as* $\Gamma(k) \triangleq \prod_{i=k}^{1} \left(\sum_{j=1}^{m} \pi_j(i)(A_j \otimes A_j) \right)$, *which is the product of matrices in reverse order w.r.t. time. The proof is given in Appendix.*

Corollary 2.2 *Suppose that* $\{\sigma_k\}$ *is an arbitrary switching sequence of the jump linear system with the occupation probability* $\pi(k)$, *satisfying* $\pi_{i_k}(k) = 1$ *and* $\pi_{j_k}(k) = 0$, $\forall i_k \neq j_k$ *for all time* k. *Then, the jump linear system is m.s. stable iff there is a finite time* k *such that*

$$\| A_{i_k} A_{i_{k-1}} \cdots A_{i_2} A_{i_1} \| < 1, \tag{2.24}$$

where $A_{i_j} \in \{A_1, A_2, \ldots, A_m\}$, $\forall j$ *and* $\| \cdot \|$ *denotes any matrix norm.*

Proof If $\pi(k)$ obeys $\pi_{i_k}(k) = 1$ and $\pi_{j_k}(k) = 0$, $\forall i_k \neq j_k$ for all k, then the matrix $\Gamma(k)$ in (2.23) becomes $\Gamma(k) = \prod_{p=k}^{1} \left(A_{i_p} \otimes A_{i_p}\right)$.

Since we have

$$\left|\left| \prod_{j=k}^{1} A_{i_j} \right|\right| = \left(\left|\left| \prod_{j=k}^{1} \left(A_{i_j} \otimes A_{i_j}\right) \right|\right| \right)^{\frac{1}{2}},$$

it is easily shown that

$$\begin{aligned} &\| \left(A_{i_k} \otimes A_{i_k}\right) \left(A_{i_{k-1}} \otimes A_{i_{k-1}}\right) \cdots \left(A_{i_2} \otimes A_{i_2}\right) \left(A_{i_1} \otimes A_{i_1}\right) \| < 1 \\ \iff \quad &\| A_{i_k} A_{i_{k-1}} \cdots A_{i_2} A_{i_1} \| < 1. \end{aligned}$$

Therefore, $\Gamma(k)$ is a contraction mapping and hence $W \to 0 \Leftrightarrow$ m.s. stable, if $\| A_{i_k} A_{i_{k-1}} \cdots A_{i_2} A_{i_1} \| < 1$, $\forall k$. The necessity can be proved by contradiction. $\square$

The condition for the switching probability $\pi(k)$, described in Corollary 2.2 denotes the deterministic jump process. In [68], the authors addressed that the equation in (2.24) guarantees a global uniform asymptotic stability for deterministic jump linear systems. Note that Corollary 2.2 stands for the stability condition in the mean square sense. However, in the case when the initial distribution is given by Dirac PDF located at arbitrary x_0, i.e., $\varsigma_0 = \delta(x - x_0)$, the proposed Wasserstein framework also recovers the uniform stability since there is no uncertainty for both the system state x and the deterministic switching $\pi(k)$. Hence, the $\mathbb{E}[\cdot]$ can be obviated, resulting in the uniform stability.

Now, we prove the already-known stability conditions for i.i.d. jump processes through the convergence of the Wasserstein distance.

Suppose that $\{\sigma_k\}$ is generated by an i.i.d. process with probability distribution $\{\pi_1, \pi_2, \cdots, \pi_m\}$ over the set $\{1, 2, \ldots, m\}$. In Corollary 2.7 of [33], the necessary and sufficient condition for m.s. stability of an i.i.d. jump linear system is given by that the matrix

$$\mathcal{A} \triangleq \sum_{j=1}^{m} \pi_j \left(A_j \otimes A_j\right) = \pi_1(A_1 \otimes A_1) + \pi_2(A_2 \otimes A_2) + \cdots + \pi_m(A_m \otimes A_m) \quad (2.25)$$

is Schur stable. We next *recover* this result from the Wasserstein distance perspective.

Theorem 2.4 *Consider an i.i.d. jump linear system, where $\pi(k)$ is a stationary probability vector $\{\pi_1, \pi_2, \cdots, \pi_m\}$ for all k. The i.i.d. jump linear system is m.s. stable if and only if the matrix*

$$\mathcal{A} \triangleq \sum_{j=1}^{m} \pi_j \left(A_j \otimes A_j\right)$$

is Schur stable, i.e. $\rho(\mathcal{A}) < 1$.

Proof Since the jump stochastic process is i.i.d., the underlying probability vector $\pi(k)$ that generates the switching sequence $\{\sigma_k\}$, is a time-invariant probability vector $\{\pi_1, \pi_2, \cdots, \pi_m\}$. As a consequence, (2.23) can be simplified as

$$W^2(k) = \mathrm{vec}(I_n)^\top (\mathcal{A}^k) \mathrm{vec}(\hat{\mu}(0)\hat{\mu}(0)^\top + \hat{\Sigma}(0)),$$

where $\mathcal{A} = \left(\sum_{j=1}^{m} \pi_j \left(A_j \otimes A_j\right)\right)$. However, it is well known that

$$\lim_{k \to \infty} \mathcal{A}^k = 0$$

iff $\rho(\mathcal{A}) < 1$. Therefore,

$$\lim_{k\to\infty} W^2 \to 0 \Leftrightarrow \rho(\mathcal{A}) < 1.$$

In addition, Proposition 2.1 tells us that

$$\lim_{k\to\infty} W \to 0 \Leftrightarrow \text{m.s. stability.}$$

Combining these two, we arrive at $\rho(\mathcal{A}) < 1 \Leftrightarrow \lim_{k\to\infty} W \to 0 \Leftrightarrow$ m.s. stability for i.i.d. jump linear system. □

2.4 Numerical Example

Consider the inverted pendulum on cart in Fig. 2.2 with parameters described in Table 2.1. Originally, this example was introduced in [97] with single communication delay term τ_k between sensor and controller.

Table 2.1: Nomenclature for Inverted Pendulum Dynamics.

Symbol	definition	Symbol	definition
m_1	cart mass	m_2	pendulum mass
L	pendulum length	x	cart position
θ	pendulum angle	u	input force

The system states are $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$. We assume that $m_1 = 1$kg, $m_2 = 0.5$kg, $L = 1$m with friction-free floor. Later, this example was further exploited by [104] with two random delays τ_k and d_k which are sensor-to-controller and controller-to-actuator delays, respectively. The sets of mode are $\mathcal{M}(\tau_k) = \{0, 1, 2\}$ and $\mathcal{M}(d_k) = \{0, 1\}$. When the control action is taken at

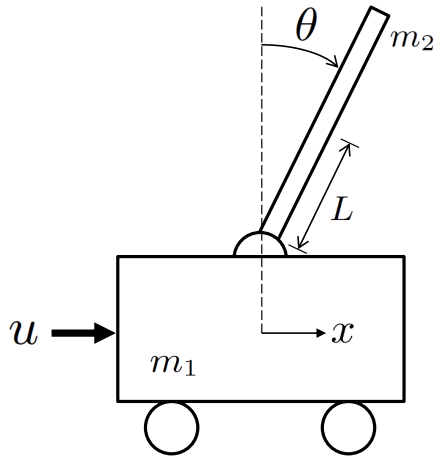


Figure 2.2: Inverted pendulum on cart.

time k, the controller-to-actuator delay d_k is unknown, but τ_k and d_{k-1} are found. Accordingly, controller gain F is dependent on τ_k and d_{k-1}. Hence, the linearized closed-loop system model with sampling time $T_s = 0.1$ is denoted by

$$x(k+1) = Ax(k) + BF(\tau_k, d_{k-1})x(k - \tau_k - d_k),$$

where

$$A = \begin{bmatrix} 1 & 0.1 & -0.0166 & -0.0005 \\ 0 & 1 & -0.3374 & -0.0166 \\ 0 & 0 & 1.0996 & 0.1033 \\ 0 & 0 & 2.0247 & 1.0996 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0045 \\ 0.0896 \\ -0.0068 \\ -0.1377 \end{bmatrix}$$

with the controller gain F's given in [104]:

$$F(0,0) = \begin{bmatrix} 0.1690 & 0.8824 & 19.5824 & 4.3966 \end{bmatrix},$$
$$F(0,1) = \begin{bmatrix} 0.5625 & 0.6259 & 24.8814 & 5.1886 \end{bmatrix},$$
$$F(1,0) = \begin{bmatrix} -0.3076 & 0.9370 & 12.0069 & 5.9910 \end{bmatrix},$$
$$F(1,1) = \begin{bmatrix} -0.0097 & 0.7109 & 15.2518 & 7.3154 \end{bmatrix},$$
$$F(2,0) = \begin{bmatrix} -0.3212 & 1.0528 & 11.9330 & 6.3809 \end{bmatrix},$$
$$F(2,1) = \begin{bmatrix} 0.0427 & 0.8640 & 16.0874 & 7.8361 \end{bmatrix}.$$

Therefore, the closed-loop dynamics of this system has total 6 modes $\mathcal{M} = \{1, 2, \ldots, 6\}$. In this example, we consider two different types of communication delays as follows.

1) Markovian Communication Delays:

We denote the transition probability of sensor-to-controller and controller-to-actuator delays as λ_{ij} and ω_{rs}, respectively. Then, λ_{ij} and ω_{rs} are defined by

$$\lambda_{ij} = \mathbb{P}(\tau_{k+1} = j | \tau_k = i),\ \omega_{rs} = \mathbb{P}(d_{k+1} = s | d_k = r),$$

where $\lambda_{ij}, \omega_{rs} \geq 0$ and $\sum_{j=0}^{2} \lambda_{ij} = 1$, $\sum_{s=0}^{1} \omega_{rs} = 1$. Given individual Markov transition probability matrices

$$\Lambda = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.3 & 0.6 & 0.1 \\ 0.3 & 0.6 & 0.1 \end{bmatrix}, \quad \Omega = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \end{bmatrix}$$

corresponding to λ_{ij} and ω_{rs}, the Markov transition probability matrix P for 6 modes MJLS is obtained from $P = \Lambda \otimes \Omega$ as in [97]. The switching probability distribution $\pi(k)$ is updated by the linear recursion equation $\pi(k+1) = \pi(k)P$ with initial probability distribution $\pi(0) = [1,\ 0,\ 0,\ 0,\ 0,\ 0]$.

2) i.i.d. Communication Delays:

Although the previous examples in [97,104] assumed that the communication delays are governed by Markov process, we adopt an i.i.d. jump process to manifestly show that the proposed methods are also applicable to other types of SJLSs. In case of i.i.d. jump process, the occupation probability vector $\pi(k)$ is stationary, and hence it does not change over time. We assume that the switching probabilities π_{sc} and π_{ca} are given by

$$\pi_{sc} = [0.7,\ 0.2,\ 0.1], \quad \pi_{ca} = [0.5,\ 0.5],$$

where π_{sc} and π_{ca} stand for the switching probability distribution with respect to sensor-to-controller and controller-to-actuator, respectively. Then, the occupation probability vector π for this inverted pendulum system is computed by $\pi = \pi_{sc} \otimes \pi_{ca}$.

Differently from [104] where the initial state is deterministically given, we assume that the system contains initial state uncertainty, given as a Gaussian distribution $\rho(0) = \mathcal{N}(\mu(0), \Sigma(0))$ where the mean $\mu(0) = [0, 0, 0.1, 0]^\top$, the covariance $\Sigma(0) = 0.25^2 I_{4\times4}$, and $I_{4\times4}$ denotes 4×4 identity matrix. Also, we test another type of initial state uncertainty, given as an MoG PDF expressed by a bimodal Gaussian in the following form

$$\rho(0) = \sum_{j=1}^{2} \alpha_j(0) \mathcal{N}(\mu_j(0), \Sigma_j(0)),$$

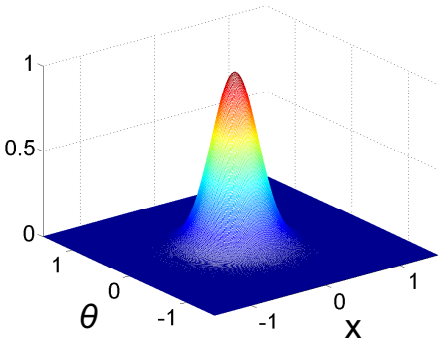
where $\alpha_1(0) = 0.5$ and $\alpha_2(0) = 0.5$. The mean and the covariance for each Gaussian component are given by

$$\mu_1(0) = \begin{bmatrix} 0.5, & 0.25, & -0.12, & 0.05 \end{bmatrix}^\top, \ \Sigma_1(0) = 0.25^2 I_{4\times 4},$$
$$\mu_2(0) = \begin{bmatrix} -0.4, & 0.35, & 0.07, & -0.1 \end{bmatrix}^\top, \ \Sigma_2(0) = 0.3^2 I_{4\times 4}.$$

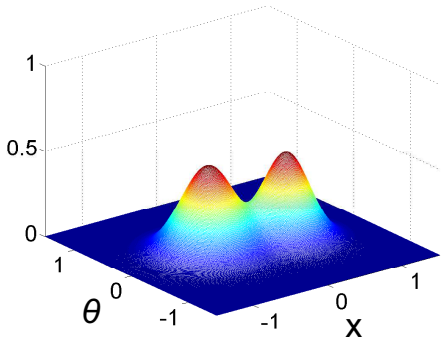
This type of multimodal uncertainties is caused by various factors such as sensing under interference [39], distributed sensor networks [57], multitaget tracking problems [72] and so forth. The bivariate marginal distribution associated with state x and θ for these Gaussian and MoG PDF are shown in Fig. 2.4 and Fig. 2.4, respectively.

In Fig. 2.4, the performance of this inverted pendulum system, incorporating with two different initial state uncertainties with two different stochastic jump processes, are depicted via $\widehat{W}$ computation. For all cases, it is shown that the system is m.s. stable from the convergence of $\widehat{W}$. However, the rate of convergence and the transient performance show different aspects in the transient time. Among all cases, $\widehat{W}$ for i.i.d. jump process with initial MoG PDF converges fast with low bounce, whereas $\widehat{W}$ for MJLS with initial Gaussian PDF slowly converges with high bounce.

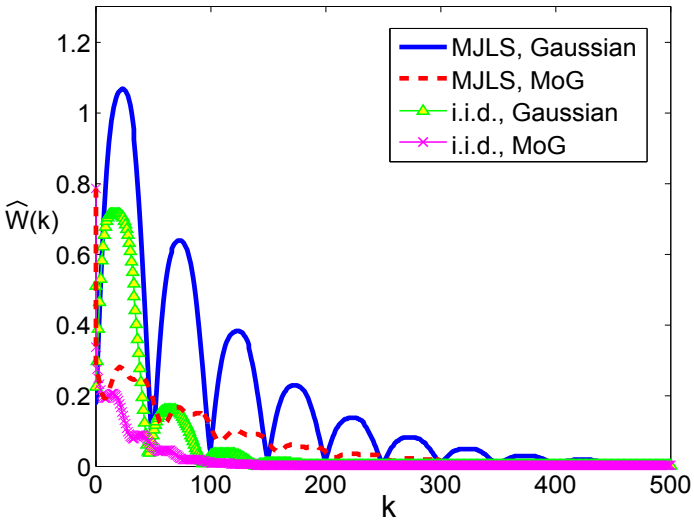
At every time step, the "Split-and-Merge" algorithm, presented in Section 2.3.2.2 is used to propagate the state PDFs. Without using these techniques, it is practically impossible to propagate density functions and calculate W (i.e., the Wasserstein distance between actual state PDF $\rho(x)$ and $\delta(x)$) even for finite switching modes. The number of Gaussian components that represents the state PDF after N time steps is 6^N, which soon becomes computationally intractable. For an m-mode SJLS, the growth rate is m^N. With the implementation of the proposed "Split-and-Merge"



(a) Gaussian marginal distribution



(b) MoG marginal distribution



(c) Wasserstein distance with different stochastic jump processes and initial PDFs; MJLS with Gaussian (blue solid), MJLS with MoG (red dashed), i.i.d. with Gaussian (green triangle), and i.i.d. with MoG (purple cross).

Figure 2.3: Simulation result for performance and robustness analysis of inverted pendulum system with the existence of both random communication delays and initial state uncertainties.

algorithm, $\widehat{W}$ that is equivalent to W was computed without scalability problems. From this example, it is clearly shown that the performance and the robustness for SJLSs can be measured via $\widehat{W}$, which quantifies the uncertainties.

2.5 Concluding Remarks

This chapter provided new tools for the performance and the robustness analysis of stochastic jump linear systems. New methods for analysis of such systems were developed with the Wasserstein distance. Scalability issues in uncertainty propagation, due to the exponential increase in the number of modal PDFs, was circumvented with a novel "Split-and-Merge" algorithm. Also, it was shown that the convergence of the Wasserstein distance, with a particular choice of Dirac distribution as a reference PDF, implies mean square stability. These results address both transient and steady-state behavior of stochastic jump linear systems and can be implemented in a computationally efficient manner. The new framework for the performance and the robustness analysis are applicable to general jump linear systems, that may not satisfy Markovian properties. Finally, the practical usefulness of the proposed methods were verified using examples.

3. OPTIMAL SWITCHING SYNTHESIS FOR SWITCHED SYSTEMS WITH GAUSSIAN INITIAL STATE UNCERTAINTY

3.1 Introduction

A jump linear system is defined as a dynamical system consisting of a finite number of subsystems and a switching rule that governs a switching between the family of linear subsystems. Over decades, a variety of researches for jump linear systems have been investigated because of its practical implementation. For example, a jump linear system can be used for power systems, manufacturing systems, aerospace systems, networked control systems, etc [10], [14], [24].

In general, problems for jump linear systems branch out into two different fields. The first one is the stability analysis under given switching laws. Since a certain switching law between individually stable subsystem can make the jump linear system unstable [66], it is very important to identify conditions under which system can be stable. Interestingly, the jump linear system also can be stable by switching between unstable subsystems. Fang *et al.* [31] showed sufficient conditions for stability of jump linear systems under arbitrary switching using linear matrix inequalities (LMIs). Lin *et al.* [68] showed necessary and sufficient conditions for asymptotic stability of jump linear systems using finite n-tuple switching sequences, satisfying a certain condition. In addition, broad analysis regarding stability for jump linear systems has been accomplished within few decades [34], [67], [49], [89], [63], [64].

On the other hand, switching synthesis problem, which is another branch of jump linear systems, is relatively new and few investigations have been achieved. Since the main objective is to design switching sequences that establish both the stability and the performance, this case is much harder than stability analysis problem. For

instance, Das and Mukherjee [22] solved the problem for an optimal switching of jump linear systems using Pontryagin's minimum principle. In this method, two-point boundary value problem was solved via relaxation method, where ordinary differential equations are approximated by finite difference equations on mesh points. Therefore, the optimality and computational cost depend on mesh size. In addition, the time to find optimal solution varies according to guess solution. Egerstedt *et al.* [28] addressed a method to find derivative of the cost function with respect to switching time. However, in this chapter, switching sequences are already given and the main focus is to find switching time. Although several other researches regarding optimal control problem together with optimal switching were studied for switched nonlinear systems [48], [100], [5], they may not fit to pure optimal switching problem for jump linear systems.

Here we address optimal switching problem for jump linear systems with given multi-controllers. Multi-controller switching scheme is widely used, such as plant stabilization [75], system performance [68], adaptive control [76], and resource-constrained scheduling [8]. Under the assumption that more than two controllers are given to user, our objective is to find the optimal switching sequence which attains the best performance of the system by controller switching. We can also extend our method to multi-model switching problem by generalizing the multi-controller switching problem. Consequently, we aim to synthesize switching protocols that result in the optimality for the performance of jump linear systems. Moreover, we address the optimal switching problem with initial state uncertainties. In the practical perspective, initial state may contain uncertainties that usually come from measurement errors or sensor inaccuracies. Then, the system state is expressed as random variables represented by PDFs. We assume that the initial state PDF has a form of Gaussian distribution that is very common for real implementation. In order to measure the performance

of the jump linear system with a given Gaussian PDF, we need to adopt a proper metric. In this chapter, Wasserstein metric that assesses the distance between PDFs is used as a tool for both the stability and the performance measure. Hence, we introduce the optimal switching synthesis to achieve the optimality of the system performance with given Gaussian initial PDF by minimizing the objective function that is expressed in terms of Wasserstein distance. We also prove that the convergence of Wasserstein distance implies the mean square stability for the jump linear systems.

Rest of this chapter is organized as follows. Section 3.2 introduces the problem we want to solve. Brief explanations of Wasserstein distance are described in Section 3.3. Section 3.4 provides a way to solve optimal switching problems using receding horizon framework when Gaussian initial state uncertainty exists. Then, Section 3.5 demonstrates the validation of proposed methods by examples and Section 3.6 concludes this chapter.

3.2 Problem Description

Consider a discrete-time linear system with multi-controller, given by

$$x(k+1) = Ax(k) + Bu_{\sigma_k}(x), \quad k \in \mathbb{Z}^+, \sigma_k \in \mathcal{I} \tag{3.1}$$

$$u_{\sigma_k}(x) = K_{\sigma_k} x \tag{3.2}$$

where the state vectors $x \in \mathbb{R}^n$, control inputs $u_\sigma \in \mathbb{R}^m$, the system matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and the set of modes $\mathcal{I} = \{1, 2, \cdots, m\}$. Note that the system matrix A is time-invariant and user can select one controller K_{σ_k} out of multiple choices. Without loss of generality, we can convert system (3.1)-(3.2) to the following

jump linear systems by letting $A_{\sigma_k} := A + BK_{\sigma_k}$.

$$x(k+1) = A_{\sigma_k} x(k), \quad k \in \mathbb{Z}^+, \sigma_k \in \mathcal{I} \tag{3.3}$$

where the system matrices $A_{\sigma_k} \in \mathbb{R}^{n \times n}$.

The system in (3.3) represents not only the controller switching as depicted in (3.1)-(3.2), but also the system mode switching. Hence, we consider the jump linear system model (3.3) and we assume that individual subsystem dynamics A_{σ_k} are Schur stable. Our objective is to find the switching sequence, $\sigma = \{\sigma_1, \sigma_2, \cdots\}$, which guarantees the optimal performance of the switched system. For example, with multi-controller, we want to design a switching law which makes the system states reach the origin as fast as possible. Therefore, our aim is not to design controllers, but rather to synthesize the optimal switching sequence.

For simplicity, we assume that there are two different controllers, which are good and poor in terms of system performance. The closed-loop dynamics are given by A_1 and A_2, respectively. In general, the reason to design multi-controller with respect to single system is to attain not only the system performance but also system stability, robustness, resource-optimal scheduling, etc.

The convergence rate of system state is determined by spectral radius $\rho(A_\sigma) := \max_j |\lambda_\sigma^j|$, where $\lambda_\sigma = \{\lambda_\sigma^1, \lambda_\sigma^2, \cdots, \lambda_\sigma^n\}$ is the set of eigenvalues for A_σ mode. According to characteristics of subsystem A_σ, there may exist a surge or an elevation in the state trajectory. In Fig. 3.1, we show one possibility where the switching is necessary for better performance of the system. Solid line represents the state trajectory of A_1 while dashed line shows that of A_2. In contrast to A_2, which has slow convergence rate with no surge, A_1 reaches the origin faster with a surge. Therefore, for better performance, it is clear that A_2 mode has to be used from the beginning,

and then system has to switch to A_1 mode at time t_k as described in arrows in Fig. 3.1.

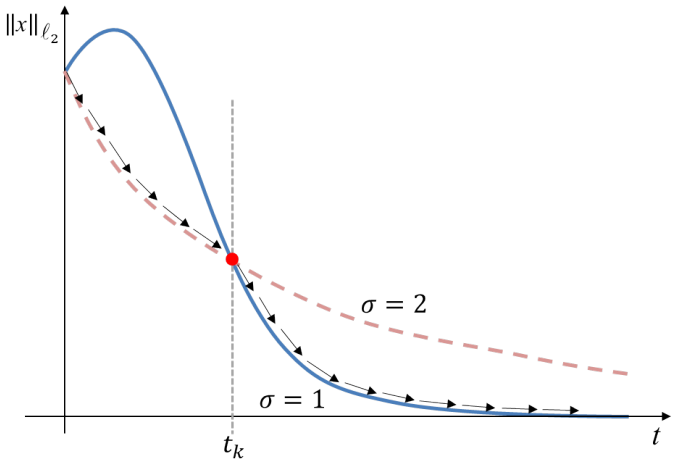


Figure 3.1: Schematic of optimal switching for the switched system.

In this chapter, motivated by the above example we address the following two questions.

1. Is there a switching sequence for a jump linear system to get better performance compared to single mode?

2. If the above holds true, can we find the optimal switching sequence?

In general, it is difficult to answer the first question directly. Instead, we want to show the case where the switching synthesis is not required because single mode attains the best performance. When $\rho(A_1) < \rho(A_2)$, A_1 mode has faster convergence to the origin than A_2 mode. In addition, if $\|A_1 x(k)\| < \|A_2 x(k)\|$ for all k, then $\|x(k)\|$ using A_1 mode is always less than $\|x(k)\|$ using A_2 mode. As a result, A_1 mode attains the best performance and jump is not necessary.

For the second question, which is the main contribution of this chapter, we introduce the optimal switching sequence using receding horizon framework and it is explained in section IV. Since, in most cases, initial condition of system state contains uncertainties, which come from measurement errors or sensor inaccuracies, we will use probability for initial state uncertainty of the system. Moreover, we assume that the type of initial state uncertainties is given by Gaussian distribution. The deterministic single initial state is a special case for Gaussian distribution with zero covariance. Therefore, in this chapter we conceptually cover much broader one. Due to this Gaussian PDF, system states become a random number, and hence we cannot use ℓ_2-norm for the performance measure. As a consequence, we need to adopt a proper metric to quantify the distance between PDFs to measure the performance. For this reason, instead of using ℓ_2-norm $\|\cdot\|_{\ell_2}$, Wasserstein distance is used as a tool for measuring the performance of jump linear systems. Brief explanations of Wasserstein distance are introduced in the next section.

3.3 Wasserstein Distance

Definition 3.1 *(Wasserstein distance) Consider the metric space $\ell_2(\mathbb{R}^n)$ and let the vectors $x_1, x_2 \in \mathbb{R}^n$. Let $\mathcal{P}_2(\varsigma_1, \varsigma_2)$ denote the collection of all probability measures ς supported on the product space $\mathbb{R}^{2n}$, having finite second moment, with first marginal ς_1 and second marginal ς_2. Then the L_2 Wasserstein distance of order 2, denoted as ${}_2W_2$, between two probability measures ς_1, ς_2, is defined as*

$$
\begin{aligned}
&{}_2W_2(\varsigma_1, \varsigma_2) \triangleq \\
&\left(\inf_{\varsigma \in \mathcal{P}_2(\varsigma_1, \varsigma_2)} \int_{\mathbb{R}^{2n}} \| x_1 - x_2 \|^2_{\ell_2(\mathbb{R}^n)} \; d\varsigma(x_1, x_2) \right)^{\frac{1}{2}}.
\end{aligned}
\tag{3.4}
$$

Remark 3.1 *Intuitively, Wasserstein distance equals the* least amount of work *needed*

to morph one distributional shape to the other, and can be interpreted as the cost for Monge-Kantorovich optimal transportation plan [93]. For notational ease, we henceforth denote ${}_2W_2$ as W. Further, one can prove (p. 208, [93]) that W defines a metric on the manifold of PDFs.

Next, we present new results for stability in terms of W.

Proposition 3.1 *If we fix Dirac distribution as the reference measure, then distributional convergence in Wasserstein metric is* necessary and sufficient *for convergence in m.s. sense.*

Proof Consider a sequence of n-dimensional joint PDFs $\{\varsigma_j(x)\}_{j=1}^{\infty}$, that converges to $\delta(x)$ in distribution, i.e., $\lim_{j\to\infty} W(\varsigma_j(x), \delta(x)) = 0 = \lim_{j\to\infty} W^2(\varsigma_j(x), \delta(x))$. From (3.4), we have

$$\begin{aligned} W^2(\varsigma_j(x), \delta(x)) &= \inf_{\varsigma \in \mathcal{P}_2(\varsigma_j(x), \delta(x))} \mathbb{E}\left[\| X_j - 0 \|^2_{\ell_2(\mathbb{R}^n)}\right] \\ &= \mathbb{E}\left[\| X_j \|^2_{\ell_2(\mathbb{R}^n)}\right] \end{aligned} \tag{3.5}$$

where the random variable $X_j \sim \varsigma_j(x)$, and the last equality follows from the fact that $\mathcal{P}_2(\varsigma_j(x), \delta(x)) = \{\varsigma_j(x)\}\ \forall\, j$, thus obviating the infimum. From (3.5), $\lim_{j\to\infty} W(\varsigma_j(x), \delta(x)) = 0 \Rightarrow \lim_{j\to\infty} \mathbb{E}\left[\| X_j \|^2_{\ell_2}\right] = 0$, establishing distributional convergence to $\delta(x) \Rightarrow$ m.s. convergence. Conversely, m.s. convergence $\Rightarrow$ distributional convergence, is well-known [44] and unlike the other direction, holds for arbitrary reference measure. □

Proposition 3.2 *(W^2 between Gaussian and Dirac PDF (see e.g., p. 160-161, [46])) The Wasserstein distance between Gaussian and Dirac PDF supported on $\mathbb{R}^n$,*

with respective joint PDFs $\varsigma = \mathcal{N}(\mu, \Sigma)$ *and* $\delta(x) = \lim_{\mu,\Sigma\to 0} \mathcal{N}(\mu, \Sigma)$*, is given by,*

$$W^2(\mathcal{N}(\mu, \Sigma), \delta(x)) = \| \mu \|^2_{\ell_2(\mathbb{R}^n)} + tr(\Sigma). \tag{3.6}$$

3.4 Switching Synthesis using Receding Horizon Framework with Wasserstein Metric

3.4.1 Optimal Switching Problem

W^2 defined in (3.6) represents square Wasserstein distance at fixed time. However, because the state PDF changes over time along dynamics, W^2 also changes as time goes. The following proposition expresses time-varying square W distance between $\mathcal{N}(\mu, \Sigma)$ and $\delta(x)$ at time k.

Proposition 3.3 *Let* $W^2(k)$ *denote square Wasserstein distance between* $\mathcal{N}(\mu, \Sigma)$ *and* $\delta(x)$ *at time k. Then* W^2 *distance at time k is given by*

$$W^2(k) = vec(I_n) \prod_{p=1}^{k} \left(A_{\sigma_p} \otimes A_{\sigma_p}\right) vec\left(\mu_0 \mu_0^\top + \Sigma_0\right) \tag{3.7}$$

where μ_0 *and* σ_0 *are mean and covariance of initial Gaussian PDF.*

Proof From (3.6), W^2 at time $k+1$ is defined as

$$W^2(k+1) = \| \mu(k+1) \|^2 + \text{tr}\left(\Sigma(k+1)\right) \tag{3.8}$$

$$= \text{tr}\left(\mu(k+1)\mu(k+1)^\top + \Sigma(k+1)\right) \tag{3.9}$$

Note that $\mathcal{N}(\mu(k), \Sigma(k))$ remains Gaussian PDF for all time k, even in the mode switching between sublinear dynamics. The following property are used for updating

mean and covaraince of Gaussian PDF.

$$\mu(k+1) = A_{\sigma_k}\mu(k) \tag{3.10}$$

$$\Sigma(k+1) = A_{\sigma_k}\Sigma(k)A_{\sigma_k}^{\top} \tag{3.11}$$

Substituting (3.10) and (3.11) into (3.9), we get

$$W^2(k+1) = \mathrm{tr}\left(A_{\sigma_k}^{\top}A_{\sigma_k}\left(\mu(k)\mu(k)^{\top} + \Sigma(k)\right)\right) \tag{3.12}$$

Using $\mathrm{tr}(X^{\top}Y) = vec(X)^{\top}vec(Y)$, (3.12) can be expressed as

$$W^2(k+1) = vec(A_{\sigma_k}^{\top}I_nA_{\sigma_k})^{\top}vec\left(\mu(k)\mu(k)^{\top} + \Sigma(k)\right) \tag{3.13}$$

Further, by applying $vec(ABC) = (C^{\top} \otimes A)vec(B)$ to the first term of right hand side in (3.13), we get

$$\begin{aligned} W^2(k+1) = vec(I_n)^{\top}&\left(A_{\sigma_k} \otimes A_{\sigma_k}\right) \\ &vec\left(\mu(k)\mu(k)^{\top} + \Sigma(k)\right) \end{aligned} \tag{3.14}$$

Similarly, W^2 at time k is also obtained as

$$W^2(k) = vec(I_n)^{\top}vec\left(\mu(k)\mu(k)^{\top} + \Sigma(k)\right) \tag{3.15}$$

From (3.14) and (3.15), and by induction, we conclude that $W^2(k)$ can be ex-

pressed in terms of initial mean and covariance as follows.

$$W^2(k) = vec(I_n)^\top \prod_{p=1}^{k} \left(A_{\sigma_p} \otimes A_{\sigma_p}\right) vec\left(\mu_0\mu_0^\top + \Sigma_0)\right) \tag{3.16}$$

□

We aim to find the switching sequence which guarantees the optimality of the system performance. One way of doing that is to minimize the area of Wasserstein distance, and hence minimize the time for the state PDF $\mathcal{N}\left(\mu(k), \Sigma(k)\right)$ to reach the reference PDF $\delta(x)$. In this case, we can formulate the cost function as

$$J(\sigma) = \int_0^\infty W^2 dt = \sum_{k=0}^{\infty} W^2(k) dk \tag{3.17}$$

where dk is a sampling time for discrete-time system. We use discrete-time W^2, and hence equality between second and last equations in (3.17) holds. From the cost function in (3.17), the optimal switching problem is defined as follows.

Optimal Switching Problem

$$J(\sigma^*) = \min_{\sigma} J(\sigma) \tag{3.18}$$

The solution of the above optimal switching problem can be obtained by finding optimal switching sequence $\sigma^* = \{\sigma_1^*, \sigma_2^*, \cdots\}$ out of all switching possibilities. For example, if the terminal time is finite and is set to be n instead of ∞ in (3.18), we have to check total m^n switching sequences for optimal solution, where m is total number of modes. Therefore, this problem is same as a conventional tree-search problem [20]. Since the growth of tree size is exponential in time, this problem is extremely difficult to solve and it requires large computational time. More details

with respect to issues on complexity are discussed in the last subsection. Therefore, we want to simplify the original problem by the next assumption.

Assumption 3.1 *For the jump linear system in* (3.2)*, switching sequence* σ *is constant over given horizon* T*.*

Using assumption 3.1, we can apply the receding horizon framework and the cost function over horizon length T can be defined as

$$J = \sum_{k=t_j}^{T+t_j} W^2(k)dk \tag{3.19}$$

$$= \sum_{k=t_j}^{T+t_j} vec(I_n) \left(\prod_{p=1}^{k} \left(A_{\sigma_p} \otimes A_{\sigma_p} \right) \right) vec(\mu_0 \mu_0^\top + \Sigma_0)dk \tag{3.20}$$

$$= \sum_{k=t_j}^{T+t_j} vec(I_n) \left(\left(A_\sigma \otimes A_\sigma \right)^k \right) vec(\mu_0 \mu_0^\top + \Sigma_0)dk \tag{3.21}$$

$$= \sum_{k=t_j}^{T+t_j} W_\sigma^2(k)dk \tag{3.22}$$

Switching sequence, denoted as σ, is fixed and we get (3.21) from (3.20) for $\sigma_p = \sigma =$ constant under the assumption 3.1.

Then, the optimal cost-to-go function is defined as:

Optimal Switching with Receding Horizon

$$J^* = \min_\sigma \left(\sum_{k=t_j}^{T+t_j} W_\sigma^2(k)dk \right) \tag{3.23}$$

$$s.t. \quad W_\sigma^2(t_{j+1}) - W_\sigma^2(t_{j-1}) \leq -\epsilon_\sigma(t_j) \tag{3.24}$$

where $\epsilon_\sigma(\cdot)$ is a positive definite function and the constraint (3.24) is enforced for stability.

It is well known [66] that switching between individually stable modes can make a stable system unstable. Therefore, the constraint (3.24) should be enforced to ensure stability. Fig. 3.2 shows schematic of optimal switching sequence using receding horizon framework. At time t_j, the solution of (3.23)-(3.24) provides the optimal switching sequence for horizon T and there is no switching during time $k \in [t_j, t_{j+1})$. When time k reaches t_{j+1}, we again compute optimal switching for next horizon T.

Note that although (3.24) implies piecewise monotone decreasing in W^2, it is not so restrictive condition because (3.24) is only applied to the time t_j at which jump occurs. In other words, $W^2(\cdot)$ can increase in between times $k \in [t_j, t_{j+1})$ as depicted in Fig.3.2.

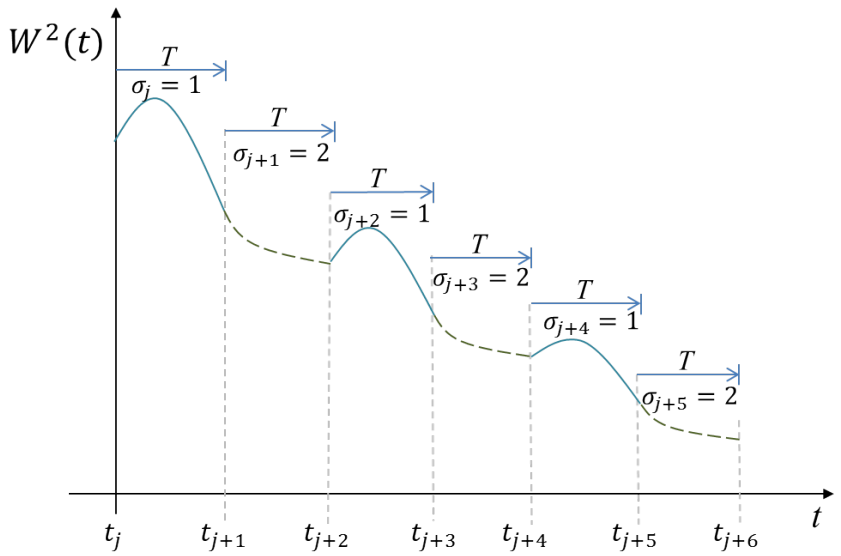


Figure 3.2: Optimal switching strategy for the switched linear system using receding horizon framework.

3.4.2 Stability Issues

The reason we choose time t_{j-1} and t_{j+1} for piecewise monotone decreasing condition in (3.24) is as follows. Switching takes place at every time instance t_j. Between time $k \in [t_{j-1}, t_j)$, there is no switching. At the end of the horizon T, which is at time t_j, we can compute the next optimal switching sequence for time $k \in [t_j, t_{j+1})$ using (3.23)-(3.24). Since the individual subsystem is Schur stable, there is no stability problem if there is no switching. However, if jump occurs, there may be a bump in the state trajectory, and hence in W^2 right after the switching. This may cause instability of the jump linear system. Therefore, the constraint (3.24) which is sufficient condition for the stability should be enforced. The following lemma and theorem prove the stability of jump linear systems in the context of mean square sense under the receding horizon framework.

Lemma 3.1 *For jump linear systems with the receding horizon framework* (3.23)*,* $W^2(t_j)$ *converges to zero under the constraint* (3.24)*, where* t_j *is jump time.*

Proof For piecewise monotone decreasing sequence $W^2(\cdot)$, $\exists n_0 \in \mathbb{Z}^+$ such that $W^2(t_{n_0}) < N$ and N is any arbitrary positive real number $\mathbb{R}^+$. By the monotone decreasing condition above, for all $n > n_0$, $W^2(t_n) < N$. Since N is any arbitrary positive real number $\mathbb{R}^+$ and the lower bound of $W^2(\cdot)$ is 0, $W^2(t_j) \to 0$ as $j \to \infty$.

□

Lemma 3.1 proves piecewise convergence of W^2 under the constraint given in (3.24). Although $W^2(t_j)$ converges to zero, it does not necessarily guarantee no oscillation at time $k \in [t_j, t_{j+1})$. Therefore, we have to show that if $W^2(t_j) \to 0$, then $W^2(k)$ is also zero for all $t \in [t_j, t_{j+1})$. The following lemma proves the above argument.

Lemma 3.2 *Once $W^2(t_j) = 0$ at time t_j, then $W^2(k)$ is always zero for all $k \geq t_j$.*

Proof From (3.6), in order for $W^2(t_j)$ to be zero, both mean and covariance have to be zero. According to (3.10) and (3.11), used for updating mean and covariance, they remain zero for all $k \geq t_j$ once they become zero. □

Using Lemma 3.1 and Lemma 3.2, following theorem shows the m.s stability of jump system under the proposed switching policy.

Theorem 3.1 *Jump linear systems in* (3.3) *under the receding horizon framework* (3.23)-(3.24) *is m.s. stable.*

Proof By Proposition 3.1, the system is m.s. stable if and only if $W(\cdot) = 0$. From Lemma 3.1 and Lemma 3.2, it is shown that $W^2(\cdot)$ converges to zero, and hence $W(\cdot)$ also converges to zero. Therefore, jump linear system in (3.3) is m.s. stable. □

3.4.3 Horizon Length Issues

Primbs *et al.* [82] have shown the unified framework between pointwise min-norm($T = 0$), optimality($T = \infty$), and receding horizon T. The horizon length T can vary according to available time for online computation and in general, we can attain better results for longer horizon length T. However, unlike receding horizon control, longer horizon length T for optimal switching does not imply better performance of jump systems. The effect of different receding horizon length T in optimal switching can be analysed as follows.

1. *Pointwise minimum ($T = 0$)*: When $T = 0$, the solution of optimal switching problem is obtained by solving (3.23)-(3.24) with $T = 0$. This is equivalent to finding pointwise minimum of $W^2_{\sigma_k}(k)$ at every time k. However, since there is no prediction for the future behaviour of the system, pointwise minimum does

not guarantee the optimal switching of jump systems. Therefore, it may cause worse performance than good or even poor controller itself without switching.

2. *Infinite horizon ($T = \infty$)*: In case of infinite horizon, the optimal switching problem is trivial. By assumption 3.1, switching does not occur over infinite horizon. Therefore, the solution of the optimal switching is to choose single mode which achieves the minimum area of W^2 from time $k = 0$ to ∞.

From the above fact, receding horizon length T should be $0 < T < \infty$. However, there is no guideline for the optimal horizon length T. One necessary condition for T is that it has to be chosen to satisfy the stability constraint in (3.24). For instance, if jump occurs at time t_j and as a result there might be a bump right after the switching, then the constraint(3.24) may not be satisfied for short T. Therefore, we can set the receding horizon length T as follows.

Theorem 3.2 *For optimal switching problem with receding horizon framework in* (3.23)-(3.24)*, the receding horizon length T has to be set to satisfy stability constraint* (3.24) *and such that,*

$$T \geq \tau_j := t_{j+1} - t_{j-1} \tag{3.25}$$

where τ_j is updating time interval for receding horizon, and there always exist τ_j satisfies stability constraint (3.24) *under the assumption that each dynamics is Schur stable.*

Proof From Assumption 3.1 and that individual systems are Schur stable, there is no switching in fixed horizon T. For linearly stable system, which is globally uniformly asymptotically stable, there exists time t_{j+1} such that $\|x(t_{j+1})\| < \|x(t_{j-1})\|$ for $t_{j+1} > t_{j-1}$. By taking square and expectation for both side of above equation,

we get $W^2(t_{j+1}) < W^2(t_{j-1})$. Therefore, stability constraint (3.24) is satisfied with some positive definite function ϵ_j. □

Note that the horizon length T is not necessarily to be constant. For each different jump time t_j we can set a different horizon length T, satisfying the condition given in Theorem 3.2.

3.4.4 Complexity Issues

Two problems associated with the original optimal switching problem (3.18) give rise to complexity issues. First, infinite time causes infinite size in total possible numbers of switching. Second, even if the switching is finite and hence (3.18) is equivalent to tree-search problem [20], the computational complexity to solve this problem is NP-complete [3].

However, the optimal switching with receding horizon framework in (3.23)-(3.24) enable us to simplify the problem. Once the horizon length T satisfying (3.24) is obtained, then the solution of optimal switching problem is same with choosing $\min\{W_1^2, W_2^2, \cdots, W_m^2\}$, where m is total number of modes. Hence, this is same with sorting problems, where computational complexity is $O(n \log n)$ in general. As a consequence, optimal switching with receding horizon can be solved fast enough for online computation.

3.5 Examples

3.5.1 Jump Linear System with Five Different Modes Dynamics

Consider a following discrete-time jump linear system.

$$x(k+1) = A_\sigma x(k), \qquad \sigma \in \{1, 2, \ldots, 5\}$$

This system has a five different mode dynamics given by

$$
\begin{aligned}
A_1 &= \begin{bmatrix} 1.01 & -0.17 \\ 0.32 & -0.48 \end{bmatrix}, \\
A_2 &= \begin{bmatrix} 0.06 & 0.80 \\ 0.01 & -0.77 \end{bmatrix}, \\
A_3 &= \begin{bmatrix} 0.72 & 0.48 \\ 0 & 0.55 \end{bmatrix}, \\
A_4 &= \begin{bmatrix} -0.33 & -0.65 \\ -0.46 & 0.69 \end{bmatrix}, \\
A_5 &= \begin{bmatrix} -0.13 & 0.12 \\ -1.33 & -1.05 \end{bmatrix}
\end{aligned}
$$

In addition, we assume that initial state has an uncertainty represented by Gaussian PDF with mean μ_0 and covariance Σ_0 as follows.

$$
\mu_0 = [5, 5]^\top, \qquad \Sigma_0 = \begin{bmatrix} 2.25 & 0 \\ 0 & 2.25 \end{bmatrix}
$$

.

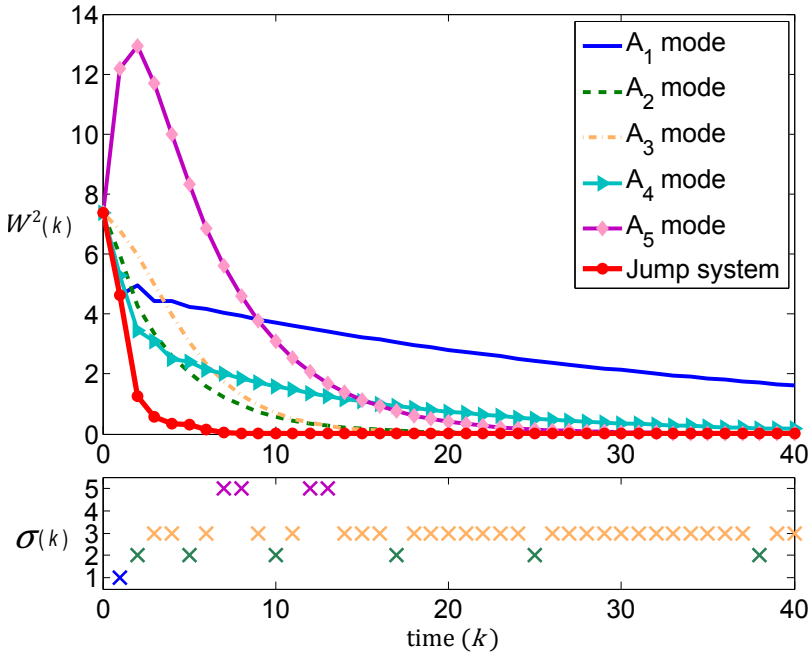
For this system with given Gaussian initial state PDF, we aim to design a switching sequence that attains the optimality of the system performance. The simulation results are shown in Fig. 3.3(a). The cross mark represents the mode that is used at a specific switching sequence. According to this result, jump system shows the fastest convergence to the origin under the proposed receding horizon framework. The spectral radius for individual mode dynamics are $\rho(A_1) = 0.97$, $\rho(A_2) = 0.78$,

$\rho(A_3) = 0.72$, $\rho(A_4) = 0.93$, and $\rho(A_5) = 0.82$. From this, we know that A_3 dynamics converges to the origin faster than other mode dynamics. However, since other mode dynamics shows good performance in the beginning, it is desirable to use that mode dynamics initially. This intuition coincide with the optimal switching results as shown in Fig.3.3(a). Additionally, total W^2 area that stands for the system performance, is depicted in Fig.3.3(b) to compare the performance between different mode dynamics and a jump system. In Fig.3.3(b), A_2 mode shows the minimal W^2 area between individual dynamics without switching. The jump system using optimal switching synthesis shows about 3.5 times less W^2 area compared to A_2 mode that attains the best performance between the individual mode dynamics. In this example, the optimal switching synthesis provided in this chapter shows the best performance and beats any other mode dynamics without switching.

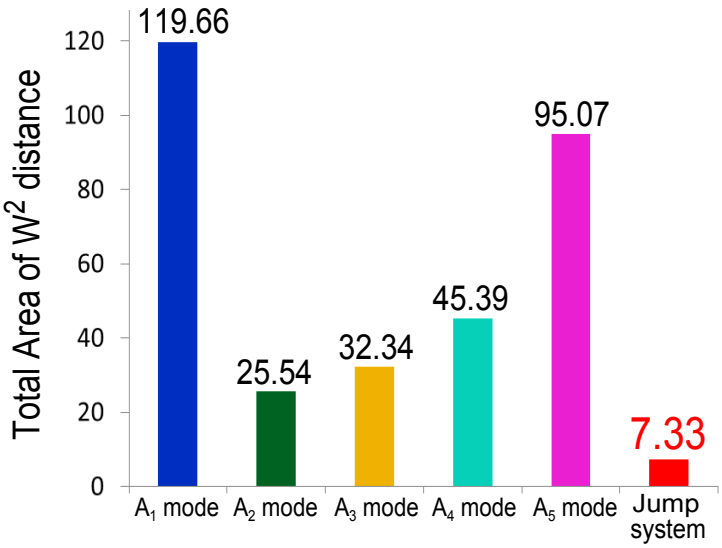
3.5.2 Linearized Quadrotor Dynamics with Two Controllers

Here we consider 6-state linearized nonlinear quadrotor dynamics. The first controller (C_{High}) provides higher performance by commanding aggressive control actions and is designed using full-state feedback. The second controller is a lead-lag compensator (C_{Low}) which provides poorer performance by commanding less aggressive control actions. Implementation of C_{High} requires more computational time and consumes more energy (batttery) and C_{Low} is resource economical in terms of both CPU time and energy usage. More details about this controller can be found in [54]. In this example, we want to design the optimal switching sequence using both C_{high} and C_{low} to obtain better performance.

The states of the quadrotor are $x = [\phi, \theta, \psi, p, q, r]^\top$ and nonlinear dynamics is



(a) W^2 distance and optimal switching sequence σ



(b) Total area of W^2 for each mode and Jump system

Figure 3.3: Simulation results of optimal switching synthesis for the switched system with 5 different modes.

given by

$$\dot{p} = \frac{qr(I_{yy} - I_{zz}) + qJ_r\Omega_r + bl(-\Omega_2^2 + \Omega_4^2)}{I_{xx}},$$
$$\dot{q} = \frac{pr(I_{zz} - I_{xx}) - pJ_r\Omega_r + bl(\Omega_1^2 - \Omega_3^2)}{I_{yy}},$$
$$\dot{r} = \frac{pq(I_{xx} - I_{yy}) + d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)}{I_{zz}},$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix},$$

where symbols are defined in Table 3.1.

Table 3.1: Nomenclature for Quadrotor Dynamics

Symbol	definition	Symbol	definition
ϕ	roll angle	p	roll rate
θ	pitch angle	q	pitch rate
ψ	yaw angle	r	yaw rate
$I_{xx,yy,zz}$	body inertia	J_r	rotor inertia
b	thrust factor	d	drag factor
l	lever	Ω_r	rotor speed

Linearized quadrotor dynamics is obtained by linearizing the nonlinear equations of motion about hover. Two continuous-time closed-loop systems A_1 and A_2 are discretized with sampling time 0.01s. The switching policy determines the sequence for σ, which is deterministic.

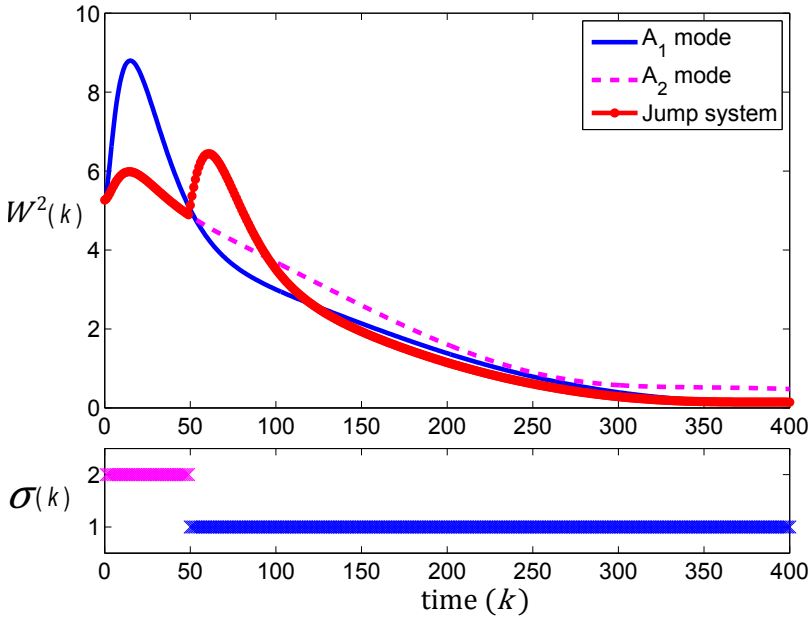
The initial condition uncertainty is assessed with respect to initial condition uncertainty given by Gaussian PDF $\mathcal{N}(\mu_0, \Sigma_0)$, with $\mu_0 = [0.5, -1.5, -5, 0.1, 0.2, 0.1]^\top$

and $\Sigma_0 = 0.0225 \times I_{6\times 6}$, where $I_{6\times 6}$ is the 6×6 identity matrix. The control objective is to maintain hover, which corresponds to equilibrium state $x_{eq} = [0, 0, 0, 0, 0, 0]^\top$.

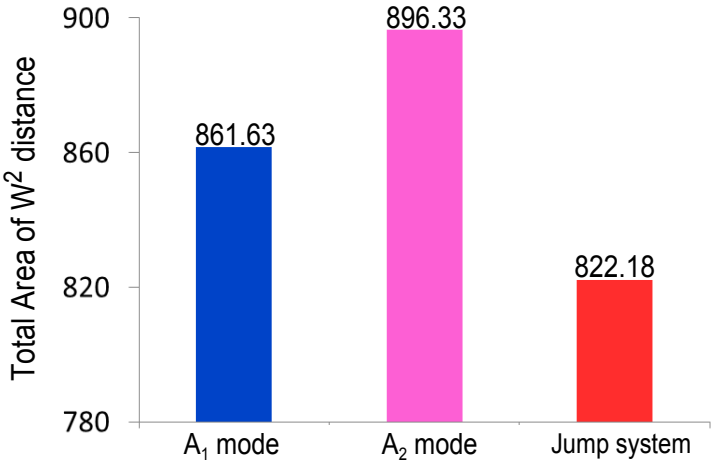
Fig. 3.4(a) and 3.4(b) present the result of switching synthesis using proposed method in this chapter. From the beginning in Fig. 3.4(a), A_1 dynamics shows large elevation in W^2 distance while A_2 does not. As a result, the optimal switching with receding horizon selects A_2 dynamics. However, after $k = 50$, an optimality is obtained by switching to A_1 via optimal switching synthesis proposed in this chapter. Fig. 3.4(b) presents the performance of each mode in terms of total W^2 area. It is clear that the lowest area, which is the best performance, can be attained by switching.

3.6 Concluding Remarks

In this chapter, we proposed the optimal switching synthesis for jump linear systems with Gaussian initial state uncertainty. The Wasserstein metric that defines a distance between PDFs was adopted to measure both the performance and the stability of the jump linear system. We showed that the optimality of the system performance can be obtained by synthesizing switching laws via minimization of objective function expressed in terms of Wasserstein distance. Also, the mean square stability of the jump linear system was guaranteed under the proposed switching synthesis. The efficiency and the usefulness of the proposed methods were demonstrated by examples.



(a) W^2 distance and optimal switching sequence σ



(b) Total area of W^2 for each mode and Jump system

Figure 3.4: Simulation results of optimal switching synthesis for linearized quadrotor dynamics.

4. ANALYSIS OF MASSIVELY PARALLEL ASYNCHRONOUS NUMERICAL ALGORITHMS

4.1 Introduction

Exascale computing systems will soon be available to study computation intensive applications such as multi-physics multi-scale simulations of natural and engineering systems. Many scientific and practical problems can be described very accurately by ordinary or partial differential equations which may be tightly coupled with long-range correlations. These exascale systems may have $O(10^5 - 10^6)$ processors ranging from multicore processors to symmetric multiprocessors [30], [79], [77]. Furthermore, such systems are likely to be heterogeneous using both heavily multi-threaded CPUs as well as GPUs. Many challenges must be overcome before exascale systems can be utilized effectively in such applications. One such obstacle is the communication in tightly coupled problems during parallel implementation of any iterative numerical algorithm. This communication requires massive data movement in turn leading to idle time as the cores need to be synchronized after each time step.

Recent literature has proposed relaxing these synchronization requirements across the PEs [25]. This potentially eliminates the overhead associated with extreme parallelism and significantly reduces computational time. However, the price to pay is loss of predictability possibly resulting in calculation errors. Thus, a rigorous analysis of the tradeoff between speed and accuracy is critical. This chapter presents a framework for quantifying this tradeoff by analyzing the asynchronous numerical algorithm as a switched dynamical system [19], [69], [59], [47], [97], [104], [73], [63], [64], [61]. While there is a large literature for analysis of such systems, these techniques are not applicable to our application. The reason is that due to the large number of PEs,

the switched system model has an extremely large number of modes, which makes the available analysis tools intractable. Key contributions in this chapter include new techniques for a) *stability analysis*, or quantification of steady-state error with respect to the synchronous solution; b) *convergence rate analysis* of the expected value of this error; and c) *probabilistic bounds* on this error. These techniques are developed to be computationally efficient, and avoid the aforementioned scalability issue.

The chapter is organized as follows. Section 4.2 addresses the problems for the asynchronous numerical algorithm. In section 4.3, we introduce a switched system framework to model the system structure for the asynchronous numerical scheme. The stability results are presented in section 4.4, and section 4.5 shows the convergence rate analysis. Then, the error analysis in probability is developed in section 4.6. Section 4.7 demonstrates the usefulness of the proposed method by examples. Finally, section 4.8 concludes this chapter.

4.2 Problem Formulation

Notation: The symbol $||\cdot||$ and $||\cdot||_\infty$ stand for the Euclidean and infinity norm, respectively. The set of positive integers are denoted by $\mathbb{N}$. Further, $\mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}$. Also, $\lambda(\cdot)$ represents an eigenvalue of a square matrix. In particular, $\lambda_{max}(\cdot)$ and $\lambda_{min}(\cdot)$ denote the largest and the smallest eigenvalue in magnitude, respectively. The symbols $\otimes$, $\det(\cdot)$, $\mathrm{tr}(\cdot)$, and $\mathrm{vec}(\cdot)$ denote Kronecker product, matrix determinant, trace operator, and vectorization operator, respectively. Finally, the symbol $\mathbf{Pr}(\cdot)$ stands for the probability.

In this chapter we demonstrate our framework and techniques on the one-dimensional

heat equation, given by

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad t \geq 0, \tag{4.1}$$

where u is the time and space-varying state of the temperature, and t and x are continuous time and space respectively. The constant $\alpha > 0$ is the thermal diffusivity of the given material.

The PDE is solved numerically using the finite difference method by Euler explicit scheme, with a forward difference in time and a central difference in space. Thus (4.1) is approximated as

$$\frac{u_i(k+1) - u_i(k)}{\Delta t} = \alpha \left(\frac{u_{i+1}(k) - 2u_i(k) + u_{i-1}(k)}{\Delta x^2} \right), \tag{4.2}$$

where $k \in \mathbb{N}_0$ is the discrete-time index and u_i is the temperature value at i^{th} grid space point. The symbols Δt and Δx denote the sampling time and the grid resolution in space, respectively. Further, if we define a constant $r \triangleq \alpha \frac{\Delta t}{\Delta x^2}$, then (4.2) can be written as

$$u_i(k+1) = r u_{i+1}(k) + (1 - 2r) u_i(k) + r u_{i-1}(k), \tag{4.3}$$

It is important to observe that (4.3) is a discrete-time linear dynamical system.

Fig. 4.1 illustrates the numerical scheme over the discretized 1D spatial domain. A typical *synchronous* parallel implementation of this numerical scheme assigns several of these grid points to each PE. The updates for the temperature at the grid points assigned to each PE, occur in parallel. However, at every time step k, the data associated with the boundary grid points, where the communication is necessary are synchronized, and used to compute $u_i(k+1)$. This synchronization across PEs is

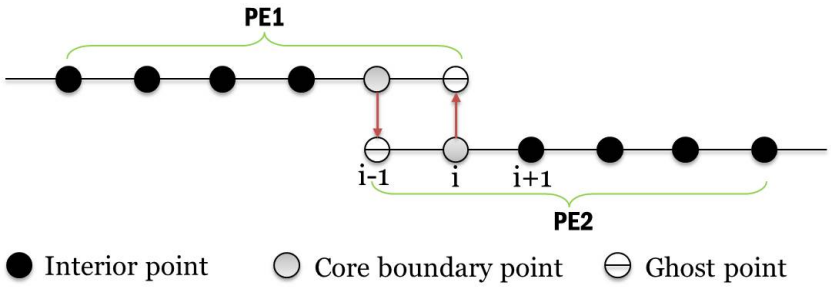


Figure 4.1: Discretized one-dimensional domain with an asynchronous numerical algorithm. the PE denotes a group of grid points, assigned to each core.

slow, especially for massively parallel systems (estimates of idle time due to this synchronization give figures of up to 80% of the total time taken for the simulation as idle time). Recently, an alternative implementation which is *asynchronous* has been proposed. In this implementation, the updates in a PE occur without waiting for the other PEs to finish and their results to be synchronized. The data update across PEs occurs sporadically and independently. This asynchrony directly affects the update equation for the boundary points, as they depend on the grid points across PEs. For these points, the update is performed with the most recent available value, typically stored in a buffer. The effect of this asynchrony then propagates to other grid points. Within a PE, we assume there is no asynchrony and data is available in a common memory.

Thus, the asynchronous numerical scheme corresponding to (4.3) is given by

$$u_i(k+1) = ru_{i+1}(k^*_{i+1}) + (1-2r)u_i(k) + ru_{i-1}(k^*_{i-1}), \tag{4.4}$$

where $k^*_i \in \{k, k-1, k-2, \ldots, k-q+1\}$, $i = 1, 2, \ldots, N$, denotes the randomness caused by communication delays between PEs. The subscript i in k^*_i depicts that

each grid space point may have different time delays. The parameter q is the length of a buffer that every core maintains to store data transmitted from the other cores. In this chapter, we treat k_i^* as a random variable and thus (4.4) can be considered to be a linear discrete-time dynamical system with stochastic updates.

Although (4.4) is derived for the 1D heat equation, the treatment above can be developed for any parabolic PDEs. This observation encourages us to consider using tools from dynamical systems to analyze the effect of asynchrony in parallel numerical algorithms. Therefore, the primary goal of this study is to investigate the *stability*, *convergence rate*, and *error probability* of the asynchronous numerical algorithm in the framework of stochastic switched dynamical systems.

4.3 A Switched System Approach

Let us define the state vector $U_j(k) \in \mathbb{R}^n \triangleq [u_1^j(k), u_2^j(k), \ldots, u_n^j(k)]^\top$, where $u_i^j(k)$ stands for the i^{th} grid space point in the j^{th} PE and n is the total number of grid points in the j^{th} PE. Therefore, (4.3) can be compactly written as

$$U(k+1) = AU(k), \quad k \in \mathbb{N}_0,$$

where $U(k) \in \mathbb{R}^{Nn} \triangleq [U_1(k)^\top, U_2(k)^\top, \ldots, U_N(k)^\top]^\top$, N is the total number of PEs, n is the size of the state for each PE, and system matrix $A \in \mathbb{R}^{Nn \times Nn}$ is given by

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ r & 1\text{-}2r & r & 0 & \cdots & 0 \\ 0 & r & 1\text{-}2r & r & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & & r & 1\text{-}2r & r \\ 0 & \cdots & & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{Nn \times Nn}.$$

Note that the first and the last row of A matrix specify the Dirichlet boundary conditions (see pp. 150, [80]). i.e., we have the constant in time boundary temperatures for simplicity.

Next, we define the augmented state $X(k) \in \mathbb{R}^{Nnq} \triangleq [U(k)^\top, U(k-1)^\top, \ldots, U(k-q+1)^\top]^\top$, where, as stated before, q is the buffer length. For pedagogical simplicity (and without loss of generality), we consider the case with $q = 2$ and $N = 3$. Further, we let $n = 1$, which implies there is only one grid point in each PE. For this particular case, we construct the following matrices,

$$W_1 = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ r & \text{1-2}r & r & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline \multicolumn{3}{c:}{\underline{\mathrm{I}}} & \multicolumn{3}{c}{\underline{0}} \end{array}\right], \; W_2 = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{1-2}r & r & r & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline \multicolumn{3}{c:}{\underline{\mathrm{I}}} & \multicolumn{3}{c}{\underline{0}} \end{array}\right],$$

$$W_3 = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ r & \text{1-2}r & 0 & 0 & 0 & r \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline \multicolumn{3}{c:}{\underline{\mathrm{I}}} & \multicolumn{3}{c}{\underline{0}} \end{array}\right], \; W_4 = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{1-2}r & 0 & r & 0 & r \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline \multicolumn{3}{c:}{\underline{\mathrm{I}}} & \multicolumn{3}{c}{\underline{0}} \end{array}\right],$$

where $\underline{\mathrm{I}} \in \mathbb{R}^{Nn \times Nn}$ and $\underline{0} \in \mathbb{R}^{Nn \times Nn}$ are the identity and the zero matrices with appropriate dimensions. As in [25], we assume that the condition $0 < r \leq 0.5$ holds from now on. The asynchronous numerical scheme can then be written as a switched

system

$$X(k+1) = W_{\sigma_k} X(k), \quad \sigma_k \in \{1, 2, \ldots, m\},\ k \in \mathbb{N}_0, \tag{4.5}$$

where the matrices $W_{\sigma_k} \in \mathbb{R}^{Nnq \times Nnq}$, are the subsystem dynamics. In general, the total number of switching modes is $m = q^{2(N-2)}$ that is obtained by considering all cases to distribute every components r in W_1 matrix, where the number of r in W_1 is given by $2(N-2)$, into q numbers of zero block matrix as in the above example. Therefore, the number of modes increase exponentially with the number of PEs, which is quite large for massively parallel systems.

At every time step, the numerical scheme evolves using one of the m modes, which depends on the variable k_i^*. In this chapter, we model the variable k_i^* as a random variable that evolves in an independently and identically distributed (i.i.d.) fashion in time, and independently from one core to the next. Hence, we let π_j be the modal probability for W_j which is assumed to be stationary in time. Let $\Pi \triangleq \{\pi_1, \pi_2, \ldots, \pi_m\}$, be the switching probabilities such that $0 \leq \pi_j \leq 1$, $\forall j$ and $\sum_{j=1}^{m} \pi_j = 1$. The system in (4.5) is thus an i.i.d jump linear system, which is a simpler case of the more well-known Markovian jump linear systems [64]. Even though the analysis theory for such systems is well developed, the existing tools are not suitable for our application because of the extremely large number of modes, particularly when N is large. Thus, we now develop an analysis theory for the i.i.d. jump linear systems which scales better with respect to the number of modes.

4.4 Stability

The first requirement is that of convergence of (4.5). Because of the Dirichlet boundary conditions, we expect the temperature to converge to a constant value for every grid point. We proceed to analyze the conditions for convergence (or

stability) of the system. To this end, we may try to use the infinity norm and apply the sub-multiplicative property to obtain $||X(k+1)||_\infty = ||W_{\sigma_k} X(k)||_\infty \leq ||W_{\sigma_k}||_\infty ||X(k)||_\infty = ||X(k)||_\infty$, where the last equality holds since we have $||W_j||_\infty = 1$, $\forall j$. This can be written as

$$\frac{\| X(k+1) \|_\infty}{\| X(k) \|_\infty} \leq 1. \tag{4.6}$$

The above result only shows that the solution from the asynchronous algorithm is *marginally* stable and we are unable to determine the steady-state solution.

In fact, we can show that the asynchronous scheme also attains the same steady-state value as the synchronous scheme, regardless of the specific realization of $\{\sigma_k\}$. Using spectral decomposition, the matrices W_j can be expressed in terms of the eigenvalues and corresponding eigenvectors as

$$W_j \in \mathbb{R}^{Nnq \times Nnq} = \sum_{i=1}^{Nnq} \lambda_i^j v_i^j s_i^j, \quad j = \{1, 2, \ldots, m\}, \tag{4.7}$$

where $\lambda_i^j \in \mathbb{R}$, $v_i^j \in \mathbb{R}^{Nnq \times 1}$, and $s_i^j \in \mathbb{R}^{1 \times Nnq}$ denote the eigenvalues, right eigenvectors, and left eigenvectors of W_j, respectively.

Since $\max_i |\lambda_i^j| \leq ||W_j||_\infty = 1$, $\forall j$, the spectral radius of W_j, $j = 1, 2, \ldots, m$, is less than or equal to 1. Therefore, we may order the eigenvalues as $1 \geq |\lambda_1^j| \geq |\lambda_2^j| > \cdots \geq |\lambda_{Nnq}^j| \geq 0$. It can be shown that all W_j have two eigenvalues with value 1, irrespective of the size of q and N. Therefore, the eigenvalues for W_j are ordered as $1 = |\lambda_1^j| = |\lambda_2^j| > |\lambda_3^j| \geq \cdots \geq |\lambda_{Nnq}^j| \geq 0$.

Moreover, the left and right eigenvectors for eigenvalues equal to 1 are common eigenvectors for all matrices W_j, $j = 1, 2, \ldots, m$. These common left and right eigenvectors are

1) **Left eigenvectors:**

$$s_1 = [1, 0, \cdots, 0 \ , \ \boldsymbol{0} \ , \cdots , \ \boldsymbol{0} \ ] \in \mathbb{R}^{1\times Nnq}, \tag{4.8}$$

$$s_2 = [0, \cdots, 0, 1 \ , \ \boldsymbol{0} \ , \cdots , \ \boldsymbol{0} \ ] \in \mathbb{R}^{1\times Nnq}, \tag{4.9}$$

2) **Right eigenvectors:**

$$v_1 = [\mu_1, \mu_1, \cdots, \mu_1]^\top \in \mathbb{R}^{Nnq\times 1}, \tag{4.10}$$

$$v_2 = [\mu_2, \mu_2, \cdots, \mu_2]^\top \in \mathbb{R}^{Nnq\times 1}, \tag{4.11}$$

where $\boldsymbol{0} \in \mathbb{R}^{1\times Nn}$ denotes a row vector with all zero elements, and

$$\begin{aligned}
&\mu_1 \triangleq [1, \frac{Nn-2}{Nn-1}, \cdots, \frac{Nn-j}{Nn-1}, \cdots, \frac{1}{Nn-1}, 0] \in \mathbb{R}^{1\times Nn},\\
&\mu_2 \triangleq [0, \frac{1}{Nn-1}, \frac{2}{Nn-1}, \cdots, \frac{j-1}{Nn-1}, \cdots, \frac{Nn-2}{Nn-1}, 1] \in \mathbb{R}^{1\times Nn},\\
&j = 1, 2, \ldots, Nn.
\end{aligned}$$

Notice that we have $W_j v_i = v_i$ and $s_i W_j = s_i$, $i = 1, 2$, $\forall j$. Then, the steady-state value for the asynchronous scheme is given by the following result.

Proposition 4.1 *Consider the i.i.d. jump linear system in* (4.5) *with subsystem matrices* W_j, $j = 1, 2, \ldots, m$ *and a stationary switching probability* Π. *For a given initial condition* $X(0)$, *if we define* $\Psi \triangleq v_1 s_1 + v_2 s_2$, *where* v_i *and* s_i, $i = 1, 2$, *are given in* (4.8)–(4.11), *then, the steady-state value* X_{ss} *has the following form:*

$$X_{ss} \triangleq \lim_{k\to\infty} X(k) = \Psi X(0),$$

irrespective of the switching sequence $\{\sigma_k\}$.

Proof Let the eigenvalues of W_j be ordered in magnitude by $1 = |\lambda_1^j| = |\lambda_2^j| > |\lambda_3^j| \geq \cdots \geq |\lambda_{Nnq}^j| \geq 0$. Also, let v_i^j and s_i^j be the right and left eigenvector corresponding to λ_i^j, respectively. Using the spectral decomposition, W_j can be alternatively expressed by $W_j = \sum_{i=1}^{Nnq} \lambda_i^j v_i^j s_i^j = \Psi + \sum_{\lambda_i^j \neq 1} f^j(i)$, where $\Psi \triangleq v_1 s_1 + v_2 s_2$ and $f^j(i) \triangleq \lambda_i^j v_i^j s_i^j$.

Then, starting with $X(0)$, the realization of the switching sequence σ_k results in

$$\begin{aligned} X(k) &= W_{\sigma_{k-1}} W_{\sigma_{k-2}} \cdots W_{\sigma_1} W_{\sigma_0} X(0) \\ &= \Big(\Psi + \sum_{\lambda_i^{\sigma_{k-1}} \neq 1} f^{\sigma_{k-1}}(i)\Big) \cdots \Big(\Psi + \sum_{\lambda_i^{\sigma_0} \neq 1} f^{\sigma_0}(i)\Big) X(0) \\ &= \Big(\Psi^k + g(k)\Big) X(0), \end{aligned}$$

where in above equation, $g(k)$ represents all the other multiplication terms except Ψ^k term. Note that $g(k)$ is formed by the product of λ_i^j, where $0 \leq |\lambda_i^j| < 1$, $\forall i > 2$, $\forall j$. Consequently, if $k \to \infty$, then $g(k)$ is asymptotically convergent to zero since the *infinite* number of multiplication of the term λ_i^j, $\forall i > 2$, converges to zero. Therefore, we have

$$X_{ss} = \lim_{k \to \infty} X(k) = \lim_{k \to \infty} \Psi^k X(0) = \Psi X(0).$$

The last equality in above equation holds because $\Psi^k = \Psi^{k-1} = \cdots = \Psi$, $\forall k \in \mathbb{N}$. $\square$

4.5 Convergence Rate

In this section, we investigate how fast the expected value of the state converges to the steady-state X_{ss} by analyzing the transient behavior of the asynchronous algorithm. Let us define a new state variable $e(k) \triangleq X(k) - X_{ss}$. The expected value of $e(k)$ is given by $\bar{e}(k) \triangleq \mathbb{E}[X(k) - X_{ss}] = \mathbb{E}[X(k)] - X_{ss} = \bar{X}(k) - X_{ss}$, where $\bar{X}(k) \triangleq \mathbb{E}[X(k)]$. Therefore, the convergence rate of $||\bar{e}(k)||$ will provide bound for

the convergence rate of $||\bar{X}(k) - X_{ss}||$.

To obtain an upper bound for the convergence rate of $||\bar{e}(k)||$, we use the following matrix transformation. As described in (4.7), each modal matrix W_j can be alternatively expressed by $W_j = \sum_{i=1}^{Nnq} \lambda_i^j v_i^j s_i^j$, where λ_i^j, v_i^j, and s_i^j denote the eigenvalues, right and, respectively, left eigenvectors for W_j. If we define the transformed matrix $\tilde{W}_j \triangleq W_j - \sum_{\lambda_i^1=1} \lambda_i^j v_i^j s_i^j = W_j - \Psi = \sum_{\lambda_i^j \neq 1} \lambda_i^j v_i^j s_i^j$, then the modal dynamics with the corresponding state $e_j(k)$, is given by

$$e_j(k+1) = \tilde{W}_j e_j(k), \quad j = \{1, 2, \ldots, m\},\ k \in \mathbb{N}_0. \tag{4.12}$$

Moreover, as in (4.5), the error state $e(k) = X(k) - X_{ss}$, is governed by

$$e(k+1) = \tilde{W}_{\sigma_k} e(k), \quad \sigma_k \in \{1, 2, \ldots, m\},\ k \in \mathbb{N}_0. \tag{4.13}$$

The system in (4.13) is also a switched linear system. The transformed matrix $\tilde{W}_j$ are the modes of the error dynamics. Generally, it is difficult to estimate the convergence rate of the ensemble with stochastic jumps. Previous works [45,56,66,87] have used the common Lyapunov function approaches, to analyze stability and the convergence rate. However, the existence of a common Lyapunov function is the *only sufficient condition* for the system stability, and hence there may not exist a common Lyapunov function for the asynchronous algorithm. Moreover, extremely large values of m make it very difficult to test every conditions for the existence of such a common Lyapunov function. For this reason, we bound the convergence rate of $\bar{e}(k)$, instead of bounding $e(k)$ directly.

Lemma 4.1 *Consider an i.i.d. jump linear system given by* (4.13) *with the switching probability* $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$. *If the initial state* $e(0)$ *is given and has no*

uncertainty, the expected value of $e(k)$ *is updated by*

$$\bar{e}(k) \triangleq \mathbb{E}[e(k)] = \Lambda^k e(0) \quad or \quad \bar{e}(k+1) = \Lambda \bar{e}(k), \tag{4.14}$$

where $\Lambda \triangleq \sum_{i=1}^{m} \pi_i \tilde{W}_i$.

Proof For an i.i.d. jump process with a given deterministic initial error $e(0)$, we have

$$\begin{aligned}
\mathbb{E}[e(k)] &= \mathbb{E}[\tilde{W}_{\sigma_{k-1}} e(k–1)] \\
&= \mathbb{E}[\tilde{W}_{\sigma_{k-1}} \tilde{W}_{\sigma_{k-2}} \dots \tilde{W}_{\sigma_1} \tilde{W}_{\sigma_0} e(0)] \\
&= \underbrace{\mathbb{E}[\tilde{W}_{\sigma_{k-1}}]}_{=\Lambda} \dots \underbrace{\mathbb{E}[\tilde{W}_{\sigma_1}]}_{=\Lambda} \underbrace{\mathbb{E}[\tilde{W}_{\sigma_0}]}_{=\Lambda} e(0) = \Lambda^k e(0).
\end{aligned}$$

□

Since the matrix Λ is given by $\Lambda = \sum_{i=1}^{m} \pi_i \tilde{W}_i$, the computation of Λ requires all matrices W_j, $j = 1, 2, \dots, m$. As pointed out earlier, this calculation is intractable due to the extremely large number of the switching modes m. Therefore, instead of using (4.14), we provide a computationally efficient method to bound $||\bar{e}(k)||$ through a Lyapunov theorem.

Consider a discrete-time Lyapunov function $V(k) = \bar{e}(k)^\top P \bar{e}(k)$, where P is a positive definite matrix. Since it is shown that the original state $X(k)$ is convergent to the unique steady-state X_{ss} as $k \to \infty$ irrespective of $\{\sigma_k\}$, the expected error $\bar{e}(k) \triangleq \bar{X}(k) - X_{ss}$ is asymptotically stable. Therefore, one can employ the *Converse Lyapunov Theorem* [68], which guarantees the existence of a positive definite matrix P, satisfying the following linear matrix inequality (LMI) condition $\Lambda^\top P \Lambda - P < -Q$, where Q is some positive definite matrix. The matrix inequality can be interpreted

in the sense of positive definiteness. (i.e., $A > B$ means the matrix $A - B$ is positive definite.) Then, the above LMI condition results in $\Delta V(k) = V(k+1) - V(k) = \bar{e}(k)^\top(\Lambda^\top P\Lambda - P)\bar{e}(k) < -\bar{e}(k)^\top Q\bar{e}(k) \leq -\lambda_{min}(Q) \parallel \bar{e}(k) \parallel^2$. Also, the Lyapunov function $V(k)$ satisfies

$$\lambda_{min}(P) \parallel \bar{e}(k) \parallel^2 \leq V(k) \leq \lambda_{max}(P) \parallel \bar{e}(k) \parallel^2,$$

resulting in $- \parallel \bar{e}(k) \parallel^2 \leq -\dfrac{V(k)}{\lambda_{max}(P)}$. Therefore, we have

$$\begin{aligned} \Delta V(k) < -\lambda_{min}(Q) \parallel \bar{e}(k) \parallel^2 \leq -\frac{\lambda_{min}(Q)}{\lambda_{max}(P)} V(k). \\ \Rightarrow V(k+1) < \Big(1 - \frac{\lambda_{min}(Q)}{\lambda_{max}(P)}\Big) V(k). \end{aligned} \tag{4.15}$$

Hence, $\parallel \bar{e}(k) \parallel$ is bounded by a following equation:

$$\parallel \bar{e}(k) \parallel^2 < K\left(1 - \frac{\lambda_{min}(Q)}{\lambda_{max}(P)}\right)^k \parallel e(0) \parallel^2, \tag{4.16}$$

where $K > 0$ is some constant.

Next, we bound the convergence rate for $||\bar{e}(k)||$ by using the result in (4.16) as follows.

Proposition 4.2 *For a stable i.i.d. jump linear system* (4.13) *with a stationary switching probability* Π*, consider a Lyapunov candidate function for the state* $\bar{e}$*, given by* $V \triangleq \bar{e}^\top P\bar{e}$*, where* P *is a positive definite matrix. In addition, a Lyapunov candidate function for* (4.12) *is given by* $V_j \triangleq e_j^\top P_j e_j$*,* $j = 1, 2, \ldots, m$*, where* P_j *is a positive definite matrix. According to the Converse Lyapunov Theorem, there exist* $P_j > 0$ *and* $P > 0$ *such that* $\tilde{W}_j^\top P_j \tilde{W}_j - P_j < -Q_j$*,* $j = 1, 2, \ldots, m$ *and*

$\Lambda^\top P\Lambda - P < -Q$, *where* Q_j *and* Q *are any positive definite matrices. Then, with a particular choice of these matrices, we assume that* P_j *and* P *satisfy the following conditions:*

$$\tilde{W}_j^\top P_j \tilde{W}_j - P_j = -I, \qquad j = 1, 2, \ldots, m, \tag{4.17}$$

$$\Lambda^\top P\Lambda - P \leq -\varepsilon_j I, \qquad \textit{for some } j, \tag{4.18}$$

where $\varepsilon_j \triangleq \dfrac{\lambda_{max}(P)}{\lambda_{max}(P_j)} > 0$, $\tilde{W}_j$ *are the modal matrices in* (4.12), *and* $\Lambda \triangleq \sum_{j=1}^{m} \pi_j \tilde{W}_j$.

Then, $||\bar{e}(k)||^2$ *is bounded by*

$$\| \bar{e}(k) \|^2 < K\left(1 - \frac{1}{\lambda_{max}(P_j)}\right)^k \| e(0) \|^2, \tag{4.19}$$

where $K > 0$ is some constant.

Proof By applying the result in (4.16) into (4.18), we have

$$\begin{aligned} \| \bar{e}(k) \|^2 &< K\left(1 - \frac{\lambda_{min}(\varepsilon_j I)}{\lambda_{max}(P)}\right)^k \| e(0) \|^2 \\ &= K\left(1 - \frac{\varepsilon_j}{\lambda_{max}(P)}\right)^k \| e(0) \|^2 \\ &= K\left(1 - \frac{1}{\lambda_{max}(P_j)}\right)^k \| e(0) \|^2 . \end{aligned}$$

The last equality in above equation holds by the definition of ε_j. □

Proposition 4.2 says that we can always guarantee the bound for $||\bar{e}(k)||$ if (4.18) holds. Consequently, the existence of such a P, satisfying (4.18) is the major concern in order to guarantee the bound $||\bar{e}(k)||$. The following lemma and theorem can be used to prove the existence of such a P.

Lemma 4.2 *Suppose that P_j is a positive definite matrix, satisfying* (4.17)*. Then, the largest eigenvalue of P_j is strictly greater than* 1 *for all j, i.e., $\lambda_{max}(P_j) > 1$, $\forall j$.*

Proof From (4.17), $P_j = \tilde{W}_j^\top P_j \tilde{W}_j + I$, $\forall j$. Then, with the eigenvectors $y \in \mathbb{R}^{Nnq}$ of P_j, the largest eigenvalue of P_j is given by its definition as follows:

$$\begin{aligned}
\lambda_{max}(P_j) &= \lambda_{max}(\tilde{W}_j^\top P_j \tilde{W}_j + I) \\
&= \max_{\substack{y \\ ||y||^2=1}} y^\top (\tilde{W}_j^\top P_j \tilde{W}_j + I) y \\
&= \max_{\substack{y \\ ||y||^2=1}} \left(y^\top \tilde{W}_j^\top P_j \tilde{W}_j y \right) + \underbrace{y^\top y}_{=||y||^2=1}
\end{aligned}$$

Since P_j is a positive definite matrix, $\tilde{W}_j^\top P_j \tilde{W}_j$ becomes a positive semi-definite matrix at least. Then, the scalar term $y^\top \tilde{W}_j^\top P_j \tilde{W}_j y$ cannot be zero unless $\tilde{W}_j^\top P_j \tilde{W}_j$ is a zero matrix or a triangular matrix with zero diagonal components, which is not the case. Hence, it is guaranteed that $y^\top \tilde{W}_j^\top P_j \tilde{W}_j y > 0$, implying $\lambda_{max}(P_j) > 1$, $\forall j$. □

Theorem 4.1 *Consider Lyapunov functions for* (4.12) *and* (4.13) *given by $V_j \triangleq e_j^\top P_j e_j$, $j = 1, 2, \ldots, m$, and $V \triangleq \bar{e}^\top P \bar{e}$, respectively, where the matrices $P_j > 0, \forall j$ and $P > 0$. By the Converse Lyapunov Theorem, we assume that the matrices P_j, $\forall j$, satisfies the condition* (4.17)*.*

Then, there exists a positive definite matrix P such that

$$\Lambda^\top P \Lambda - P \leq -\varepsilon_j I, \quad \textit{for some } j, \tag{4.20}$$

where $\varepsilon_j \triangleq \dfrac{\lambda_{max}(P)}{\lambda_{max}(P_j)} > 0.$

Proof We prove by contradiction. Suppose that there exist no such $P > 0$, satisfying

(4.20), which is equivalent to that for ***all*** matrices $P > 0$, the inequality $\Lambda^\top P\Lambda - P > -\varepsilon_j I$ holds $\forall j$. The above inequality can be interpreted in the quadratic sense. In other words, for any non-zero vector v that has a proper dimension, the following condition holds:

$$v^\top \left(\Lambda^\top P\Lambda - P + \varepsilon_j I\right) v > 0,\ \forall j \tag{4.21}$$

As a particular choice of v, we let the vector v be the eigenvector of the matrix Λ, i.e., $\Lambda v = \lambda\Lambda$, where λ is the eigenvalue of Λ. Since (4.21) holds for any matrix $P > 0$, we let $P = I$, which results in $\varepsilon_j = \dfrac{\lambda_{max}(I)}{\lambda_{max}(P_j)} = \dfrac{1}{\lambda_{max}(P_j)}$. Hence, we have

$$\begin{aligned} 0 &< v^\top \left(\Lambda^\top\Lambda - I + \frac{1}{\lambda_{max}(P_j)} I\right) v \\ &= (\underbrace{\Lambda v}_{=\lambda v})^\top (\underbrace{\Lambda v}_{=\lambda v}) - ||v||^2 + \frac{1}{\lambda_{max}(P_j)}||v||^2 \\ &= \left(\lambda^2 - 1 + \frac{1}{\lambda_{max}(P_j)}\right)||v||^2, \quad \forall j. \end{aligned}$$

From the structure of the matrix Λ, it can be shown that $\det(\Lambda) = 0$. Therefore, one of the eigenvalues λ is zero. Moreover, Lemma 4.2 states that $\dfrac{1}{\lambda_{max}(P_j)} < 1$, $\forall j$. As a consequence, with $\lambda = 0$, we have

$$0 < \underbrace{\left(-1 + \frac{1}{\lambda_{max}(P_j)}\right)}_{<0} \underbrace{||v||^2}_{>0} < 0, \quad \forall j.$$

which is a *contradiction*. □

Remark 4.1 *Proposition 4.2 provides a very efficient way to bound the convergence rate for $||\bar{e}(k)||$. According to the proposed methods, it is unnecessary to compute the matrix Λ and to keep all matrices W_j, $j = 1, 2, \ldots, m$ since $||\bar{e}(k)||$ is bounded by*

the proposed Lyapunov function. Also, Theorem 4.1 guarantees the condition (4.18)*, which is assumed in Proposition 4.2.*

Note that we specify the modal matrix W_m *in* (4.5) *as the most delayed case – all PEs use the oldest value in the buffer. Therefore, it can be inferred that* $\lambda_{max}(P_m) \geq \lambda_{max}(P_j)$*,* $\forall j$*, which results in*

$$||\bar{e}(k)||^2 < K\Big(1 - \frac{1}{\lambda_{max}(P_m)}\Big)^k ||e(0)||^2, \tag{4.22}$$

where K *is a positive constant. Therefore, the only information required to compute the convergence rate of* $||\bar{e}(k)||$*, is the matrix* W_m *with the corresponding positive definite matrix* P_m*. As a result, the rate of convergence can be calculated by the proposed methods without any scalability problems.*

4.6 Error Analysis

In this section, we investigate the error probability, which quantifies the deviation of the random vector $X(k)$ from its steady-state value X_{ss} in probability. To measure this error probability, the Markov inequality given by $\mathbf{Pr}\big(X \geq \epsilon\big) \leq \dfrac{\mathbb{E}[X]}{\epsilon}$, where X is a nonnegative random variable and ϵ is a positive constant, is used. First of all, we investigate the term $\text{vec}\left(e(k)e(k)^\top\right)$ as follows:

$$\begin{aligned}\text{vec}\left(e(k)e(k)^\top\right) &= \text{vec}\left(\tilde{W}_{\sigma_{k-1}} e(k-1)e(k-1)^\top \tilde{W}_{\sigma_{k-1}}^\top\right) \\ &= \big(\tilde{W}_{\sigma_{k-1}} \otimes \tilde{W}_{\sigma_{k-1}}\big)\text{vec}\big(e(k-1)e(k-1)^\top\big).\end{aligned} \tag{4.23}$$

In the second equality of above equation, we used the property that $\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B)$.

By taking the expectation with new definitions $y(k) \triangleq \text{vec}\left(e(k)e(k)^\top\right)$, $\bar{y}(k) \triangleq$

$\mathbb{E}[y(k)]$, and $\Gamma_{\sigma_k} \triangleq \tilde{W}_{\sigma_k} \otimes \tilde{W}_{\sigma_k}$, (4.23) becomes

$$\begin{aligned}\bar{y}(k) \triangleq \mathbb{E}[y(k)] &= \mathbb{E}\left[\Gamma_{\sigma_{k-1}} y(k-1)\right] \\ &= \sum_{r=1}^{m} \mathbb{E}\left[\Gamma_{\sigma_{k-1}} y(k-1) \,\middle|\, \sigma_{k-1} = r\right] \mathbf{Pr}(\sigma_{k-1} = r) \\ &= \sum_{r=1}^{m} \pi_r \Gamma_r \mathbb{E}\left[y(k-1)\right],\end{aligned}$$

resulting in $\bar{y}(k) = \left(\sum_{r=1}^{m} \pi_r \Gamma_r\right) \bar{y}(k-1)$, where in the second line we applied the law of total probability and the last equality holds by $\mathbf{Pr}(\sigma_{k-1} = r) = \pi_r$ for i.i.d. switching.

By the exactly same argument given in Lemma 4.1 and Proposition 4.2, the upper bound for $\bar{y}(k)$ is obtained as follows:

$$||\bar{y}(k)|| < K\left(1 - \frac{1}{\lambda_{max}(\tilde{P}_m)}\right)^{k/2} ||y(0)||, \quad \forall k \in \mathbb{N}, \tag{4.24}$$

where K is some positive constant and $\tilde{P}_m$ is a positive definite matrix, satisfying the condition $\Gamma_m^\top \tilde{P}_m \Gamma_m - \tilde{P}_m = -I$. However, unlike the positive definite matrix $P_m \in \mathbb{R}^{Nnq \times Nnq}$ in (4.17), the dimension of the matrix $\tilde{P}_m$ is given by $\tilde{P}_m \in \mathbb{R}^{(Nnq)^2 \times (Nnq)^2}$, which may be large in size, and hence incurs computational intractabilities to obtain such a $\tilde{P}_m$. Therefore, we introduce the following proposition and theorem in order to further facilitate the computation of $\lambda_{max}(\tilde{P}_m)$ as follows.

Proposition 4.3 *Consider a positive definite matrix $\tilde{P}_m$, satisfying the following condition $\Gamma_m^\top \tilde{P}_m \Gamma_m - \tilde{P}_m = -I$, where $\Gamma_m \triangleq \tilde{W}_m \otimes \tilde{W}_m$, and $\tilde{W}_m$ is any real square matrix. If we assume that there exist finite, positive constants k_0, c_0, and c_1 such*

that

$$1 \leq ||\tilde{W}_m^k||^4 \leq c_0, \qquad \textit{for } k \in [0, k_0), \tag{4.25}$$

$$||\tilde{W}_m^k||^4 \leq c_1 < 1, \qquad \textit{for } k \in [k_0, \infty), \tag{4.26}$$

then, the largest eigenvalue of $\tilde{P}_m$ *is bounded by the following function:*

$$\lambda_{max}(\tilde{P}_m) < \sum_{k=0}^{\infty} ||\tilde{W}_m^k||^4 \leq k_0 c_0 \left(\frac{1}{1-c_1}\right), \tag{4.27}$$

Proof The leftmost inequality in (4.27) can be proved as follows. The positive definite matrix $\tilde{P}_m$ satisfying the condition $\Gamma_m^\top \tilde{P}_m \Gamma_m - \tilde{P}_m = -I$, is analytically computed by $\tilde{P}_m = \sum_{k=0}^{\infty} \left(\Gamma_m^{\top k}\right) I \left(\Gamma_m^k\right) = \sum_{k=0}^{\infty} \Gamma_m^{\top k} \Gamma_m^k$. Then, for a given matrix $\Gamma_m \triangleq \tilde{W}_m \otimes \tilde{W}_m$, we have

$$\begin{aligned}
\Gamma_m^{\top k} \Gamma_m^k &< \rho(\Gamma_m^{\top k} \Gamma_m^k) I \\
&= \rho({\Gamma_m^k}^\top \Gamma_m^k) I \\
&= \sigma_{max}^2(\Gamma_m^k) I \\
&= ||\Gamma_m^k||^2 I \\
&= ||(\tilde{W}_m \otimes \tilde{W}_m)^k||^2 I \\
&= ||\tilde{W}_m^k||^4 I,
\end{aligned} \tag{4.28}$$

where $\rho(\cdot)$ and $\sigma_{max}(\cdot)$ denote the spectral radius and the spectral norm, respectively. For equality conditions in (4.28), we used the known property that $\sqrt{\rho({\Gamma_m^k}^\top \Gamma_m^k)} = \sigma_{max}(\Gamma_m^k) = ||\Gamma_m^k||$ and $||(\tilde{W}_m \otimes \tilde{W}_m)^k|| = ||\tilde{W}_m^k \otimes \tilde{W}_m^k|| = ||\tilde{W}_m^k||^2$, $\forall k \in \mathbb{N}_0$. By summing up from $k = 0$ to ∞, and then taking the largest eigenvalue in (4.28), we have $\lambda_{max}(\tilde{P}_m) = \lambda_{max}\left(\sum_{k=0}^{\infty} \Gamma^{\top k} \Gamma^k\right) < \sum_{k=0}^{\infty} ||\tilde{W}_m^k||^4$.

For the rightmost inequality in (4.27), the assumptions in (4.25)-(4.26) result in

$$\begin{aligned}
\sum_{k=0}^{\infty} ||\tilde{W}_m^k||^4 &= \underbrace{\sum_{k=0}^{k_0-1} ||\tilde{W}_m^k||^4}_{\leq k_0 c_0} + \sum_{k=k_0}^{\infty} ||\tilde{W}_m^k||^4 \\
&\leq k_0 c_0 + \sum_{k=k_0}^{2k_0-1} ||\tilde{W}_m^k||^4 + \sum_{k=2k_0}^{3k_0-1} ||\tilde{W}_m^k||^4 + \cdots \\
&= k_0 c_0 + \sum_{k=0}^{k_0-1} ||\tilde{W}_m^{(k_0+k)}||^4 + \sum_{k=2k_0}^{3k_0-1} ||\tilde{W}_m^k||^4 + \cdots \\
&\leq k_0 c_0 + \underbrace{||\tilde{W}_m^{k_0}||^4}_{\leq c_1} \underbrace{\sum_{k=0}^{k_0-1} ||\tilde{W}_m^k||^4}_{\leq k_0 c_0} + \sum_{k=0}^{k_0-1} ||\tilde{W}_m^{(2k_0+k)}||^4 + \cdots \\
&\leq k_0 c_0 + k_0 c_0 c_1 + \underbrace{||\tilde{W}_m^{2k_0}||^4}_{\leq c_1^2} \underbrace{\sum_{k=0}^{k_0-1} ||\tilde{W}_m^k||^4}_{\leq k_0 c_0} + \cdots \\
&\leq k_0 c_0 + k_0 c_0 c_1 + k_0 c_0 c_1^2 + \cdots \\
&= k_0 c_0 \left(\sum_{n=0}^{\infty} c_1^n \right) = k_0 c_0 \left(\frac{1}{1-c_1} \right).
\end{aligned}$$

Hence, we have $\sum_{k=0}^{\infty} ||\tilde{W}_m^k||^4 \leq k_0 c_0 \left(\frac{1}{1-c_1} \right)$. □

Theorem 4.2 *Consider a stable, i.i.d. jump linear system with subsystem dynamics $\tilde{W}_j$ given in* (4.13)*. Then, the probability of $||e(k)||^2 > \epsilon$, where ϵ is some positive constant, is given by*

$$\boldsymbol{Pr}\Big(||e(k)||^2 > \epsilon \Big) \leq \min(1, \beta), \quad k \in \mathbb{N}_0, \tag{4.29}$$

where $\beta \triangleq \frac{\sqrt{n}K}{\epsilon} \left(1 - \frac{1-c_1}{k_0 c_0} \right)^{k/2} ||y(0)||$, $K > 0$ is a constant, c_0, c_1, k_0 are positive constants such that the conditions (4.25)-(4.26) *are satisfied.*

Proof At first, we consider the following equality condition given by

$$\begin{aligned} ||e(k)||^2 &= e(k)^\top e(k) \\ &= \mathrm{tr}(e(k)^\top e(k)) \\ &= \mathrm{tr}\big(I\left(e(k)e(k)^\top\right)\big) \\ &= \mathrm{vec}(I)^\top \mathrm{vec}(e(k)e(k)^\top) \\ &= \mathrm{vec}(I)^\top y(k), \end{aligned} \tag{4.30}$$

where we used the cyclic permutation property for the trace operator in the first line and the equality in the second line holds by the property $\mathrm{tr}(X^\top Y) = \mathrm{vec}(X)^\top \mathrm{vec}(Y)$ for any square matrix $X, Y \in \mathbb{R}^{n\times n}$.

We take the expectation in both sides of (4.30), which leads to

$$\mathbb{E}\big[||e(k)||^2\big] = \mathrm{vec}(I)^\top \mathbb{E}\big[y(k)\big] = \mathrm{vec}(I)^\top \bar{y}(k). \tag{4.31}$$

Since the term $\mathbb{E}\big[||e(k)||^2\big]$ is a scalar value, taking the Euclidean norm returns the same value. Hence, applying the Euclidean norm in (4.31) results in

$$\begin{aligned} \mathbb{E}\big[||e(k)||^2\big] &= ||\mathrm{vec}(I)^\top \bar{y}(k)|| \\ &\leq ||\mathrm{vec}(I)^\top|| \cdot ||\bar{y}(k)|| = \sqrt{n} \cdot ||\bar{y}(k)||. \end{aligned} \tag{4.32}$$

Now, plugging (4.24) and (4.27) into (4.32) leads to

$$\mathbb{E}\big[||e(k)||^2\big] < \sqrt{n}K\left(1 - \frac{1-c_1}{k_0 c_0}\right)^{k/2} ||y(0)||.$$

Finally, by applying the Markov inequality the above equation ends up with

$$\mathbf{Pr}\Big(||e(k)||^2 > \epsilon\Big) \leq \frac{\mathbb{E}\big[||e(k)||^2\big]}{\epsilon} < \beta,$$

where $\beta \triangleq \dfrac{\sqrt{n}K}{\epsilon}\left(1 - \dfrac{1-c_1}{k_0 c_0}\right)^{k/2} ||y(0)||$.

Since the probability cannot exceed one, we have $\mathbf{Pr}\Big(||e(k)||^2 > \epsilon\Big) \leq \min(1, \beta)$

□

Theorem 6.1 represents the error probability for a given bound ϵ. Since $e(k)$ is a time-varying variable, the probability $\mathbf{Pr}\left(||e(k)||^2 > \epsilon\right)$ also changes with respect to time. Starting from a given initial condition $y(0)$, this probability will converge to zero if $\left(1 - \dfrac{1-c_1}{k_0 c_0}\right) < 1$.

4.7 Simulations

In order to test the proposed methods, simulation was carried out for the one-dimensional heat equation. We implemented the asynchronous parallel algorithm with `CUDA C++ programming` on `nVIDIA Tesla`™ `C2050` GPU, which has 448 `CUDA cores`. The simulations were performed with the following parameters:

- Simulation Parameters:

$$\begin{aligned}
\Delta x = &\quad 0.1, \Delta t = 0.01, \alpha = 0.5, r = \alpha\frac{\Delta t}{\Delta x^2} = 0.5 \\
I.C.: &\quad u_i = \cos^2\left(\frac{3\pi i}{2(N-1)}\right),\ i = 1, 2, \ldots, N \\
B.C.: &\quad u_1(k) = 1,\ u_N(k) = 0,\ \forall k
\end{aligned}$$

- Buffer length: $q = 3$

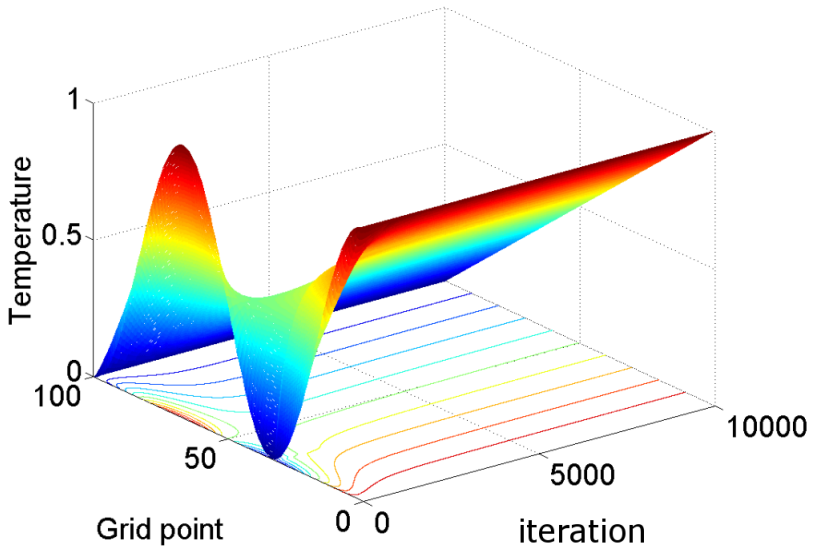
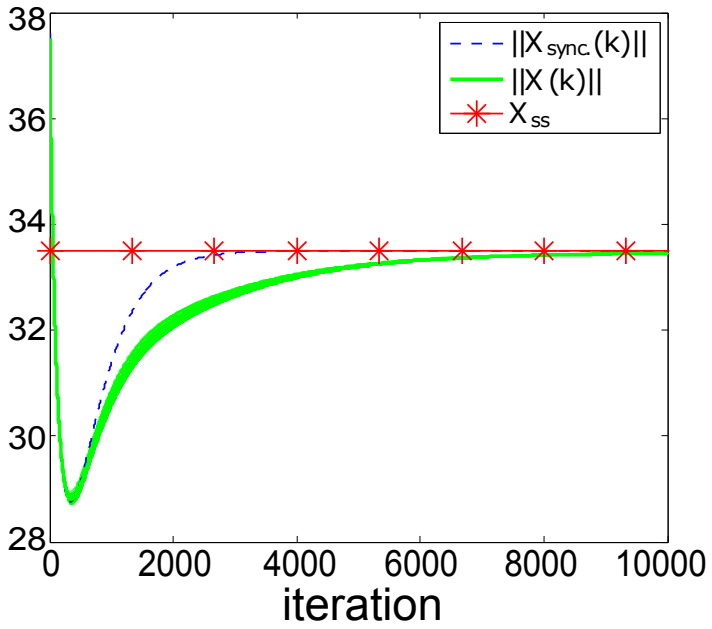


Figure 4.2: The spatio-temporal change of the temperature. Initially, the temperature was given by the cosine square function. The total grid points are 100, and the simulation was terminated when $k = 10000$.

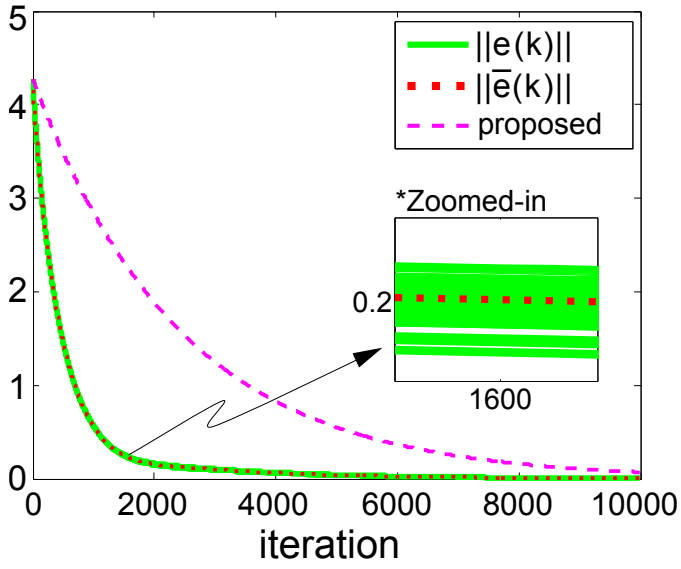
- Number of PEs: $N = 100$.
- Number of grid points in PE: $n = 1$

For a given initial temperature, the spatio-temporal evolution of the state is presented in Fig. 4.2. As time k increases, the curved shape of the temperature, given as a cosine square function initially, flattens out. This simulation represents the synchronous case.

In Fig. 4.3 (a), the ensemble of the trajectories is shown for the asynchronous algorithm. The solid lines show the trajectories of total 300 simulations. Due to the randomness in the asynchronous algorithm, the trajectories differ from each other. For a reference, the synchronous scheme is also shown by a dashed line. Although it seems that the synchronous scheme converges faster with respect to the given



(a) Stability



(b) Convergence rate

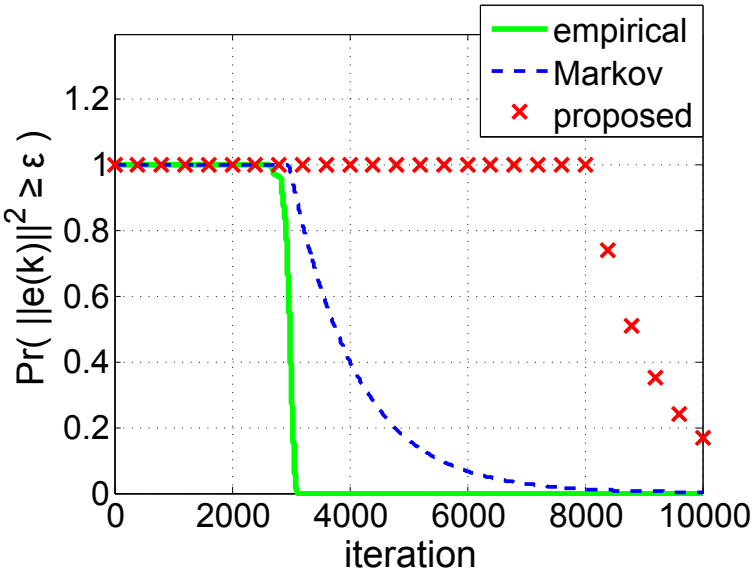
Figure 4.3: The results for the stability and convergence rate. (a) The solid lines represent the ensembles of total 300 simulations. The synchronous case is given by dashed line. The steady-state is depicted by starred line. (b) The solid and dotted lines represent 300 ensembles for $||e(k)||$ and the normed empirical mean $||\bar{e}(k)||$, respectively. The dashed line shows the upper bound of $||\bar{e}(k)||$ from the proposed Lyapunov function, respectively.

iteration step, the physical simulation time may take more because the idle time is necessary at each iteration in the synchronous case. As the proposed method guarantees the stability through the common eigenvectors, both synchronous and asynchronous trajectories converged to the same steady-state value X_{ss}, depicted by starred line.

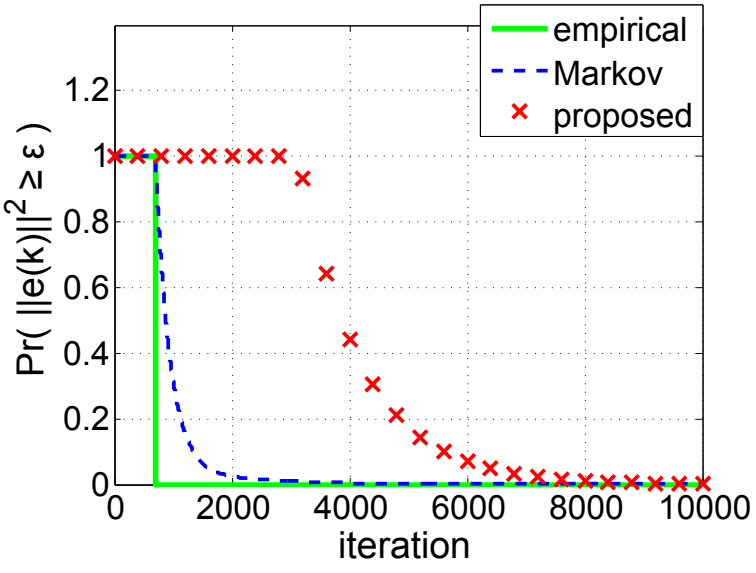
Next, we present the result for the convergence rate of the asynchronous algorithm. We assume that the switching probability Π has the form of an i.i.d. jump process. Fig. 4.3 (b) shows the convergence rate of $||\bar{e}(k)||$, which describes how fast the expected value of the state converges to X_{ss}. The solid lines are 300 sample trajectories of $||e(k)||$, starting from the given initial condition: $e(0) = X(0) - X_{ss}$. The dotted line depicts the time history of the normed *empirical* mean $||\bar{e}(k)||$, whereas the dashed line shows an upper bound by the proposed Lyapunov method (4.22). Note that $||e(k)||$ is a random variable, and hence the normed empirical mean $||\bar{e}(k)||$ was obtained by averaging the data over 300 simulations. In the proposed method, however, it is not necessary to execute the simulation multiple times.

Fig. 4.5 represents the result for the error probability with respect to time and ϵ. For different values of ϵ, Fig. 4.5 (a) and (b) describe the time history of the error probabilities. The solid line denotes the empirical probability obtained from data – i.e., the number of samples satisfying $||e(k)||^2 > \epsilon$ divided by the total number of samples. The dashed line depicts the Markov inequality, computed from $\frac{\mathbb{E}\left[||e(k)||^2\right]}{\epsilon}$, where $\mathbb{E}\left[||e(k)||^2\right]$ is obtained by the statistics. Finally, the cross symbols mean the upper bound by the proposed method. As shown in Fig. 4.5 (a) and (b), the probabilities for all cases converge to zero since the error is asymptotically convergent.

On the other hands, Fig. 4.5 (c), (d) show the error probability with respect to ϵ at fixed time instance. In this result, the time is fixed at $k = 9000$ out of total 10000 iteration times, and the probability is computed while increasing ϵ values. In Fig.

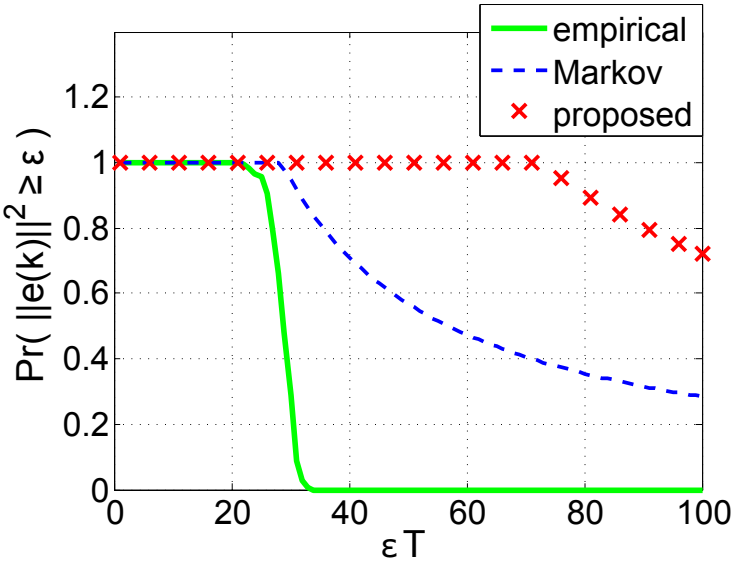


(a) $\epsilon = 0.01$

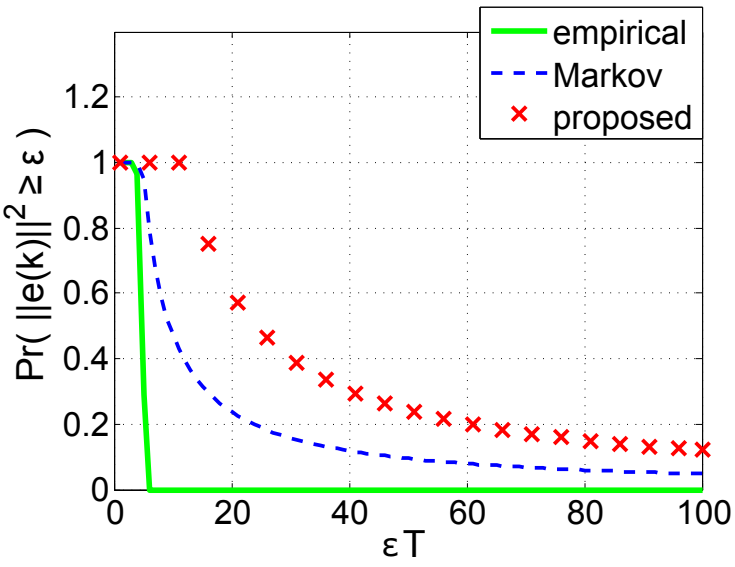


(b) $\epsilon = 1$

Figure 4.4: Error probability with respect to iteration step. The solid line and dashed line represent empirical error probability and empirical Markov inequality, respectively. The cross symbol denotes the upper bound for the error probability by the proposed method.



(c) $T = 2 \times 10^{-6}$



(d) $T = 12 \times 10^{-6}$

Figure 4.5: Error probability with respect to given constant error bound ϵ. The solid line and dashed line represent empirical error probability and empirical Markov inequality, respectively. The cross symbol denotes the upper bound for the error probability by the proposed method.

4.5 (c) and (d), ϵT is given by the index along x-axis, where the value of T is given in Fig. 4.5 (c) and (d), respectively. In both cases, the error probabilities decrease as ϵ increases.

Although the proposed methods provide a conservative bound, it does not require executing the code multiple times to predict the convergence rate or the error probability. In addition to that, the proposed methods are carried out in a computationally efficient manner without storing all subsystem matrices. In this example, we have $m = 3^{2(100-2)} \approx 3^{200}$, and keeping 3^{200} numbers of matrices is intractable in the real implementation. The proposed method, however, guarantees the convergence rate and the error probability, without any scalability issues. Therefore, the presented methods provide a computationally efficient tool to analyze the asynchronous numerical schemes.

4.8 Concluding Remarks

This chapter studied the stability, convergence rate, and error probability of the asynchronous parallel numerical algorithm. The asynchronous algorithm achieves better performance in terms of the total simulation time, particularly when massively parallel computing is required because it doesn't wait for synchronization across PEs. In order to analyze the asynchronous numerical algorithm, we adopted the switched linear system framework. Although modeling of massively parallel numerical algorithms as switched dynamical systems results in a very large number of modes, we developed new methods that circumvent this scalability issue. While the results presented here are based on 1D heat equation, the analysis approach is generic and be applicable to other PDEs as well.

5. ON THE CONVERGENCE OF ASYNCHRONOUS DISTRIBUTED QUADRATIC PROGRAMMING VIA DUAL DECOMPOSITION

5.1 Introduction

Recent advancement of distributed and parallel computing technologies has brought massive processing capabilities in solving large-scale optimization problems. Distributed and parallel computing may reduce computation time to find an optimal solution by leveraging the parallel processing in computation. Particularly, distributed optimization will likely be considered as a key element for large-scale statistics and machine learning problems, currently represented by the word "big data". One of the reasons for the preference of distributed optimization in big data is that the size of data set is so huge that each data set is desirably stored in a distributed manner. Thus, global objective is achieved in conjunction with local objective functions assigned to each distributed node, which requires communication between distributed nodes in order to attain an optimal solution.

For several decades, there have been remarkable studies that have enabled to find an optimal solution in a decentralized fashion, for example, dual decomposition [21], [4], [29], [38], [6], augmented Largrangian methods for constrained optimization [50], [81], [35], [1], alternating direction method of multipliers (ADMM) [41], [37], [36], Spingarn's method, [88], Bregman iterative algorithms for ℓ_1 problems [12], [15], [27], Douglas-Rachford splitting [26], [70], and proximal methods [92]. More details about history of developments on the methods listed above can be found in the literature [11]. In this study, we mainly focus on the analysis of *asynchronous* distributed optimization problems. In particular, we aim to investigate the behavior of asynchrony in the Lagrangian dual decomposition method for *distributed quadratic*

programming (QP) problems, where QP problems refer to the optimization problems with a quadratic objective function associated with linear constraints. This type of QP problems has broad applications including least square with linear constraints, regression analysis and statistics, SVMs, lasso, portfolio optimization problems, etc. With an implementation of Lagrangian dual decomposition, the original QP problems that are separable can be solved in a distributed sense. For this dual decomposition technique, we will study how the asynchronous computing algorithms affect on the convergence as well as the rate of convergence for the dual variable.

Typically, distributed optimization requires synchronization of the data set at each iteration step due to the interdependency of data. For massive parallelism, this synchronization may result in a large amount of waiting time as load imbalance between distributed computing resources would take place at each iteration step. In this case, some nodes that have completed their tasks should wait for others to finish assigned jobs, which causes idle process of computing resources, incurring waste of computation time. In this chapter, we attack this restriction on synchronization penalty necessarily required in distributed and parallel computing, through the implementation of *asynchronous computing algorithms*. The asynchronous computing algorithms that do not suffer from synchronization latency thus have a potential to break through the paradigm of distributed and parallel optimization. Unfortunately, it is not completely revealed yet what is the effect of asynchrony on the convergence as well as the rate of that in the distributed optimization. Due to the stochastic behavior of asynchrony, the solution for the asynchronous distributed QP may diverge even if it is guaranteed that the synchronous scheme provides a convergence to an optimal solution. Although Bertsekas [7] introduced a sufficient condition for the convergence of general asynchronous fixed-point iterations (see chapter 6.2), which is equivalent to a diagonal dominance condition for QP problems, however, this condi-

tion is known to be very strong and thus conservative, according to the literature [71]. Therefore, the primal emphasis of this research is placed on: 1) convergence analysis; 2) analytic estimation on the rate of convergence, by employing a new framework for analysis of distributed QP problems with an asynchronous update of dual variable.

For this purpose, we will adopt the *switched system* [34], [33], [32], [63], [64], [62] framework as an analysis tool. In general, the switched system is defined as a dynamical system that consists of a set of subsystem dynamics and a certain switching logic that governs a switching between subsystems. For asynchronous algorithms of which dynamics is modeled by the switched system, subsystem dynamics denotes all possible asynchronous computing due to the difference of data processing time in each distributed computing devices. Then, a certain switching logic can be implemented to stand for a random switching between subsystem dynamics. Thus, the switched system framework can be used to properly model the dynamics of asynchronous computing algorithms. Lee [62], for example, introduced the switched system to represent the behavior of asynchrony in massively parallel numerical algorithms. In this literature, the authors applied the switched dynamical system framework in order to analyze the convergence, rate of convergence, and error probability for asynchronous parallel numerical algorithms. Based on this switched system framework, this chapter will provide a new approach for analysis of asynchronous distributed QP problems with dual decomposition, to guarantee both necessary and sufficient convergence conditions in the mean square sense. In addition, we will study how fast each scheme (e.g., synchronous and asynchronous scheme) converges to an optimal solution by studying the rate of convergence in analytic form. Therefore, this chapter will present fundamental yet important analysis on the asynchronous distributed QP problems through the switched system framework, which facilitates investigation on the stochastic behavior of asynchrony.

Rest of this chapter is organized as follows. In section 5.2, preliminaries are presented in connection with problem formulations for asynchronous distributed QP problems using dual decomposition. Section 5.3 introduces the switched system to model the asynchrony in the asynchronous distributed QP problems. The results for the convergence and the rate of convergence by employing the switched system framework are derived in section 5.4 and 5.5, respectively. The numerical example with a real implementation of distributed and parallel QP is provided in section 5.6, to verify the validity of the proposed methods. Finally, section 5.7 concludes the chapter.

5.2 Preliminaries and Problem Formulation

Notation: The real number, positive integer, and the non-negative integer are denoted by the symbol $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{N}_0$, respectively. The symbol $^\top$ represents the transpose operator. For any real matrix $A, B \in \mathbb{R}^{n\times n}$, the inequality $A < B$ is interpreted by the quadratic sense. (i.e., $v^\top Av < v^\top Bv$ for any real vector $v \in \mathbb{R}^n$). In addition, the symbol $\otimes$ and $\mathsf{diag}(\cdot)$ stand for the Kronecker product and diagonal operator for the square matrix. respectively. Finally, for any real number a and any real matrix A, the symbol $|a|$ and $|A|$ represent the absolute value of a and the matrix with absolute value of all elements in A, respectively.

5.2.1 Duality Problem

Consider the following QP problem with a linear inequality constraint.

$$\text{minimize} \qquad f(x) \tag{5.1}$$

$$\text{subject to} \qquad Ax \leq b, \tag{5.2}$$

where $f(x)$ is given by a quadratic form, meaning $f(x) = \frac{1}{2}x^\top Qx + c^\top x$, the matrix $Q \in \mathbb{R}^{n\times n}$ is a symmetric, positive definite and $c \in \mathbb{R}^n$ is a vector. Further, in the inequality constraint (5.2), it is such that $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. If we define the Lagrangian as $L(x, y) \triangleq f(x) + y^\top(Ax - b)$, where $y \in \mathbb{R}^m$ is the dual variable or Lagrange multiplier, then the dual problem for above QP is formulated as follows:

Duality using Lagrangian:

$$\text{maximize} \qquad \inf_x L(x, y) \tag{5.3}$$

$$\text{subject to} \qquad y \geq 0. \tag{5.4}$$

The primal optimal point $x^\star$ is obtained from a dual optimal point $y^\star$ as

$$x^\star = \operatorname*{argmin}_x L(x, y^\star).$$

By implementing gradient ascent, one can solve the dual problem, provided that $\inf L(x, y)$ is differentiable. In this case, the iteration to find the $x^\star$ is constructed as follows:

$$x^{k+1} := \operatorname*{argmin}_x L(x, y^k), \tag{5.5}$$

$$y^{k+1} := y^k + \alpha^k(Ax^{k+1} - b), \tag{5.6}$$

where α^k is a step size and the upper script denotes the discrete-time index for iteration.

For the quadratic objective function $f(x)$, the value $\operatorname*{argmin}_x L(x, y^k)$ can be alter-

natively obtained by $\nabla_x L(x, y^k) = 0$, which leads to

$$\begin{aligned}\underset{x}{\operatorname{argmin}}\, L(x, y^k) &= \nabla_x \left(\frac{1}{2} x^\top Q x + c^\top x + {y^k}^\top (Ax - b) \right) \\ &= Qx + c + A^\top y^k = 0.\end{aligned}$$

From (5.5), we have

$$x^{k+1} = -Q^{-1}(A^\top y^k + c). \tag{5.7}$$

Plugging (5.7) into (5.6) results in

$$\begin{aligned}y^{k+1} &= y^k + \alpha^k \left(A \left(-Q^{-1}(A^\top y^k + c) \right) - b \right) \\ &= (I - \alpha^k A Q^{-1} A^\top) y^k - \alpha^k (A Q^{-1} c + b).\end{aligned} \tag{5.8}$$

With the assumption that $y^k \geq 0 \ \forall k$, the above equation provides the solution for $y^\star$ and hence $x^\star$, if $\rho(I - \alpha^k A Q^{-1} A^\top) < 1$ as follows:

$$y^\star = (I - \alpha^k A Q^{-1} A^\top) y^\star - \alpha^k (A Q^{-1} c + b).$$

$$\Rightarrow \begin{cases} y^\star = -(A Q^{-1} A^\top)^{-1} (A Q^{-1} c + b), & \text{(if } A Q^{-1} A^\top \text{ is non-singular)}, \\ x^\star = -Q^{-1}(A^\top y^\star + c). \end{cases} \tag{5.9}$$

5.2.2 Dual Decomposition with Synchronous Update

In this subsection, we consider that $f(x) = \frac{1}{2}x^\top Qx + c^\top x$ is *separable*, which means

$$\begin{aligned} f(x) &= \sum_{i=1}^{N} f_i(x_i) \\ &= \sum_{i=1}^{N} \left(\frac{1}{2} x_i^\top Q_i x_i + c_i^\top x_i \right), \end{aligned}$$

where $x = [x_1^\top, x_2^\top, \ldots, x_N^\top]^\top$ and the variables $x_i \in \mathbb{R}^{n_i}$, $i = 1, 2, \ldots, N$ are subvectors of x. Also, the matrix A in (5.2) satisfies $Ax = \sum_{i=1}^{N} A_i x_i$, where A_i is such that $A = [A_1, A_2, \ldots, A_N]$.

Then, the equations (5.5) and (5.6) are updated by

$$x_i^{k+1} := \underset{x_i}{\operatorname{argmin}}\, L(x_i, y^k) = -Q_i^{-1}(A_i^\top y^k + c), \tag{5.10}$$

$$y^{k+1} := y^k + \alpha^k (Ax^{k+1} - b). \tag{5.11}$$

Note that when updating x_i^{k+1}, $i = 1, 2, \ldots, N$, each value is computed by distributed nodes. Hence, the computation for x_i^{k+1} can be processed in parallel and then, each value of x_i^{k+1} is transmitted to the master node to compute y^{k+1} in the gathering stage. Therefore, as in (5.11), updating y^{k+1} requires synchronization of x_i^{k+1} across all spatial index i at time $k+1$ because x^{k+1} is obtained by stacking x_i^{k+1} from $i = 1$ to N. In Fig. 5.1, we described the conceptual schematic of synchronous update for dual variable y. If computing delay occurs among one of the index i due to the difference of processing time in distributed node, the process to update y^{k+1} has to be paused until all data is received from distributed nodes. This implies that the more parallel computing we have, the more delays may take place, resulting in

a large amount of the idle time. Consequently, this idle time for synchronization becomes dominant compared to the pure computation time to solve the QP problem in parallel. In massive parallel computing algorithm, it has been reported that the synchronization latency may be up to 50% of total computation time according to the literature [13]. In order to mitigate or avoid this type of restriction that severely affects on the performance to obtain an optimal solution, we introduce *asynchronous computing algorithm* in the following subsection.

5.2.3 Dual Decomposition with Asynchronous Update

In order to alleviate this synchronization penalty, we consider asynchronous update of dual variable y. In this case, the master node to compute y^{k+1} does not wait until all x_i^{k+1} is gathered. Rather, it proceeds with the value for x_i saved in the buffer memory. Thus, y value is updated asynchronously. To model the asynchronous dynamics of dual decomposition, we consider the new state vectors as follows.

- The state for the *Asynchronous* model:

$$\tilde{x}^k := [{x_1^{k_1^*}}^\top, {x_2^{k_2^*}}^\top, \ldots, {x_N^{k_N^*}}^\top]^\top,$$

  where $k_i^* \in \{k, k-1, \ldots, k-q+1\}$, $i = 1, 2, \ldots, N$, denotes delay term that may take place due to the load imbalance in distributed nodes, and the term $q \in \mathbb{N}$ represents the maximum possible delay.

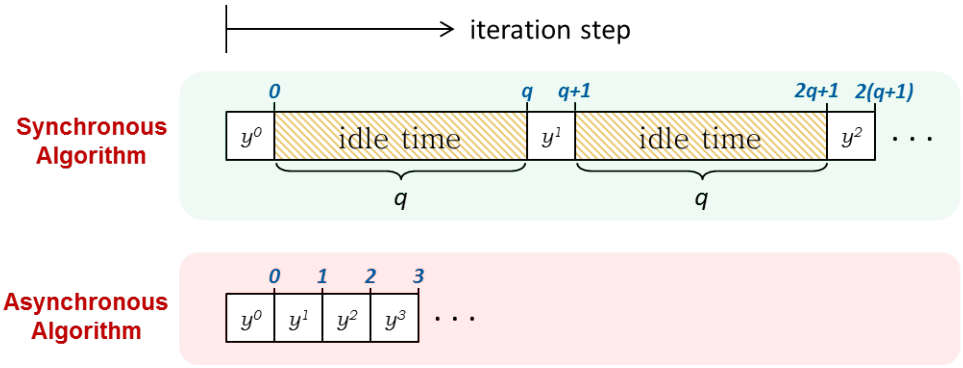


Figure 5.1: The schematic of update timing for the variable y^k; upper one shows the *synchronous algorithm*, where q is the length of maximum delay – i.e., all delays are bounded by q; bottom one shows *asynchronous algorithm*. The time to compute y^k is given by 1 CPU time.

For this asynchronous case, y-update is given by

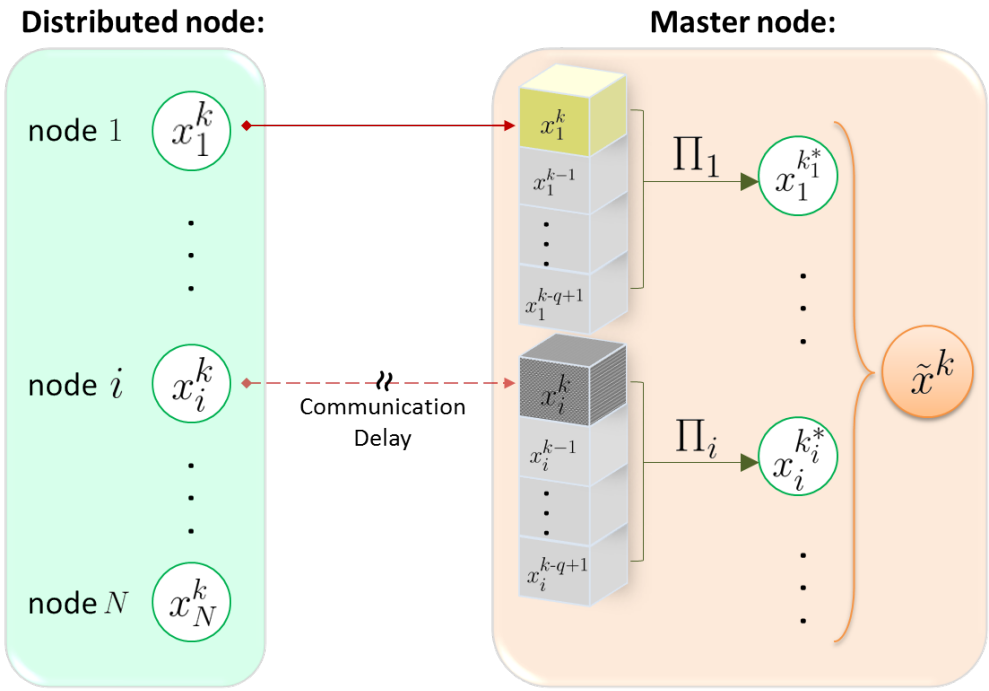
$$\begin{aligned} y^{k+1} &:= y^k + \alpha^k(A\tilde{x}^{k+1} - b) \\ &= y^k + \sum_{i=1}^{N}\left(\alpha_i^k A_i \tilde{x}_i^{k+1} - \frac{1}{N}\alpha_i^k b\right), \end{aligned} \tag{5.12}$$

where α_i^k is the step size for the index i.

Although α_i^k may vary at each time step, we let α_i^k be a constant value, denoted by α_i, for simplicity. Hence, it satisfies that $\alpha := \sum_{i=1}^{N}\alpha_i$, which is a fixed value.

There are two different ways to update dual variable y. Throughout the chapter, we denote these two different cases as the *deterministic asynchronous algorithm* and the *stochastic asynchronous algorithm*, respectively, in order to clarify and differentiate them. The deterministic asynchronous algorithm stands for the case where the variable k_i^* is considered as a constant value and is given by $k_i^* := k-q+1, \forall i$. Thus, it leads to $\tilde{x}^k := [x_1^{k-q+1^\top}, x_2^{k-q+1^\top}, \ldots, x_N^{k-q+1^\top}]$. In this case, it is assumed that the value x_i^{k-q+1}, which is a q-step prior value of x_i^k, is always available to the master

node. In other words, all delays are assumed to be bounded by the finite value q. Therefore, one can proceed with y-update, given in (5.12), without synchronization when applying the deterministic asynchronous algorithm. Note that there is no randomness in the deterministic asynchronous algorithm. Although this deterministic case obviates the unnecessary idle time by avoiding synchronization, it always utilizes



q : the maximum possible delay

x_i^k : the value of x_i at time k

$x_i^{k_i^*}$: the random variable such that $x_i^{k_i^*} \in \{x^k, x^{k-1}, \ldots, x^{k-q+1}\}$

$\Pi_i := [(\pi_1)_i, (\pi_2)_i, \ldots, (\pi_q)_i]$, where $(\pi_j)_i$, $j = 1, 2, \ldots, q$, stands for the modal probability for $x_i^{k_i^*}$

$\tilde{x}^k := [x_1^{k_1^*\top}, x_2^{k_2^*\top}, \ldots, x_N^{k_N^*\top}]^\top$

Figure 5.2: The schematic of the stochatic asynchronous algorithm in the distributed quadratic programming. In this figure, the maximum delay is bounded by $k-q+1 \leq k_i^* \leq k$, $\forall i$. Each node has the probability Π_i to represent random delays.

q-step prior values saved in the buffer memory. In the real implementation of the distributed optimization, however, k_i^* varies from distributed nodes and also changes over each iteration. Thus, we consider another case by letting $x_i^{k_i^*}$ as a random vector, where k_i^* becomes one of the values in the given set $\{k, k-1, \ldots, k-q+1\}$. To distinguish this case with the deterministic asynchronous algorithm, it is referred to as the stochastic asynchronous algorithm.

Fig. 6.1 describes the conceptual schematic of the stochastic asynchronous algorithm using the dual decomposition in QP problem. Depending on the processing capability and load balance in distributed nodes, the value for x_i^k is available or not in the master node at each iteration step. We assume that this delay is bounded by the finite value q. To describe the randomness of such delays, we adopt a probability $\Pi_i := [(\pi_1)_i, (\pi_2)_i, \ldots, (\pi_q)_i] \in \mathbb{R}^{1\times q}$ that predicts which value for x_i^k will be used to update y^k as shown in Fig. 6.1.

Starting from (5.12), with the definition of the set $\mathcal{S}^k := \{k_i^* | k_i^* = k\}$ and the symbol $\Phi_i := -\alpha_i A_i Q_i^{-1} A_i^\top$, the state dynamics of the stochastic asynchronous algo-

rithm is then given by

$$
\begin{aligned}
y^{k+1} &= y^k + \sum_{i=1}^{N} \left(\alpha_i A_i \tilde{x_i}^{k+1} - \frac{1}{N} \alpha_i^k b \right) \\
&= y^k + \sum_{i \in \mathcal{S}^{k+1}} \alpha_i A_i x^{k+1} + \sum_{i \in \mathcal{S}^k} \alpha_i A_i x^k + \cdots \\
&\qquad + \sum_{i \in \mathcal{S}^{k-q+2}} \alpha_i A_i Q_i^{-1} A_i^\top x^{k-q+2} + \left(\sum_{i=1}^{N} -\frac{1}{N} \alpha_i b \right) \\
&= \left(I - \sum_{i \in \mathcal{S}^{k+1}} \Phi_i \right) y^k - \left(\sum_{i \in \mathcal{S}^k} \Phi_i \right) y^{k-1} - \cdots \qquad \text{(by (5.10))} \\
&\qquad - \left(\sum_{i \in \mathcal{S}^{k-q+2}} \Phi_i \right) y^{k-q+1} + \left(\sum_{i=1}^{N} -\alpha_i A_i Q_i^{-1} c - \frac{1}{N} \alpha_i b \right).
\end{aligned}
\tag{5.13}
$$

The above equation is simplified by the following definitions, given by

$$
R_i(k) := \sum_{j \in \mathcal{S}^{k-i+2}} \Phi_j, \tag{5.14}
$$

$$
B := \left(\sum_{i=1}^{N} -\alpha_i A_i Q_i^{-1} c - \frac{1}{N} \alpha_i b \right), \tag{5.15}
$$

resulting in

$$
y^{k+1} = \left(I - R_1(k) \right) y^k - R_2(k) y^{k-1} - \cdots - R_q(k) y^{k-q+1} + B, \tag{5.16}
$$

where the time-varying matrix $R_i(k)$ completely depends on the value k_i^* that is a random event.

As described in [7], it is a very challenging task to analyze the stochastic asynchronous algorithm (see page 101, chapter 1). The primary goal of this chapter is, therefore, to analyze not only the convergence but also the rate of that for the

stochastic asynchronous algorithm which brings stochastic process for the state y^k. For this purpose, we adopt a *switched linear system (or jump linear system, interchangeably)* framework that will be introduced in the next section in more detail.

5.3 A Switched System Approach

In order to solve the dual decomposition problem with random delays in distributed nodes, we define a new augmented state $Y^k := [{y^k}^\top, {y^{k-1}}^\top, \ldots, {y^{k-q+1}}^\top]^\top$. Then, one can define the following recursive dynamics:

$$\underbrace{\begin{bmatrix} y^{k+1} \\ y^k \\ y^{k-1} \\ \vdots \\ y^{k-q+2} \end{bmatrix}}_{=Y^{k+1}} = \underbrace{\begin{bmatrix} I - R_1(k) & -R_2(k) & -R_3(k) & \cdots & -R_q(k) \\ I & 0 & \cdots & & 0 \\ 0 & I & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & I & 0 \end{bmatrix}}_{=W(k)} \underbrace{\begin{bmatrix} y^k \\ y^{k-1} \\ y^{k-2} \\ \vdots \\ y^{k-q+1} \end{bmatrix}}_{=Y^k} + \underbrace{\begin{bmatrix} B \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{=C}, \tag{5.17}$$

where I and 0 are identity and zero matrices with proper dimensions, respectively. Consequently, the above recursive equation ends up with the following simple form:

$$\Rightarrow Y^{k+1} = W(k)Y^k + C$$

In fact, the structure of the time-varying matrix $W(k)$ is not arbitrary, but it has a finite number of forms, given by q^N, which counts all possible scenarios to distribute N numbers of Φ_i, $i = 1, 2, \ldots, N$, matrices into the finite number of q. In the switched system, this number is referred to as the *"switching mode number"*, and we particularly denote this number with the symbol m. For instance, when $q = 2$

and $N = 2$, the switching mode number is given by $m = 2^2 = 4$. Thus, at each time k, the matrix $W(k)$ has one of the following form:

$$W_1 = \begin{bmatrix} I - \Phi_1 - \Phi_2 & 0 \\ I & 0 \end{bmatrix}, \qquad W_2 = \begin{bmatrix} I - \Phi_1 & -\Phi_2 \\ I & 0 \end{bmatrix},$$
$$W_3 = \begin{bmatrix} I - \Phi_2 & -\Phi_1 \\ I & 0 \end{bmatrix}, \qquad W_4 = \begin{bmatrix} I & -\Phi_1 - \Phi_2 \\ I & 0 \end{bmatrix}.$$

Then, only one out of all set of matrices $\{W_r\}_{r=1}^m$ will be used at each time k to update the system state Y^k, which results in the *switched linear system* structure as follows.

Consider the switched system:

$$Y^{k+1} = W_{\sigma_k} Y^k + C, \quad \sigma_k \in \{1, 2, \ldots, m\},\ k \in \mathbb{N}_0, \tag{5.18}$$

where $\{\sigma_k\}_{k=0}^\infty$ denotes the switching sequence that describes how the asynchrony takes place. Then, the switching probability $\Pi(k) := \Pi_1(k) \otimes \Pi_2(k) \otimes \cdots \otimes \Pi_N(k) = [\pi_1(k), \pi_2(k), \ldots, \pi_m(k)]$, where $\Pi_i(k)$ represents the probability for $x_i^{k_i^*}$ as depicted by Fig. 6.1, determines which mode σ_k will be utilized at each time step. (Note that $\Pi_i(k)$ and hence $\Pi(k)$ are not necessarily to be stationary.) In this case, the switched linear system is named by "stochastic switched linear system" or "stochastic jump linear system" [64] because the switching is a stochastic process. The benefit when applying this stochastic switched linear system structure is that the delay in the asynchronous algorithm is naturally taken into account by the switched system framework. Hence, the randomness of the asynchronous algorithm is represented by a certain switching logic.

Remark 5.1 ***(Computational complexity due to an extremely large number of the switching modes)*** *Although the stochastic switched linear system framework is suitable for modeling the dynamics of the stochastic asynchronous algorithm in distributed QP problems, it results in an extremely large number of the switching modes, causing computational complexity. For instance, even if* $q = 2$ *and* $N = 20$*, we have* $m = q^N = 2^{20}$*, and it is impractical to store such large numbers of matrices in the real implementation. Therefore, it is necessary to develop proper methods to analyze the stochastic asynchronous algorithm using the switched linear system without any concerns for such computational complexity issues.*

To avoid the computational complexity problems stated above, we make following assumptions for analysis of both the convergence and the rate of the convergence for the stochastic asynchronous algorithm:

- **Assumption 3.1.** We consider the random delays that occur during the computation of x_i^k at each node. In this case, the probability $\Pi_i(k) = [(\pi_1(k))_i, (\pi_2(k))_i, \ldots, (\pi_q(k))_i]$ describes which value for $x_i^{k_i^*}$ will be used among the given set $\{x_i^k, x_i^{k-1}, \ldots, x_i^{k-q+1}\}$. Then, we assume that each modal probability $(\pi_j(k))_i$ is *stationary*, and hence $\Pi_i(k)$ is also stationary in time.

Under the Assumption 3.1., the switching probability $\Pi(k) := \Pi_1 \otimes \Pi_2 \otimes \cdots \otimes \Pi_N$ becomes stationary. For this case, the jump linear system with the given dynamics in (5.18) is termed as the independent, identically distributed (i.i.d.) jump linear system. Since the modal switching probability π_r is a probability, it satisfies $0 \leq \pi_r \leq 1$, $\forall r$ and $\sum_{r=1}^m \pi_r = 1$. This stationary occupation probability rules which system matrix W_r will be used at each instance. The implementation of the switching sequence $\{\sigma_k\}$, governed by Π, describes the randomness for the stochastic asynchronous algorithm *in an average sense.*

5.4 Convergence Analysis

In this section, the convergence of the state Y^k for the stochastic asynchronous model will be studied under the switched system framework. For several decades, the stability results for the switched systems with stochastic jumping parameters have been well established, for example, in the literature [63], [64], [34]. However, these methods are inapplicable to the asynchronous computing algorithm with massive parallelism because it results in extremely large numbers of switching modes, leading to *computational complexity* as explained in Remark 5.1. Therefore, we aim to investigate the convergence and the rate of convergence for the asynchronous algorithm without any concerns for such computational complexity issues. Particularly, this section will provide a convergence condition for the stochastic asynchronous algorithm in distributed QP problems.

Before proceeding further to investigate the asynchronous model, we analyze the convergence of the *synchronous* case *without delays* for a reference. Since in the synchronous algorithm all values are synchronized after each iteration, no delays occur when updating the state dynamics. Then, the state $Y^k_{\text{sync.}}$ for the synchronous case is governed by the following recursive equation:

$$\underbrace{\begin{bmatrix} y^{k+1} \\ y^k \\ y^{k-1} \\ \vdots \\ y^{k-q+2} \end{bmatrix}}_{=Y^{k+1}_{\text{sync.}}} = \underbrace{\begin{bmatrix} I-R & 0 & 0 & \cdots & 0 \\ I & 0 & \cdots & & 0 \\ 0 & I & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & I & 0 \end{bmatrix}}_{=W_{\text{sync.}}} \underbrace{\begin{bmatrix} y^k \\ y^{k-1} \\ y^{k-2} \\ \vdots \\ y^{k-q+1} \end{bmatrix}}_{=Y^k_{\text{sync.}}} + \underbrace{\begin{bmatrix} B \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{=C}, \tag{5.19}$$

where the matrix $R := \sum_{i=1}^{q} R_i(k) = \sum_{j=1}^{N} \Phi_j$ is time-invariant.

Since R is a constant matrix in above equation, the matrix $W_{\text{sync.}}$ is also time-invariant, and the steady-state value of $Y^{\star}_{\text{sync.}} := \lim_{k\to\infty} Y^{k}_{\text{sync.}}$, is obtained by

$$\begin{aligned} Y^{\star}_{\text{sync.}} &= W_{\text{sync.}} Y^{\star}_{\text{sync.}} + C. \qquad (5.20)\\ \Rightarrow Y^{\star}_{\text{sync.}} &= (I - W_{\text{sync.}})^{-1} C, \qquad \big(\text{if } (I - W_{\text{sync.}}) \text{ is non-singular}\big) \end{aligned}$$

if the condition $\rho(W_{\text{sync.}}) < 1$ holds.

However, the state in the i.i.d. switched linear system that represents the stochastic asynchronous model, evolves with the dynamics given in (5.18), where the matrix W_{σ_k} is determined by the switching probability Π. Thus, the state of the asynchronous model becomes a random vector, obstructing the convergence analysis of the stochastic asynchronous model. For the stochastic switched systems, various convergence (stability) notions have been developed [34], to guarantee the system stability. Among different convergence notions, we will focus on the mean square convergence, defined below.

Definition 5.1 *(Definition 1.1, [32]) The switched system is said to be mean square stable (convergent) if for any initial condition x_0 and arbitrary initial probability distribution $\Pi(0)$, $\lim_{k\to\infty} \mathbb{E}\big[||x(k, x_0) - x^{\star}||^2\big] = 0$, where $x^{\star}$ is the fixed-point value of x^k, i.e. $\lim\limits_{k\to\infty} x^k = x^{\star}$.*

The necessary and sufficient condition for the mean square convergence of the i.i.d. jump linear systems is described as follows:

Proposition 5.1 *(Corollary 2.7, [33]) Consider an i.i.d. jump linear system, where $\Pi(k)$ is a stationary probability vector $\{\pi_1, \pi_2, \cdots, \pi_m\}$ for all k. Then, the i.i.d.*

jump linear system is mean square stable (convergent) if and only if the matrix $\sum_{j=1}^{m} \pi_j \left(W_j \otimes W_j\right)$ *is Schur stable, i.e.*

$$\rho\left(\sum_{j=1}^{m} \pi_j \left(W_j \otimes W_j\right)\right) < 1. \tag{5.21}$$

Once again, massive parallelism results in large m, causing computational intractability. Thus, implementation of Proposition 5.1 is unfeasible to analysis of asynchronous distributed and parallel QP problems with massively parallel computing algorithm because the equation in (5.21) requires the summation over index i from 1 up to m. In order to avoid this problem, we provide Algorithm 1.

Algorithm 1

$k_i^* \leftarrow$ one of the values in $\{k, k-1, \ldots, k-q+1\}$ with probability Π_i.
$\xi \leftarrow k$
for $i \leq N$ **do**
    if $\xi \leq k_i^*$ **then**
        $\xi \leftarrow k_i^*$.
        $i \leftarrow i + 1$.
    end if
end for
$\tilde{x}^k \leftarrow [(x_1^{\xi})^\top, (x_2^{\xi})^\top, \ldots, (x_N^{\xi})^\top]^\top$

By executing Algorithm 1 at every time step in the master node, the random vector $\tilde{x}^k$ has the following form: $\tilde{x}^k = [(x_1^{\xi})^\top, (x_2^{\xi})^\top, \ldots, (x_N^{\xi})^\top]^\top$, where ξ denotes the oldest time among the recently updated values across the index $i = 1, 2, \ldots, N$. For example, if $k_i^* = k - 2$ for some i is the oldest value over all k_i^*, $i = 1, 2, \ldots, N$, then we have $\tilde{x}^k = [(x_1^{k-2})^\top, (x_2^{k-2})^\top, \ldots, (x_N^{k-2})^\top]^\top$. In this case, the modal matrix

W_r has the same structure with $W(k)$, given in (5.17), where $R_i(k)$ satisfies

$$R_i(k) = \begin{cases} R, & (\text{if } i = k - \xi + 1) \\ 0. & (\text{otherwise}) \end{cases}$$

The utilization of Algorithms 1 then drastically reduces the switching mode number by q regardless of the value N, due to the fact that at each iteration step we intentionally use the oldest updated value saved in buffer memory. For example, when $q = 2$, the matrix W_{σ_k} becomes one of the following form:

$$W_1 = \begin{bmatrix} I - R & 0 \\ I & 0 \end{bmatrix}, \qquad W_2 = \begin{bmatrix} I & -R \\ I & 0 \end{bmatrix}.$$

Since Algorithm 1 works as if it aggregates some subsets of the given switching modes, we need to redefine the switching probability Π accordingly. Then, Π is obtained by the following Theorem.

Theorem 5.1 *Consider the i.i.d. switched linear system given in* (5.18) *with the switching probability* $\Pi = \Pi_1 \otimes \Pi_2 \otimes \ldots \otimes \Pi_N \in \mathbb{R}^{1 \times q^N}$. *After the implementation of Algorithm 1, the switching probability is redefined by* $\Pi := [\pi_1, \pi_2, \ldots, \pi_q] \in \mathbb{R}^{1 \times q}$, *of which modal probability* π_i *has the following form:*

$$\pi_r := \prod_{i=1}^{N} \left(\sum_{j=1}^{r} (\pi_j)_i \right) - \left(\sum_{j=1}^{r-1} \pi_j \right), \quad r = 1, 2, \ldots, q, \tag{5.22}$$

where the term $(\pi_j)_i$ *denotes* j^{th} *modal probability for* Π_i (*i.e.,* $\Pi_i = [(\pi_1)_i, (\pi_2)_i, \ldots, (\pi_q)_i]$).

Proof For simplicity of the proof, we assume that $N = 2$. The most general case

is then proved similarly by induction. In this case, the master node takes the values for each $x_i^{k_i^*}$ according to the probability Π_i, $i = 1, 2$, which are given by

$$\Pi_1 = [(\pi_1)_1, (\pi_2)_1, \ldots, (\pi_q)_1],$$
$$\Pi_2 = [(\pi_1)_2, (\pi_2)_2, \ldots, (\pi_q)_2].$$

We let the index $j \in \{k, k-1, \ldots, k-q+1\}$ be the value explained in Algorithm 1.

When $j = 1$, the modal switching probability π_1 is obtained by

$$\begin{aligned}
\pi_1 &= \mathbf{Pr}\Big(k_1^* = k,\ k_2^* = k\Big) \\
&= \mathbf{Pr}\Big(k_1^* = k\Big) \times \mathbf{Pr}\Big(k_2^* = k\Big) \qquad \text{(since } k_1^* \text{ and } k_2^* \text{ are independent)} \\
&= (\pi_1)_1 \times (\pi_1)_2.
\end{aligned}$$

Similarly, when $j = 2$, we have

$$\begin{aligned}
\pi_2 &= \mathbf{Pr}\Big(k_1^* \in \{k, k-1\},\ k_2^* \in \{k, k-1\}\Big) - \pi_1 \\
&= \sum_{j=1}^{2} \mathbf{Pr}\Big(k_1^* = k-j+1\Big) \times \sum_{j=1}^{2} \mathbf{Pr}\Big(k_2^* = k-j+1\Big) - \pi_1 \\
&= \left(\sum_{j=1}^{2} (\pi_j)_1\right) \times \left(\sum_{j=1}^{2} (\pi_j)_2\right) - \pi_1.
\end{aligned}$$

In the first line of above equation, we have to extract π_1 because it corresponds to the case when $j = 1$.

For any arbitrary value j satisfying $j \in \{k, k-1, \ldots, k-q+1\}$, the switching

probability is therefore obtained by induction as follows:

$$\begin{aligned}\pi_r &= \mathbf{Pr}\Big(k_1^* \in \{k, k-1, \ldots, k-r+1\},\ k_2^* \in \{k, k-1, \ldots, k-r+1\}\Big) - \sum_{j=1}^{r-1} \pi_j \\ &= \left(\sum_{j=1}^{r} (\pi_j)_1\right) \times \left(\sum_{j=1}^{r} (\pi_j)_2\right) - \sum_{j=1}^{r-1} \pi_j.\end{aligned}$$

Thus, the most general case with $q, N \in \mathbb{N}$ can be induced as follows:

$$\pi_r = \prod_{i=1}^{N} \left(\sum_{j=1}^{r} (\pi_j)_i\right) - \left(\sum_{j=1}^{r-1} \pi_j\right), \quad r = 1, 2, \ldots, q.$$

□

For comparison, the switching mode number without the proposed algorithm is given by $m = q^N$ of which growth is exponential with respect to N, whereas with the proposed Algorithm 1, it is given by $m = q$ that is a constant value *irrespective of* N. Thus, by leveraging the proposed algorithm, one can apply the mean square convergence condition given in Proposition 5.1, to test the stability of the stochastic asynchronous algorithm. Note that the implementation of Proposition 5.1 was computationally intractable without Algorithm 1 due to the large numbers in m. Consequently, the proposed algorithm enables the convergence analysis of the stochastic asynchronous parallel computing algorithm in QP problems.

Once the condition (5.21) is guaranteed with a given i.i.d. switching probability Π by implementing Algorithm 1, the steady-state (fixed-point) value $Y^\star := \lim_{k\to\infty} Y^k$, where Y^k is the state for the stochastic asynchronous algorithm of which dynamics

is given in (5.18), can be obtained according to Definition 5.1 and is given by

$$Y^\star = W_{\sigma_k} Y^\star + C. \tag{5.23}$$
$$\Rightarrow Y^\star = (I - W_{\sigma_k})^{-1} C.$$

Interestingly, $Y^\star$ becomes a unique vector, regardless of σ_k that changes over time, due to the inherent structure in matrices W_{σ_k} and C, which results in $Y^\star = Y^\star_{\text{sync.}}$, where $Y^\star_{\text{sync.}}$ is defined in (5.20). Therefore, the state for the stochastic asynchronous algorithm, denoted by Y^k, converges to the unique, identical fixed-point value $Y^\star$, if the condition (5.21) holds.

5.5 Rate of Convergence Analysis

Since the rate of convergence provides information regarding how fast each scheme converges to the fixed-point value, it works as a guideline that suggests which methods will solve the given QP problem faster than other schemes. Therefore, the comparison for the rate of convergence between different schemes is advantageous in terms of estimating the time to obtain an optimal solution for the QP problem. Although asynchronous algorithms are considered to be more time-efficient for obtaining an optimal solution, it is not analytically proved yet what is the rate of convergence. Therefore, in this section we investigate the rate of convergence for three different algorithms (e.g., synchronous, deterministic asynchronous, and stochastic asynchronous algorithms) in analytic form.

i) Synchronous algorithm with delays:

For synchronous scheme, Y^k is updated after a certain amount of time due to the idle time for synchronization. As described in Fig. 5.1, we assume that all data from distributed nodes arrive at the master node within a bounded time q. In this case,

idle process time for the synchronization is given by q and Y^k can be updated at every $t(q+1)$ time step, where $t \in \mathbb{N}_0$. Consequently, at each time step, Y^k-update is given by

$$\begin{aligned}
&\text{at time } t = 1: && Y^{(q+1)} = W_{\text{sync.}} Y^0 + C \\
&\text{at time } t = 2: && Y^{2(q+1)} = W_{\text{sync.}} Y^{(q+1)} + C \\
&\text{at time } t = 3: && Y^{3(q+1)} = W_{\text{sync.}} Y^{2(q+1)} + C \\
&\qquad \vdots && \qquad \vdots \\
&\text{at arbitrary time } t+1: && Y^{(t+1)(q+1)} = W_{\text{sync.}} Y^{t(q+1)} + C, \quad t \in \mathbb{N}_0
\end{aligned}$$

Now, we consider the term $||Y^k - Y^\star||_\infty$ in order to investigate the rate of convergence for the synchronous algorithm. Then, from the dynamics for synchronous case, given by $Y^k = W_{\text{sync.}} Y^{k-1} + C$, we have

$$\begin{aligned}
||Y^k - Y^\star||_\infty &= ||W_{\text{sync.}} Y^{k-1} + C - Y^\star||_\infty \\
&= ||W_{\text{sync.}} Y^{k-1} - W_{\text{sync.}} Y^\star||_\infty && \big(\text{by (5.20)}\big) \\
&= ||W_{\text{sync.}} \left(W_{\text{sync.}} Y^{k-2} + C\right) - W_{\text{sync.}} Y^\star||_\infty \\
&= ||(W_{\text{sync.}})^2 Y^{k-2} + W_{\text{sync.}}(C - Y^\star)||_\infty \\
&= ||(W_{\text{sync.}})^2 \left(Y^{k-2} - Y^\star\right)||_\infty && \big(\text{by (5.20)}\big) \\
&\qquad \vdots \\
&= ||(W_{\text{sync.}})^k \left(Y^0 - Y^\star\right)||_\infty \\
&\leq ||(W_{\text{sync.}})^k||_\infty \cdot ||Y^0 - Y^\star||_\infty,
\end{aligned}$$

where $k = t(q+1)$, $t \in \mathbb{N}_0$. Thus, we have the upper bound of the rate of convergence

for the synchronous algorithm as follows:

$$||Y^k - Y^\star||_\infty \leq ||(W_{\text{sync.}})^k||_\infty \cdot ||Y^0 - Y^\star||_\infty, \quad k = t(q+1),\ t \in \mathbb{N}_0. \tag{5.24}$$

ii) Deterministic asynchronous algorithm:

As described in section 2.3., the deterministic asynchronous algorithm takes advantage of the q step prior value instead of waiting for all x_i values being gathered in the master node for synchronization. In this case, the system dynamics for the deterministic asynchronous scheme is given by

$$Y^{k+1} = W_{\text{det.async.}} Y^k + C,$$

where the matrix $W_{\text{det.async.}}$ is defined as

$$W_{\text{det.async.}} := \begin{bmatrix} I & 0 & 0 & \cdots & -R \\ I & 0 & \cdots & & 0 \\ 0 & I & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & I & 0 \end{bmatrix}$$

because in this case we have $\forall i \in \mathcal{S}^{k-q+2}$ in (5.13) for the deterministic asynchronous algorithm, leading to above system dynamics.

Similarly to the process in obtaining (5.24), the upper bound of the rate of con-

vergence for the deterministic asynchronous algorithm is derived by

$$||Y^k - Y^\star||_\infty = ||(W_{\text{det.async.}})^k \left(Y^0 - Y^\star\right)||_\infty \tag{5.25}$$
$$\leq ||(W_{\text{det.async.}})^k||_\infty \cdot ||Y^0 - Y^\star||_\infty, \quad k \in \mathbb{N}_0.$$

iii) Stochastic Asynchronous algorithm:

Since the state Y^k becomes a random vector in the stochastic asynchronous case, the rate of convergence for $||Y^k - Y^\star||_\infty$ forms a distribution rather than a deterministic value, and is difficult to analyze such a distribution. Thus, we take the expectation for Y^k with respect to the i.i.d. switching probability Π, and investigate the rate of convergence for $||\mathbb{E}[Y^k] - Y^\star||_\infty$.

Under the assumption that the mean square convergence condition in Proposition 5.1 holds, the fixed-point value for Y^k is deterministically given by $Y^\star$, irrespective of Π. Therefore, it satisfies $\mathbb{E}[Y^\star] = Y^\star$. Taking the expectation in (5.23) results in $\mathbb{E}[Y^\star] = Y^\star = \mathbb{E}[W_{\sigma_k} Y^\star + C] = \mathbb{E}[W_{\sigma_k}] Y^\star + C = \mathbf{Pr}\left(\sum_{r=1}^q \pi_r W_r\right) Y^\star + C$. By defining a new matrix $\Lambda := \sum_{r=1}^q \pi_r W_r$, we end up with

$$\mathbb{E}[Y^\star] = Y^\star = \Lambda Y^\star + C. \tag{5.26}$$

Then, the term $||\mathbb{E}[Y^k] - Y^\star||_\infty$ becomes

$$
\begin{aligned}
||\mathbb{E}[Y^k] - Y^\star||_\infty &= ||\mathbb{E}[W_{\sigma_{k-1}} Y^{k-1} + C] + Y^\star||_\infty \\
&= ||\sum_{r=1}^{q} \mathbf{Pr}\left(W_{\sigma_{k-1}} Y^{k-1} + C | \sigma_{k-1} = r\right) \underbrace{\mathbf{Pr}\left(\sigma_{k-1} = r\right)}_{=\pi_r} - Y^\star||_\infty \\
&= ||\sum_{r=1}^{q} \pi_r W_r \mathbf{Pr}\left(Y^{k-1} | \sigma_{k-1} = r\right) + C - Y^\star||_\infty \\
&= ||\sum_{r=1}^{q} \pi_r W_r \mathbf{Pr}\left(Y^{k-1} | \sigma_{k-1} = r\right) - \Lambda Y^\star||_\infty \qquad \text{(by (5.26))} \\
&= ||\underbrace{\left(\sum_{r=1}^{q} \pi_r W_r\right)}_{=\Lambda} \sum_{s=1}^{q} \mathbf{Pr}\left(W_{\sigma_{k-2}} Y^{k-2} + C | \sigma_{k-2} = s\right) \pi_s - \Lambda Y^\star||_\infty \\
&= ||\Lambda \left(\sum_{s=1}^{q} \pi_s W_s \mathbf{Pr}\left(Y^{k-2} | \sigma_{k-2} = s\right) + C\right) - \Lambda Y^\star||_\infty \\
&= ||\Lambda \left(\sum_{s=1}^{q} \pi_s W_s \mathbf{Pr}\left(Y^{k-2} | \sigma_{k-2} = s\right)\right) + \Lambda C - \Lambda\left(\Lambda Y^\star + C\right)||_\infty \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by (5.26))} \\
&= ||\Lambda \left(\sum_{s=1}^{q} \pi_s W_s \mathbf{Pr}\left(Y^{k-2} | \sigma_{k-2} = s\right)\right) - \left(\Lambda\right)^2 Y^\star||_\infty \\
&\qquad\qquad\qquad \vdots \\
&= ||(\Lambda)^{k-1} \left(\sum_{t=1}^{q} \pi_t W_t \mathbf{Pr}\left(Y^0 | \sigma_0 = t\right) + C\right) - \left(\Lambda\right)^k Y^\star||_\infty \\
&= ||\left(\Lambda\right)^{k-1} \underbrace{\left(\sum_{t=1}^{q} \pi_t W_t\right)}_{=\Lambda} Y_0 - (\Lambda)^k Y^\star||_\infty \\
&= ||(\Lambda)^k \left(Y^0 - Y^\star\right)||_\infty \\
&\leq ||(\Lambda)^k||_\infty \cdot ||Y^0 - Y^\star||_\infty,
\end{aligned}
$$

where we used the law of total probability in above equations.

Therefore, the rate of convergence for the asynchronous scheme is given by:

$$||\mathbb{E}[Y^k] - Y^\star||_\infty \leq ||(\Lambda)^k||_\infty \cdot ||Y^0 - Y^\star||_\infty, \tag{5.27}$$

where with implementation of Algorithm 1 the matrix $\Lambda := \sum_{r=1}^{q} \pi_r W_r$ has the following form:

$$\Lambda = \begin{bmatrix} I - \pi_1 R & -\pi_2 R & \cdots & -\pi_q R \\ I & 0 & \cdot & 0 \\ 0 & I & \cdot & 0 \\ \vdots & \cdot & \ddots & \cdot \\ 0 & 0 & \cdots & I \end{bmatrix}, \quad R := \sum_{i=1}^{N} R_i(k) = \sum_{j=1}^{N} \Phi_j. \tag{5.28}$$

5.6 Numerical Example

In this section, we test the proposed asynchronous algorithms on distributed QP problems with dual decomposition technique. The system for the test bed is given by `Intel(R) Core(TM) i7-4710HQ CPU`, which has 4 cores with 8 threads (by Hyper-Threading Technology), with `8GM memory`. Although, the number of threads for this test bed is not very large, the system is enough to show the performance of proposed asynchronous computing algorithms for distributed QP with dual decomposition. We implemented parallel processing through `OpenMP` API (Application Program Interface) developed for direct multi-threaded, shared memory parallelism.

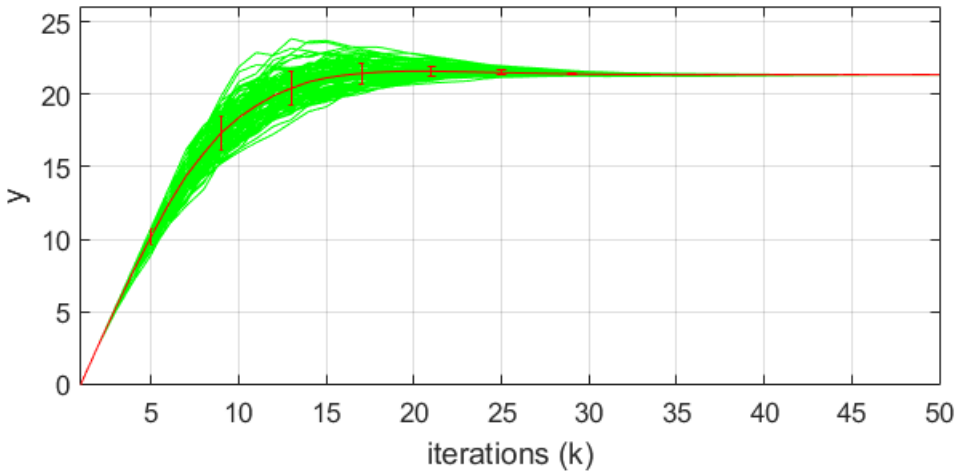


Figure 5.3: The convergence results for distributed quadratic programming with stochastic asynchronous algorithm. The (green) solid lines represent the state trajectory for y with total 100 Monte Carlo simulations (initial value was deterministically given by $y(0) = 2$ for all cases). The (red) solid-cross line denotes the mean and the standard deviation of multiple trajectories, respectively.

Let us consider the following distributed QP problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2} x_i^\top Q_i x_i + c_i^\top x_i \\ &\text{subject to} && A_i x_i \leq b_i, \qquad\qquad i = 1, 2, \ldots, N. \end{aligned}$$

The positive definite matrices Q_i, the matrices A_i, and the vectors c_i and b_i were generated by implementing PSEUDO RANDOM NUMBER generator in C++. The dimension of matrices and vectors are set to be: $Q_i \in \mathbb{R}^{n \times n}, A_i \in \mathbb{R}^{1 \times n}$, $c_i \in \mathbb{R}^{n \times 1}$, and $b_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$, where $n = 10$, $N = 20000$. Thus, computational burden for solving each distributed QP is low, whereas the total number of distributed QP is extremely high. We let the buffer length $q = 8$ and the step size $\alpha_i = 0.27$, $\forall i$.

For this type of massively distributed QP problem, the time for synchronization may become dominant in the total amount of time to solve QP. In this case, asyn-

chronous computing algorithms may lead to speedup by avoiding synchronization. We solved above distributed QP problem with the implementation of the proposed stochastic asynchronous algorithm. In Fig. 6.2, total 100 times of state trajectories for the dual variable y are given by (green) solid lines. Since y-update is stochastic process in the asynchronous algorithm, the trajectories are different from each other, resulting in the spread of the trajectories in the transient time. The i.i.d. switching probability Π_i that describes asynchronous computing for each distributed node is given by $(\pi_j)_i = \frac{e^{-3qj}}{\sum_{j=1}^{q} e^{-3qj}}$, $j = 1, 2, \ldots, q$, $\forall i$. Then, by Theorem 5.1 the switching probability for the switched system in (5.18), denoted by Π, is computed as $\Pi = \big[0, 0, 0.08, 0.8, 0.11, 0.01, 0, 0\big]$. For this i.i.d. switching probability, we calculated the spectral radius given in (5.21), which is $\rho\left(\sum_{j=1}^{m} \pi_j (W_j \otimes W_j)\right) = 0.6147 < 1$. Therefore, the convergence of the stochastic asynchronous algorithm is guaranteed in the mean square sense. The result in Fig. 6.2 also verifies the mean square convergence. The empirical mean and standard deviations are denoted by (red) solid line with cross mark and vertical bars, respectively. As the iteration step increases, the error of the mean square converges to zero (zero standard deviation).

Next, we predict the rate of convergence for three different schemes: i) synchronous case; ii) deterministic asynchronous case; iii) stochastic asynchronous case, in order to compare the performance. By employing the proposed results in section 5, we plotted the rate of convergence in Fig. 5.4. According to this result for the upper bound of the rate of convergence, the asynchronous algorithm is advantageous to speedup the total computation time for convergence of dual variable. This asynchronous scheme is up to 5 times faster than the synchronous algorithm and 2.5 times faster than the deterministic asynchronous algorithm, respectively.

In Fig. 5.5, we plotted actual computation time to find the optimal solution for three different schemes. For comparison purpose, the computation time for the

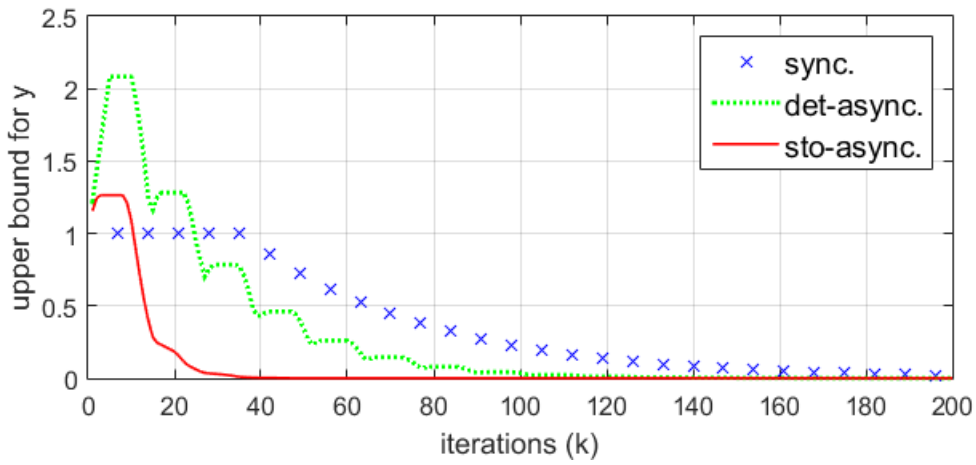


Figure 5.4: The rate of convergence results for distributed quadratic programming with three different schemes: synchronous (cross symbol); deterministic asynchronous (green dotted line); stochastic asynchronous (red solid line) algorithms.

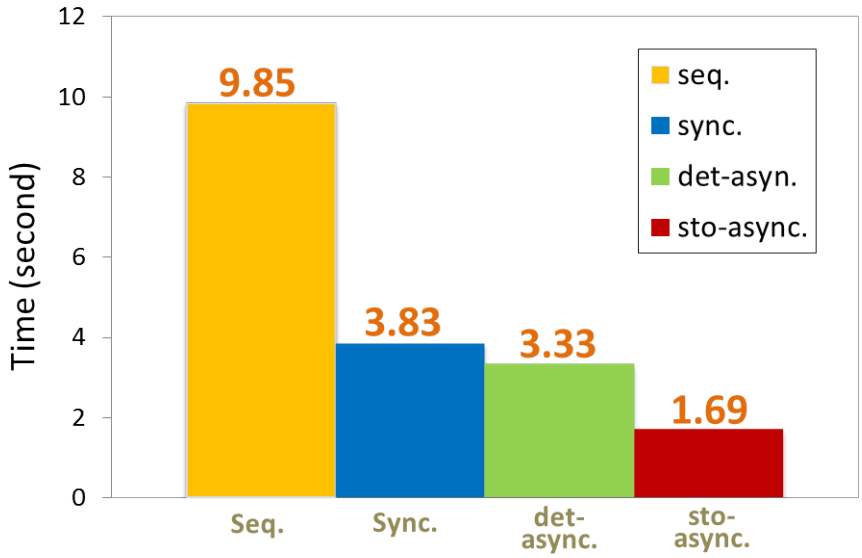


Figure 5.5: The convergence time comparison between sequential computing and three different schemes when the number of threads is given by 8 (maximum possible parallelization for the test bed.

sequential case is also given as a reference. The termination for the iteration is given by the residual tolerance $|y^k - y^{k-1}| \leq 10^{-5}$. As shown in Fig. 5.5, the proposed stochastic algorithm achieves the fastest convergence to solve the distributed QP problem. This result coincides with the result on the rate of convergence, which provides information regarding which schemes are the best to solve the given QP problem even before solving the optimization problem.

Table 5.1: Comparison of total computation time for the dual variable being convergent to the optimal value.

No. of Threads	Synchronous		Det-Asynchronous		Sto-Asynchronous	
	Time	Speedup	Time	Speedup	Time	Speedup
#2	5.2012s	1.89	7.8774s	1.25	4.4422s	2.22
#3	4.0189s	2.45	5.8558s	1.68	3.1259s	3.15
#4	3.3848s	2.91	4.8792s	2.02	2.6342s	3.74
#5	3.3511s	2.94	4.3913s	2.24	2.3071s	4.27
#6	3.3547s	2.94	3.8129s	2.58	2.0249s	4.86
#7	3.5891s	2.74	3.4590s	2.85	1.8351s	5.37
#8	3.8340s	2.57	3.3260s	2.96	1.6933s	5.81

For three different schemes, Table 5.1 presents the computation time to find the optimal solution as we increase the number of threads in the test bed. Also, we plotted speedup of three different schemes based on Table 5.1, by increasing the total number of threads. As the number of threads increases, the performance degradation occurred in the synchronous case, whereas the deterministic and stochastic asynchronous algorithms resulted in continuous speedup. When the number of threads is 8, the stochastic asynchronous algorithm led to 5.81 times speedup compared to the sequential computing, which is also 2.26 times faster than synchronous algorithms.

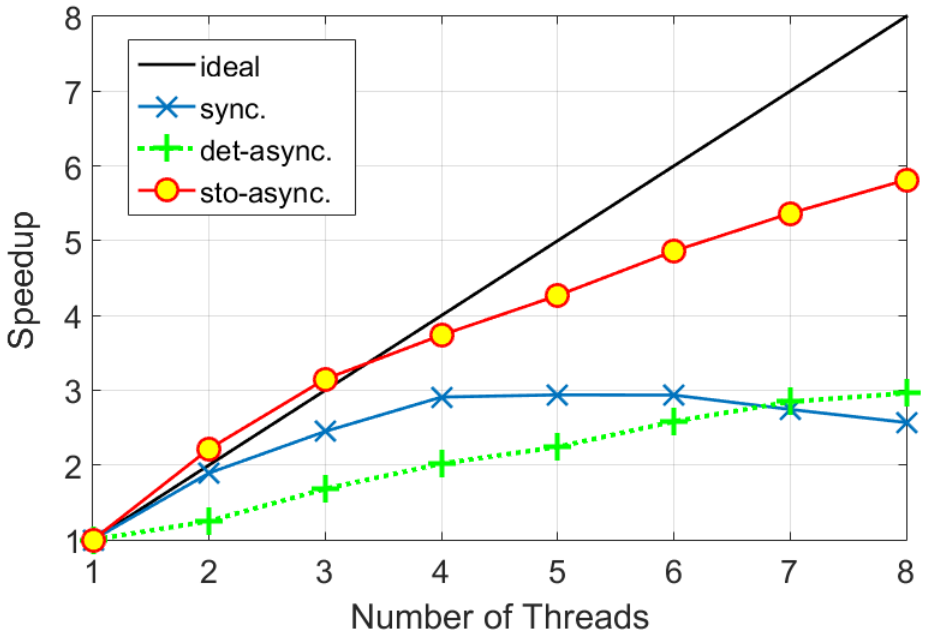


Figure 5.6: The speedup vs. numbers of threads.

As described in Remark 5.1, the computational complexity was the major concern when adopting the switched system framework for analysis of the stochastic asynchronous algorithm. To circumvent this complexity issue, we applied Algorithm 1. Thus, the number of switching modes has been drastically reduced from $q^N = 8^{20000}$ to $q = 8$, owing to Algorithm 1. Consequently, the analysis of stochastic asynchronous computing algorithm was carried out in a computationally efficient manner.

5.7 Concluding Remarks

In this chapter, we studied the convergence of asynchronous distributed QP problems via dual decomposition technique. To analyze the behavior of asynchrony in distributed and parallel computing, the switched system framework was introduced. Since the switching mode number becomes large for massively asynchronous comput-

ing algorithm, we developed a new algorithm, which drastically decreases mode number. By implementing the proposed method, the convergence condition in the mean square sense can be checked without any computational complexity issues. Also, we derived the rate of convergence for three different schemes (e.g., synchronous, deterministic asynchronous, and stochastic asynchronous algorithms), which analytically shows how fast dual variable converges to the optimal solution. The numerical example with an implementation of asynchronous distributed QP using OpenMP supports the validity of the proposed methods.

6. STABILITY OF LARGE-SCALE DISTRIBUTED NETWORKED CONTROL SYSTEMS WITH RANDOM COMMUNICATION DELAYS

6.1 Introduction

A networked control system (NCS) is a system that is controlled over a communication network. Recently, NCSs have attracted considerable research interests due to emerging networked control applications. For example, NCSs are broadly used in applications including traffic monitoring, networked autonomous mobile agents, chemical plants, sensor networks and distributed software systems in cloud computing architectures. Due to the communication network, communication delays or communication losses may occur, resulting in performance degradation or even instability. Therefore, it has led various researches to analyze NCSs associated with communication delays [16], [95], [102], [101], [78], [97], [64], [63]. Particularly in [97], the NCS with communication delays was analyzed by adopting the switched system [64], [63], [108], [34], [32], which refers to the dynamical system consists of a family of subsystems and a switching logic governing switching between subsystems.

In this chapter, we study large-scale distributed networked control system (DNCS), which denotes NCS with a large number of spatially distributed subsystems (or agents). For such large-scale systems, our primary goal is to analyze system stability when *random communication delays* exist. For decades, the system behavior with random communication delays has been widely modeled by the *Markov jump linear system* (MJLS) [97], [85], [84], [104], [86], [73] where the switching sequence is governed by the Markovian process. Since stability has been one of the major concerns, notable research studies have been reported for decades, in order to analyze the stability of the MJLS [90], [34], [17], [24], [106]. However, these results are

only applicable to the systems with a small number of switching modes, whereas the large-scale DNCSs in which we are particularly interested inevitably give rise to an extremely large number of switching modes. Thus, previous approaches developed for the stability analysis of MJLSs are inappropriate for large-scale DNCSs, due to the *computational intractability*. Although the literature [62] recently investigated the switched system that circumvents computation issues associated with a large number of switching modes, it is developed for the independent and identically distributed (i.i.d.) switching, which is not Markovian switching. Besides, we are also interested in large-scale DNCSs where the transition probabilities are inaccurately known as in [106], [107], [53] because, in practice it is difficult to accurately estimate the Markov transition probability matrix that models the random communication delays.

This chapter provides two key contributions to analyze the stability of large-scale DNCSs with random communication delays. Firstly, we guarantee the mean square stability of such systems by introducing a reduced mode model. We prove that the mean square stability for individual switched system implies a necessary and sufficient stability condition for the entire DNCS. This drastically reduces the number of modes necessary for analysis. Secondly, we present a new method to estimate the bound for uncertain Markov transition probability matrix, in order to guarantee the system stability. These results enable us to analyze large-scale systems in a computationally tractable manner.

Rest of this chapter is organized as follows. We introduce the problem for the large-scale DNCS in section 6.2. Section 6.3 presents the switched system framework for the stability analysis with communication delays. In Section 6.4, we propose the reduced mode model to efficiently analyze stability. Section 6.5 quantifies the stability region and bound for uncertain Markov transition probability matrix. This

is followed by the application of the proposed method to an example system in section 6.6, and we conclude the chapter with section 6.7.

Notation: The set of real numbers is denoted by $\mathbb{R}$. The symbols $\|\cdot\|$ and $\|\cdot\|_\infty$ stand for the Euclidean and infinity norm, respectively. The symbol $\#(\cdot)$ denotes the cardinality – the total number of elements in the given set. In addition, the symbols $\mathrm{tr}(\cdot)$, $\rho(\cdot)$, $\otimes$, and $\mathrm{diag}(\cdot)$ represent trace operator, spectral radius, Kronecker product, and block diagonal matrix operator, respectively.

6.2 Problem Formulation

6.2.1 Distributed Networked Control System with No Delays

Consider a discrete-time dynamics of each agent in the DNCS, given by:

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i} A_{ij} x_j(k), \quad i = 1, 2, \ldots, N, \tag{6.1}$$

where k is a discrete-time index, N is the total number of agents (subsystems), $x_i \in \mathbb{R}^n$ is a state for the i^{th} agent, $\mathcal{N}_i$ is a set of neighbors for x_i including the agent x_i itself, and $A_{ij} \in \mathbb{R}^{n \times n}$ is a time-invariant system matrix that represents the linear interconnections between agents. Note that we have $A_{ij} = 0$ if there is no interconnection between the agents i and j.

To represent the entire systems dynamics, we define the state $x(k) \in \mathbb{R}^{Nn \times Nn}$ as $x(k) \triangleq [x_1(k)^\top, x_2(k)^\top, \ldots, x_N(k)^\top]^\top$. Then, the system dynamics of the DNCS is given as

$$x(k+1) = \mathcal{A} x(k), \tag{6.2}$$

with the following definition for the matrix $\mathcal{A} \in \mathbb{R}^{Nn \times Nn}$

$$\mathcal{A} \triangleq \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1N} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2N} \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & A_{N3} & \cdots & A_{NN} \end{bmatrix},$$

$$A_{ij} = \begin{cases} 0, \text{ if no connection between the agents } i \text{ and } j, \\ A_{ij}, \text{ otherwise.} \end{cases}$$

For the discrete-time system in (6.2), it is well known that the system is stable if and only if the condition $\rho(\mathcal{A}) < 1$ is satisfied. We assume that the system (6.2), which is the case without communication delays is stable throughout the chapter. Then, we address the problem to analyze the stability in the presence of *random communication delays*. We remind the reader that N is very large.

6.2.2 DNCS with Communication Delays

Often, network communication between agents encounter time delays or packet losses while sending and receiving data. We denote the symbol τ as random communication delays and assume that τ has a discrete value bounded by $0 \leq \tau \leq \tau_d < \infty$, where τ_d is a finite-valued maximum delay. Then, the dynamics for the agent i with communication delays can be expressed as:

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i} A_{ij} x_j(k^*), \quad i = 1, 2, \ldots, N, \tag{6.3}$$

where $k^* \triangleq k - \tau$. Note that we have no communication delays when $i = j$ because there is no communication in this case.

The random communication delay, represented by the term k^*, forms a stochastic process. To analyze the stability of the DNCS, we define an augmented state $X(k)$ as $X(k) \triangleq [x(k)^\top, x(k-1)^\top, \cdots, x(k-\tau_d)^\top]^\top \in \mathbb{R}^{Nnq \times Nnq}$, where $q \triangleq \tau_d + 1$. Then, the dynamics for the entire system is given by

$$X(k+1) = W(k)X(k), \tag{6.4}$$

where $W(k) \triangleq \begin{bmatrix} \tilde{A}_1(k) & \tilde{A}_2(k) & \cdots & \tilde{A}_{q-1}(k) & \tilde{A}_q(k) \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \in \mathbb{R}^{Nnq \times Nnq}$,

the matrix I denotes an identity matrix with proper dimensions, and the time-varying matrices $\tilde{A}_j(k) \in \mathbb{R}^{Nn \times Nn}$, $j = 1, 2, \ldots, q$, model the randomness in the communication delays between neighboring agents.

6.3 Switched System Approach

Without loss of generality, the dynamics of the large-scale DNCS with communication delays in (6.4) can be transformed into a switched system framework as follows:

$$x(k+1) = W_{\sigma(k)}x(k), \quad \sigma(k) \in \{1, 2, \cdots, m\}, \tag{6.5}$$

where $W_{\sigma(k)}$ is the time-invariant matrix, representing communication delays in agents, $\{\sigma(k)\}$ is the switching sequence, and m is the total number of switching modes. When the switching sequence $\{\sigma(k)\}$ is stochastic, (6.5) is referred to as

a stochastic switched linear system or a stochastic jump linear system, according to the literature [64]. For the stochastic switched linear system, the switching sequence $\{\sigma(k)\}$ is governed by the mode-occupation switching probability $\pi(k) = [\pi_1(k), \pi_2(k), \ldots, \pi_m(k)]$, where π_i is a fraction number, satisfying $\sum_{i=1}^{m} \pi_i = 1$ and $0 \leq \pi_i \leq 1$, $\forall i$. In this case, each π_i denotes the modal probability corresponding to each mode dynamics W_i. In order to properly describe the behavior of random communication delays, it is necessary to adopt a certain switching logic, which is used to update the switching probability $\pi(k)$. For this purpose, the MJLS framework has been widely employed for decades as in [85], [84], [104], [86], [73]. Thus, we make the following assumption in our analysis.

- Assumption: Consider the stochastic jump linear system (6.5) with the switching probability $\pi(k) = [\pi_1(k), \pi_2(k), \ldots, \pi_m(k)]$. Then, $\pi(k)$ is updated by the Markovian process given by $\pi(k+1) = \pi(k)P$, where $P \in \mathbb{R}^{m \times m}$ is the Markov transition probability matrix.

Since the MJLS is a family of the stochastic switched linear system, various stability notions can be defined [34]. In this chapter, we will consider the mean square stability condition, defined below.

Definition 6.1 *(Definition 1.1 in [32]) The MJLS is said to be mean square stable if for any initial condition x_0 and arbitrary initial probability distribution $\pi(0)$,* $\lim_{k \to \infty} \mathbb{E}\left[||x(k, x_0)||^2\right] = 0$.

The total number of switching modes m depends on the size q and N. Since the communication delays take place independently while receiving and sending the data for each agent, m is calculated by counting all possible scenarios to distribute every matrices $A_{ij} \in \mathbb{R}^{n \times n}$ for $i \neq j$ in the block matrix $\mathcal{A} \in \mathbb{R}^{Nn \times Nn}$ given in (6.2),

into each $\tilde{A}_j(k) \in \mathbb{R}^{Nn \times Nn}$, $j = 1, 2, \ldots, q$, given in (6.4), which results in $m = q^{N(N-1)}$. Note that the large-scale DNCS that has considerably large N brings about extremely large m, which makes current analysis tools for the MJLS computationally intractable.

Before we further proceed, we introduce the following proposition that was developed for the stability analysis of the MJLS.

Proposition 6.1 *(Theorem 1 in [17]) The MJLS with the Markov transition probability matrix P is mean square stable if and only if*

$$\rho\left(\left(P^\top \otimes I\right) \operatorname{diag}(W_j \otimes W_j)\right) < 1, \tag{6.6}$$

where I is an identity matrix with a proper dimension,

$$\operatorname{diag}(W_j \otimes W_j) \triangleq \begin{bmatrix} (W_1 \otimes W_1) & 0 & 0 & \cdots & 0 \\ 0 & (W_2 \otimes W_2) & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & (W_{m\text{-}1} \otimes W_{m\text{-}1}) & 0 \\ 0 & 0 & \ldots & 0 & (W_m \otimes W_m) \end{bmatrix},$$

and m is the total number of the switching modes.

For the given set of matrices $\{W_{\sigma(k)}\}_{\sigma(k)=1}^{m}$ and the transition probability matrix P, one can always compute the spectral radius given in (6.6), and hence guarantee the system stability.

Unfortunately, this condition is not applicable to large-scale DNCSs due to enormously large m. For example, even with $q = 2$ and $N = 100$, we have $m = 2^{100 \times 99}$. It is not practically possible to compute the spectral radius in (6.6) for such problems.

To circumvent this scalability issue, we present a new analysis approach for such large-scale DNCSs in the next section.

6.4 Stability with Reduced Mode Dynamics

In this section, we define a new augmented state to reduce the mode numbers as follows:

$$\hat{x}_i(k) \triangleq [\tilde{x}_i(k)^\top,\, \tilde{x}_i(k-1)^\top,\, \cdots,\, \tilde{x}_i(k-\tau_d)^\top]^\top \in \mathbb{R}^{\hat{n}_i nq},$$

where $\hat{n}_i \triangleq \#(\mathcal{N}_i)$, $\mathcal{N}_i$ stands for the set of neighbors to x_i including itself, $\tilde{x}_i(k) \triangleq [x_i(k)^\top, x_j(k)^\top]^\top \in \mathbb{R}^{\hat{n}_i n}$, and $x_j(k) \in \mathbb{R}^n$, $j \in \mathcal{N}_i$, denotes all states that are neighbor to $x_i(k) \in \mathbb{R}^n$.

Then, we can construct a switched linear system framework similarly to (6.5) as follows:

$$\hat{x}_i(k+1) = \hat{W}^i_{\sigma_i(k)} \hat{x}_i(k), \quad \sigma_i(k) \in \{1, 2, \ldots, m_i\}, \tag{6.7}$$

where $\hat{W}^i_{\sigma_i(k)} \triangleq \begin{bmatrix} \hat{A}_1(k) & \hat{A}_2(k) & \cdots & \hat{A}_{q-1}(k) & \hat{A}_q(k) \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \in \mathbb{R}^{\hat{n}_i nq \times \hat{n}_i nq}$

with the time-varying matrix $\hat{A}_j(k) \in \mathbb{R}^{\hat{n}_i n \times \hat{n}_i n}$, $j = 1, 2, \ldots, q$. In this case, the total number of the switching modes for (6.7) is given by $m_i = q^{\hat{n}_i(\hat{n}_i - 1)}$.

By implementing the reduce mode model given in (6.7), we will provide a computationally efficient tool for the stability analysis of the original DNCS in the following theorem.

Theorem 6.1 *Consider the large-scale DNCS* (6.5) *with Markovian communication delays associated with the transition probability matrix P. The necessary and sufficient condition for the mean square stability of this system is then given by*

$$\rho\Big((P^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big) < 1, \quad \forall i = 1, 2, \ldots, N, \tag{6.8}$$

where $P^i \in \mathbb{R}^{m_i \times m_i}$ is the transition probability matrix for the reduced mode MJLS given in (6.7)*, I is an identity matrix with a proper dimension, N is the total number of the agents in the system, $m_i = q^{\hat{n}_i(\hat{n}_i-1)}$ is the total mode numbers for the reduce mode MJLS, and*

$$\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i) \triangleq \begin{bmatrix} (\hat{W}_1^i \otimes \hat{W}_1^i) & 0 & 0 & \cdots & 0 \\ 0 & (\hat{W}_2^i \otimes \hat{W}_2^i) & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & (\hat{W}_{m_i\text{-}1}^i \otimes \hat{W}_{m_i\text{-}1}^i) & 0 \\ 0 & 0 & \ldots & 0 & (\hat{W}_{m_i}^i \otimes \hat{W}_{m_i}^i) \end{bmatrix}.$$

Proof Let the matrix $Q^i(k)$ be of the form $Q^i(k) \triangleq \mathbb{E}[\hat{x}_i(k)\hat{x}_i(k)^\top]$. Then, $Q^i(k)$ is alternatively obtained by the following equation: $Q^i(k) = \sum_{s=1}^{m_i} Q_s^i(k)$, where $Q_s^i(k) \triangleq$

$\mathbb{E}\left[\hat{x}_i(k)\hat{x}_i(k)^\top|\sigma_i(k)=s\right]\pi_s^i(k)$, and $\pi_s^i(k) \triangleq \mathbf{Pr}\big(\sigma_i(k)=s\big)$. Then, $Q_s^i(k)$ satisfies

$$
\begin{aligned}
Q_s^i(k) &= \sum_{r=1}^{m_i} \mathbb{E}[\hat{x}_i(k)\hat{x}_i(k)^\top \mid \sigma_i(k)=s, \sigma_i(k-1)=r] \\
&\qquad\qquad \mathbf{Pr}(\sigma_i(k-1)=r \mid \sigma_i(k)=s)\pi_s^i(k) \\
&= \sum_{r=1}^{m_i} \mathbb{E}[\hat{x}_i(k)\hat{x}_i(k)^\top \mid \sigma_i(k)=s,\ \sigma_i(k-1)=r] \\
&\qquad\qquad \underbrace{\mathbf{Pr}(\sigma_i(k)=s \mid \sigma_i(k-1)=r)}_{\triangleq p_{rs}^i}\pi_r^i(k-1) \\
&= \sum_{r=1}^{m_i} p_{rs}^i\, \mathbb{E}[\hat{x}_i(k)\hat{x}_i(k)^\top \mid \sigma_i(k)=s, \sigma_i(k-1)=r]\pi_r^i(k-1) \\
&= \sum_{r=1}^{m_i} p_{rs}^i\, \mathbb{E}[\hat{W}_{\sigma_i(k-1)}^i \hat{x}_i(k-1)\hat{x}_i(k-1)^\top \hat{W}_{\sigma_i(k-1)}^{i^\top} \mid \sigma_i(k-1)=r]\pi_r^i(k-1) \\
&= \sum_{r=1}^{m_i} p_{rs}^i\, \hat{W}_r^i \underbrace{\mathbb{E}[\hat{x}_i(k-1)\hat{x}_i(k-1)^\top \mid \sigma_i(k-1)=r]\pi_r^i(k-1)}_{=Q_r^i(k-1)} \hat{W}_r^{i^\top} \\
&= \sum_{r=1}^{m_i} p_{rs}^i\, \hat{W}_r^i Q_r^i(k-1)\hat{W}_r^{i^\top}.
\end{aligned}
$$

In the second equality of above equation, p_{rs}^i denotes the mode transition probability from r to s in the Markov transition probability matrix P^i.

Taking the vectorization in above equation results in

$$
\begin{aligned}
\mathrm{vec}\left(Q_s^i(k)\right) &= \mathrm{vec}\left(\sum_{r=1}^{m_i} p_{rs}^i\, \hat{W}_r^i Q_r^i(k-1)\hat{W}_r^{i^\top}\right) \\
&= \sum_{r=1}^{m_i} p_{rs}^i \mathrm{vec}\left(\hat{W}_r^i Q_r^i(k-1)\hat{W}_r^{i^\top}\right) \\
&= \sum_{r=1}^{m_i} p_{rs}^i (\hat{W}_r^i \otimes \hat{W}_r^i)\mathrm{vec}(Q_r^i(k-1)).
\end{aligned}
$$

In the last equality, we used the property that $\mathrm{vec}(ABC) = (C^\top \otimes A)\mathrm{vec}(B)$.

Now We define a new variable $y^i_{(\cdot)}(k) \triangleq \text{vec}\Big(Q^i_{(\cdot)}(k)\Big)$, which leads to

$$y^i_s(k) = \sum_{r=1}^{m_i} p^i_{rs}(\hat{W}^i_r \otimes \hat{W}^i_r) y^i_r(k-1).$$

By stacking $y^i_{(\cdot)}(k)$ from 1 up to m_i, with a new definition for the augmented state $\hat{y}^i(k) \triangleq [y^i_1(k)^\top \, y^i_2(k)^\top \, \dots \, y^i_{m_i}(k)^\top]^\top$, we have the following recursion equation:

$$\hat{y}^i(k) = \underbrace{\begin{bmatrix} p^i_{11}(\hat{W}^i_1 \otimes \hat{W}^i_1) & p^i_{21}(\hat{W}^i_2 \otimes \hat{W}^i_2) & \dots & p^i_{m_i 1}(\hat{W}^i_{m_i} \otimes \hat{W}^i_{m_i}) \\ p^i_{12}(\hat{W}^i_1 \otimes \hat{W}^i_1) & p^i_{22}(\hat{W}^i_2 \otimes \hat{W}^i_2) & \dots & p^i_{m_i 2}(\hat{W}^i_{m_i} \otimes \hat{W}^i_{m_i}) \\ \vdots & \vdots & \ddots & \vdots \\ p^i_{1m_i}(\hat{W}^i_1 \otimes \hat{W}^i_1) & p^i_{2m_i}(\hat{W}^i_2 \otimes \hat{W}^i_2) & \dots & p^i_{m_i m_i}(\hat{W}^i_{m_i} \otimes \hat{W}^i_{m_i}) \end{bmatrix}}_{=(P^{i\top} \otimes I)\text{diag}(\hat{W}^i_j \otimes \hat{W}^i_j)} \underbrace{\begin{bmatrix} y^i_1(k-1) \\ y^i_2(k-1) \\ \vdots \\ y^i_{m_i}(k-1) \end{bmatrix}}_{=\hat{y}^i(k-1)}.$$

From the above equation, it is clear that $\rho\Big((P^{i\top} \otimes I)\text{diag}(\hat{W}^i_j \otimes \hat{W}^i_j)\Big) < 1$ implies $\lim_{k\to\infty} \hat{y}^i(k) = 0$, and hence this leads to $\lim_{k\to\infty} Q^i(k) = 0 \iff \lim_{k\to\infty} \text{tr}\left(Q^i(k)\right) = 0 \iff \lim_{k\to\infty} \mathbb{E}\left[||\hat{x}_i(k)||^2\right] = 0$, which is the sufficient mean square stability condition for $\hat{x}_i(k)$. On the other hand, if we have $\rho\Big((P^{i\top} \otimes I)\text{diag}(\hat{W}^i_j \otimes \hat{W}^i_j)\Big) > 1$, then $\hat{y}^i(k)$ will diverge, resulting in necessity for the mean square stability of $\hat{x}_i(k)$. Hence, the spectral radius being less than one is the necessary and sufficient mean square stability condition for the state $\hat{x}_i(k)$. Further, we have $\lim_{k\to\infty} \mathbb{E}\left[||\hat{x}_i(k)||^2\right] = 0$, $\forall i = 1, 2, \dots, N \iff \lim_{k\to\infty} \mathbb{E}\left[||x(k)||^2\right] = 0$, where $x(k)$ is the state for the DNCS defined in (6.5). This concludes the proof. □

Remark 6.1 *Theorem 6.1 provides an efficient way to analyze the stability for the large-scale DNCSs. The key idea stems from the hypothesis that the stability of each subsystem obtained by decomposing the original system will provide the stability of the entire system. Without any relaxation or conservatism, theorem 6.1 proved the*

necessary and sufficient condition for stability, which is equivalent to (6.6) *developed for the mean square stability of the MJLS. Compared to the total number of modes in the full state model* (6.5)*, which is* $q^{N(N-1)}$*, the reduced mode model* (6.7) *has total* $\sum_{i=1}^{N} q^{\hat{n}_i(\hat{n}_i-1)}$ *modes. Consequently, the growth of mode numbers in full state model is exponential with respect to* N^2*, whereas that in reduced mode model is* ***linear*** *with regard to* N*. Therefore, theorem 6.1 is computationally more efficient.*

6.5 Stability Region and Stability Bound for Uncertain Markov Transition Probability Matrix

The Markov transition probability matrix can be obtained from data of communication delays. In general, this Markov transition probability matrix obtained from statistics can be interpreted as representation of random communication delays in an average manner. Thus, one can not estimate the exact transition probability in practice, which leads to uncertainty in the Markov transition probability matrix. In this section, we aim at developing a tool to measure the stability bound for uncertain Markov transition probability matrix, in order to guarantee the stability of large-scale DNCSs with uncertain Markovian communication delays.

As explained in section 3, the dimension of the Markov transition probability matrix for the full state model is given by $P \in \mathbb{R}^{m\times m}$, where the number of switching modes is $m = q^{N(N-1)}$. For fairly large N, it is practically intractable to handle such a large-scale matrix. For example, even with $q = 2$ and $N = 100$, the mode number is given by $m = 2^{100\times 99}$, resulting in $P \in \mathbb{R}^{2^{100\times 99}\times 2^{100\times 99}}$. However, the reduced mode model developed in section 3 renders computation tractable because it results in huge decrease in the mode numbers as well as the dimension of each Markov transition probability matrix, by decomposing the original system into N numbers of reduced mode systems. As a consequence, the reduced mode model yields N numbers of new

Markov transition probability matrix P^i, $i = 1, 2, \ldots, N$, which has small size.

Here we assume that the new Markov transition probability matrix associated with the each reduced mode is given by the following form: $P^i = \bar{P}^i + \Delta P^i$, $i = 1, 2, \ldots, N$, where $\bar{P}^i$ is the nominal value and ΔP^i is the uncertainty in the Markov transition probability matrix for i^{th} subsystem. Due to the variation in ΔP^i, the system stability may change and hence we want to estimate the bound for ΔP^i, to guarantee the system stability. Here we assume that ΔP^i has the following structure:

$$\Delta P^i \triangleq \begin{bmatrix} \Delta p^i_{11} & \Delta p^i_{12} & \cdots & \Delta p^i_{1m_i} \\ \Delta p^i_{21} & \Delta p^i_{22} & \cdots & \Delta p^i_{2m_i} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta p^i_{m_i1} & \Delta p^i_{m_i2} & \cdots & \Delta p^i_{m_im_i} \end{bmatrix}, \in \mathbb{R}^{m_i \times m_i}$$

$$\text{s.t.} \sum_{s=1}^{m_i} \Delta p^i_{rs} = 0, \forall r = 1, 2, \ldots, m_i \tag{6.9}$$

Since we have a constraint such that the row sum has to be zero for ΔP^i in above equation, we aim to find the feasible maximum bound for each row, ε^i_r, satisfying the inequality $|\Delta p^i_{rs}| \leq \varepsilon^i_r$, $\forall r$, in order to guarantee the system stability. Then, each ε^i_r for $r = 1, 2, \ldots, m_i$ can be obtained by the following two steps.

Step 1: Solve Linear Programming (LP)

$$\text{maximize} \quad \mathbf{1}^\top z \qquad \text{(for upper bound)} \tag{6.10}$$

$$\Big(\text{or} \quad \text{minimize} \quad \mathbf{1}^\top z \qquad \text{(for lower bound)}\Big)$$

$$\text{subject to} \quad \mathbb{A}|z| < b_s, \quad \forall s = 1, 2, \ldots, m_i \tag{6.11}$$

$$lb_s \leq z_s \leq ub_s, \forall s = 1, 2, \ldots, m_i \tag{6.12}$$

where

$$z_s \triangleq [\Delta p^i_{1s},\ \Delta p^i_{2s},\ \cdots,\ \Delta p^i_{m_i s}]^\top,$$

$$z \triangleq [z_1^\top,\ z_2^\top,\ \cdots,\ z_{m_i}^\top]^\top,$$

$$\mathbb{A} \triangleq \Big[\alpha_1,\ \alpha_2,\ \cdots,\ \alpha_{m_i}\Big],\ \text{with}\ \alpha_j \triangleq \parallel \hat{W}^i_j \otimes \hat{W}^i_j \parallel_\infty,\ j = 1, 2, \ldots, m_i,$$

$$b_s \triangleq 1 - \sum_{r=1}^{m_i} \alpha_r \bar{p}^i_{rs},$$

$$lb_s \triangleq [-\bar{p}^i_{1s},\ -\bar{p}^i_{2s},\ \cdots\ -\bar{p}^i_{m_i s}]^\top,$$

$$ub_s \triangleq [1 - \bar{p}^i_{1s},\ 1 - \bar{p}^i_{2s},\ \cdots\ 1 - \bar{p}^i_{m_i s}]^\top.$$

The inequality constraint (6.11) in the LP problem guarantees the mean square stability by the forthcoming Lemma 6.1 and Theorem 6.2. The term lb_s and ub_s in (6.12) are the lower and upper bounds for z_s, respectively, according to $0 \leq p^i_{rs} = (\bar{p}^i_{rs} + \Delta p^i_{rs}) \leq 1$.

Step 2: Obtain Feasible Solution with Hyperplane Constraint

We can compute the feasible maximum bound for Δp^i_{rs} as follows.

$$\varepsilon^i_r = \min\left(\min(|\varepsilon^i_{r,\text{lb}}|), \min(\ |\varepsilon^i_{r,\text{ub}}|)\right), \quad r = 1, 2, \ldots, m_i. \tag{6.13}$$

where

$$\varepsilon^i_{r,\text{lb}} \triangleq [(\Delta p^i_{r1})^\star_{\text{lb}},\ (\Delta p^i_{r2})^\star_{\text{lb}},\ \ldots,\ (\Delta p^i_{rm})^\star_{\text{lb}}]^\top,$$

$$\varepsilon^i_{r,\text{ub}} \triangleq [(\Delta p^i_{r1})^\star_{\text{ub}},\ (\Delta p^i_{r2})^\star_{\text{ub}},\ \ldots,\ (\Delta p^i_{rm})^\star_{\text{ub}}]^\top,$$

and $(\Delta p^i_{rs})^\star_{lb}$, $(\Delta p^i_{rs})^\star_{ub}$ denote optimal lower and upper bounds for Δp^i_{rs}, obtained

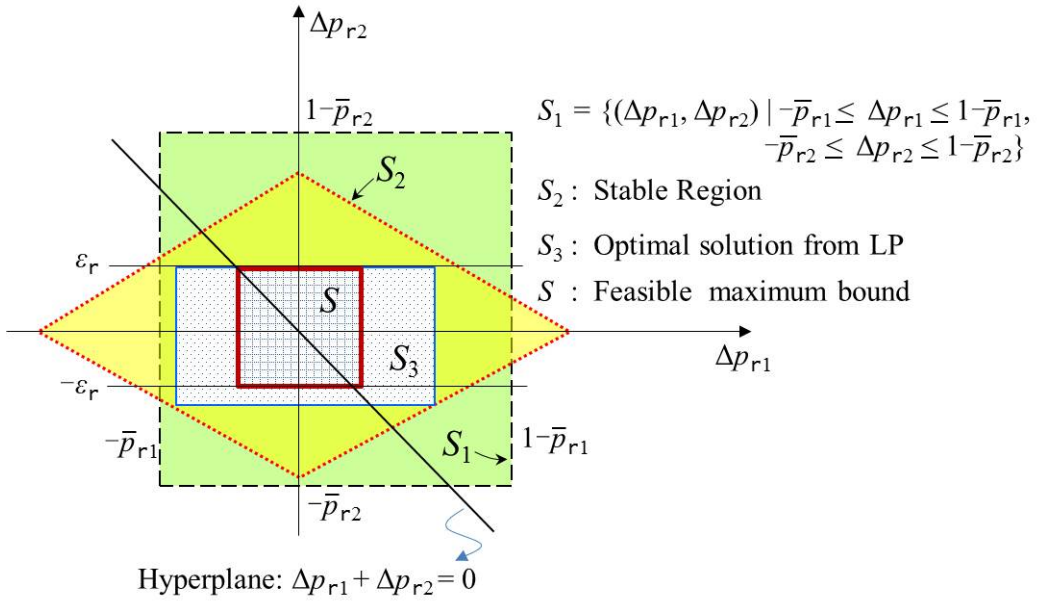


Figure 6.1: The geometry of the Stability Region Analysis for the uncertain Markov transition probability matrix when $m_i = 2$. Each region is described in the figure.

from the LP, respectively.

Since upper or lower bounds are solved by maximizing or minimizing the objective function, $(\Delta p_{rs}^i)^\star$ has different values for upper and lower bounds. Fig.6.1 shows the geometry of stability region analysis for uncertain transition probability matrix. The region S_1 stands for the bounds that come from $-\bar{p}_{rs}^i \le \Delta p_{rs}^i \le 1 - \bar{p}_{rs}^i$. S_2 can be obtained from inequality constraint (6.11). The region S_3 denotes the solution from the LP, and S is the feasible maximum bound with a stability guarantee. Note that ΔP^i satisfies $\sum_{s=1}^{m_i} \Delta p_{rs}^i = 0$, $\forall r$ and hence, feasible solutions should lie on the hyperplane, satisfying $\Delta p_{r1}^i + \Delta p_{r2}^i + \ldots + \Delta p_{rm_i}^i = 0$, $\forall r$. Therefore, we can compute the feasible maximum bound from (6.13) for each row r.

Now we prove that the inequality constraint (6.11) guarantees the system stability. For this purpose, the following Lemma 6.1 will be used.

Lemma 6.1 *Consider a block matrix X defined by*

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mm} \end{bmatrix},$$

where matrix $X_{ij} \in \mathbb{R}^{n\times n}$. Then, we have $\rho(X) < 1$, if $\sum_{j=1}^{m} \|X_{ij}\|_\infty < 1$, $\forall i = 1, 2, \ldots, m$.

Proof For the block matrix X given above, the following inequality condition $\| X \|_\infty \leq \max_i \sum_{j=1}^{m} \| X_{ij} \|_\infty$ holds. Also, it is well known that $\rho(X) \leq \| X \|_p$ for any choice of p.

Therefore, we conclude that

$$\sum_{j=1}^{m} \| X_{ij} \|_\infty < 1, \ \forall i = 1, 2, \ldots, m.$$

$$\Rightarrow \rho(X) \leq \| X \|_\infty < 1.$$

□

Theorem 6.2 *Consider the MJLS* (6.5) *for the large-scale DNCS with communication delays. Then,* (6.5) *is mean square stable if*

$$\sum_{r=1}^{m_i} \alpha_r |\Delta p_{rs}^i| < \beta_s, \quad \begin{matrix} \forall s = 1, 2, \ldots, m_i, \\ \forall i = 1, 2, \ldots, N \end{matrix}$$

where $\alpha_r = \| \hat{W}_r^i \otimes \hat{W}_r^i \|_\infty$ and $\beta_s = 1 - \sum_{r=1}^{m_i} \bar{p}_{rs}^i \| \hat{W}_r^i \otimes \hat{W}_r^i \|_\infty$, is satisfied.

Proof If the Markov transition probability matrix for the system in (6.7) has the uncertainty denoted by $P^i = \bar{P}^i + \Delta P^i$, then the term $\rho\Big((P^{i\top} \otimes I)\ \mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big)$ in (6.8) can be expressed as

$$
\begin{aligned}
&\rho\Big((P^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big) \\
=&\rho\Big(\big((\bar{P}^i + \Delta P^i)^\top \otimes I\big)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big) \\
=&\rho\Big(\Big((\bar{P}^{i\top} \otimes I) + (\Delta P^{i\top} \otimes I)\Big)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big) \\
=&\rho\Big((\bar{P}^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i) + (\Delta P^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big) \\
\leq& \parallel (\bar{P}^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i) + (\Delta P^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i) \parallel_\infty \\
\leq& \parallel (\bar{P}^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i) \parallel_\infty + \parallel (\Delta P^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i) \parallel_\infty, \qquad (6.14)
\end{aligned}
$$

In the first inequality, we used the fact that $\rho(\cdot) \leq \| \cdot \|_\infty$ and the sub-multiplicative property was applied in the last inequality. The block matrix structure for each term of the last inequality is alternatively expressed as follows:

$$
\begin{aligned}
&\Big\|(\bar{P}^{i\top} \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big\|_\infty \\
=&\left\| \underbrace{\begin{bmatrix} \bar{p}_{11}^i I & \bar{p}_{21}^i I & \cdots & \bar{p}_{m_i 1}^i I \\ \bar{p}_{12}^i I & \bar{p}_{22}^i I & \cdots & \bar{p}_{m_i 2}^i I \\ \vdots & \vdots & \cdots & \vdots \\ \bar{p}_{1m_i}^i I & \bar{p}_{2m_i}^i I & \cdots & \bar{p}_{m_i m_i}^i I \end{bmatrix}}_{=(\bar{P}^{i\top} \otimes I)} \underbrace{\begin{bmatrix} \hat{W}_1^i \otimes \hat{W}_1^i & 0 & \cdots & 0 \\ 0 & \hat{W}_2^i \otimes \hat{W}_2^i & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & \hat{W}_{m_i}^i \otimes \hat{W}_{m_i}^i \end{bmatrix}}_{=\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)} \right\|_\infty
\end{aligned}
$$

$$= \left\| \begin{matrix} \gamma_1 \bar{p}^i_{11} & \gamma_2 \bar{p}^i_{21} & \cdots & \gamma_{m_i} \bar{p}^i_{m_i 1} \\ \gamma_1 \bar{p}^i_{12} & \gamma_2 \bar{p}^i_{22} & \cdots & \gamma_{m_i} \bar{p}^i_{m_i 2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1 \bar{p}^i_{1m_i} & \gamma_2 \bar{p}^i_{2m_i} & \cdots & \gamma_{m_i} \bar{p}^i_{m_i m_i} \end{matrix} \right\|_\infty ,$$

where $\gamma_j = (\hat{W}^i_j \otimes \hat{W}^i_j)$, $j = 1, 2, \ldots, m_i$, and similarly,

$$\left\| (\Delta P^{i\top} \otimes I) \mathrm{diag}(\hat{W}^i_j \otimes \hat{W}^i_j) \right\|_\infty = \left\| \begin{matrix} \gamma_1 \Delta p^i_{11} & \gamma_2 \Delta p^i_{21} & \cdots & \gamma_{m_i} \Delta p^i_{m_i 1} \\ \gamma_1 \Delta p^i_{12} & \gamma_2 \Delta p^i_{22} & \cdots & \gamma_{m_i} \Delta p^i_{m_i 2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1 \Delta p^i_{1m_i} & \gamma_2 \Delta p^i_{2m_i} & \cdots & \gamma_{m_i} \Delta p^i_{m_i m_i} \end{matrix} \right\|_\infty .$$

By applying the result in Lemma 6.1 into (6.14), it is guaranteed that $\rho\Big((P^{i\top} \otimes I)\mathrm{diag}(\hat{W}^i_j \otimes \hat{W}^i_j)\Big) < 1$, if the following condition

$$\sum_{r=1}^{m_i} \alpha_r |\Delta p^i_{rs}| + \sum_{r=1}^{m_i} \alpha_r \bar{p}^i_{rs} < 1, \quad \forall s = 1, 2, \ldots, m_i,$$

where $\alpha_r \triangleq ||\hat{W}^i_r \otimes \hat{W}^i_r||_\infty$, is satisfied.

Therefore, (6.5) is mean square stable by Theorem 6.1 if it is guaranteed that

$$\sum_{r=1}^{m_i} \alpha_r |\Delta p^i_{rs}| < \beta_s, \quad \begin{matrix} \forall s = 1, 2, \ldots, m_i, \\ \forall i = 1, 2, \ldots, N, \end{matrix}$$

where $\beta_s \triangleq 1 - \sum_{r=1}^{m_i} \alpha_r \bar{p}^i_{rs}$. □

6.6 Numerical Example

We consider the DNCS that has random communication delays, where the total number of agents is denoted by N. The state for each agent is updated by the following scalar discrete-time dynamics:

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i} a_{ij} x_j(k^*), \quad i = 1, 2, \ldots, N, \tag{6.15}$$

$$\text{where } a_{ij} \triangleq \begin{cases} 0.7, & j = i, \\ -0.14, & j \in \{i-1, i+1\}, \\ 0, & \text{otherwise.} \end{cases}$$

When there is no communication delay (i.e., $k^* = k$), the dynamics for the entire DNCS is obtained by (6.2) as follows:

$$X(k+1) = \mathcal{A}X(k),$$

$$\mathcal{A} = \begin{bmatrix} 0.7 & -0.14 & 0 & 0 & \cdots & 0 \\ -0.14 & 0.7 & -0.14 & 0 & \cdots & 0 \\ 0 & -0.14 & 0.7 & -0.14 & \cdot & 0 \\ \vdots & \cdot & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdot & \cdot & -0.14 & 0.7 & -0.14 \\ 0 & 0 & \cdots & 0 & -0.14 & 0.7 \end{bmatrix} \in \mathbb{R}^{N \times N}.$$

In this case, the stability is computed by $\rho(\mathcal{A}) = 0.9523 < 1$, which assures that the given DNCS without communication delays is stable. However, communications (e.g., wireless network) may be delayed in practice and thus, above stability result is no longer guaranteed. Therefore, it is required to test the system stability in the

existence of random communication delays.

6.6.1 Stability Analysis with Random Communication Delays

We assume that the communication delay τ is bounded by $0 \leq \tau \leq \tau_d = 1$, i.e., $k^* = \{k,\ k-1\}$, $\forall i = 1, 2, \ldots, N$, which implies $q = 2$. Also, we assume that all communication delays for each agent (6.15) are governed by the Markov process with an initial probability distribution $\pi(0)$ and the Markov transition probability matrix P, given by

$$\pi(0) = [1,\ 0], \quad P = \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}. \tag{6.16}$$

For this system, even with $N = 100$, the full state model (6.5) has total $q^{N(N-1)} = 2^{100\times 99}$ modes. It is computationally intractable to deal with $2^{100\times 99}$ numbers of subsystem dynamics, in order to analyze system stability. In contrast, the reduce mode model (6.7) has total $\sum_{i=1}^{N} q^{\hat{n}_i(\hat{n}_i-1)} = 98 \times (2^{3\times 2}) + 2 \times (2^{2\times 1}) = 6280$ modes. Furthermore, the proposed method to reduce the mode numbers fully maximizes its own advantage by considering the symmetric property between agents, which cannot be implemented on the full state model. Since subsystems are symmetric for $\forall i = 2, 3, \ldots, N-1$ and for $\forall i = 1, N$, we only need to check the stability condition for these two cases. Taking into account the symmetric structure, the reduced mode model results in total $2^{3\times 2} + 2^{2\times 1} = 68$ modes. Compared to $2^{100\times 99}$ numbers of modes, the proposed method leads to huge decrease in the mode numbers.

The spectral radius for $i = 2, 3, \ldots, 99$ is computed by $\rho\Big((P^i{}^\top \otimes I)\ \mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big) = 0.8208 < 1$, where $P^i = (P \otimes P \otimes P \otimes P)$. For $i = 1$ and N, we have $\rho\Big((P^i{}^\top \otimes I)\mathrm{diag}(\hat{W}_j^i \otimes \hat{W}_j^i)\Big) = 0.7214 < 1$, where $P^i = (P \otimes P)$. Consequently, this system is stable in the mean square sense by Theorem 6.1. The state trajectory

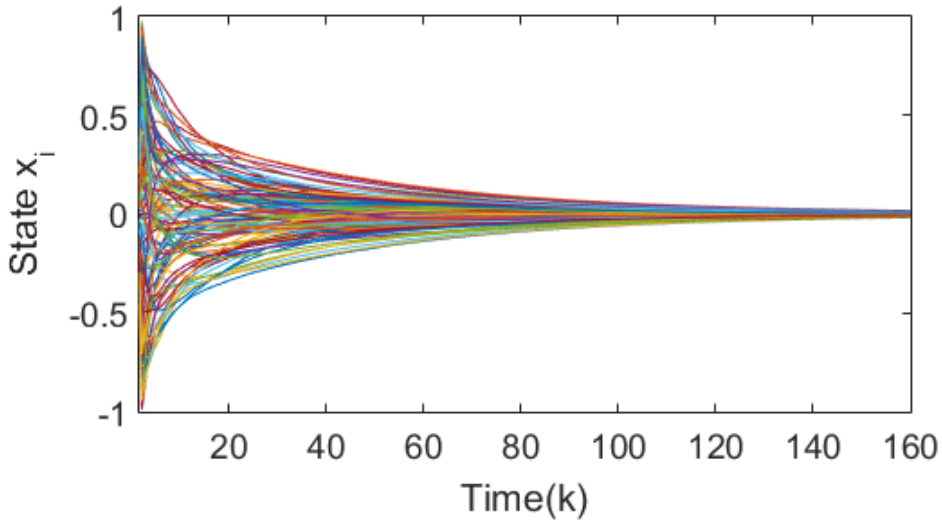


Figure 6.2: State trajectories of each agent for the DNCS with the Markovian communication delays. Initial conditions are randomly generated for the state x_i, $i = 1, 2, \ldots, 100$.

plot also supports this result, as shown in Fig. 6.2. For this simulation, initial condition was assumed to be uniformly distributed in $[-1, 1]$, and was generated by manipulating the MATLAB® command `rand(...)` that generates uniformly distributed pseudo random numbers between 0 and 1.

6.6.2 Stability Bound for Uncertain Markov Transition Probability Matrix

In order to solve the LP to estimate the bound for uncertain Markov transition probability matrix, we used MATLAB® with CVX [42], a Matlab-based software for convex optimization.

6.6.2.1 Scalar system

Although the proposed method to estimate maximum bound for uncertain Markov transition probability matrix is developed for the large-scale DNCS, it is also applicable to general MJLS. We adopted a following example, introduced in [53] to compare

the performance of maximum bound estimation.

Consider the following MJLS that has two modes with scalar discrete-time dynamics.

$$x(k+1) = a_{\sigma(k)}x(k), \quad \sigma(k) \in \{1, 2\},$$
$$a_1 = 1/2, \quad a_2 = 5/4.$$

The Markov transition probability matrix has the form of $P = \bar{P} + \Delta P$, where

$$\bar{P} = \begin{bmatrix} 0.4 & 0.6 \\ 0.5 & 0.5 \end{bmatrix}, \ \Delta P = \begin{bmatrix} \Delta p_{11} & \Delta p_{12} \\ \Delta p_{21} & \Delta p_{22} \end{bmatrix}, \ \sum_{j=1}^{2} \Delta p_{ij} = 0, \quad \forall i = 1, 2.$$

After applying the two steps proposed in this chapter, we obtained the maximum bound $\varepsilon_1 = 0.4$, $\varepsilon_2 = 0.02$ whereas [53] gives the value as $\varepsilon_1 = \varepsilon_2 = 0.021$, which is more conservative. For stability check, among all possible scenarios with $|\Delta p_{rs}| \leq \varepsilon_r$, $\forall r, s = 1, 2$, we have $\max \rho\Big((P^\top \otimes I)\text{diag}(a_j \otimes a_j)\Big) = 1$, which is a marginal value for stability. Hence, the system is stable with obtained maximum bound that is more relaxed than [53].

6.6.2.2 The DNCS with random communication delays

Recalling the DNCS example, let us denote $\bar{P}$ as a nominal Markov transition probability matrix, which represents the transition probability in $k^* \in \{k, k-1\}$. We assume that $\bar{P}$ has the same structure with P in (6.16). Then, the new Markov transition probability matrix associated with the reduced mode model is calculated by $\bar{P}^i = \bar{P} \otimes \bar{P}$ for $i = 1, N$ and $\bar{P}^i = (\bar{P} \otimes \bar{P} \otimes \bar{P} \otimes \bar{P})$ for $i = 2, 3, \ldots, N-1$. More details on synthesizing new Markov transition probability matrix can be found in [97] (see Page 2001).

The feasible solution with the LP provides the maximum bound $\varepsilon^i = [\varepsilon_1^i, \varepsilon_2^i, \ldots \varepsilon_{16}^i]$ $= 10^{-2} \times [0.37, 0.23, 0.23, 0.86, 0.23, 0.86, 0.86, 1.23, 0.23, 0.86, 0.86, 1.23, 0.86, 1.23, 1.23, 1.72]$, $\forall i = 2, 3, \ldots, N-1$. For $i = 1$ and N, we obtained $\varepsilon^i = [\varepsilon_1^i, \varepsilon_2^i, \varepsilon_3^i, \varepsilon_4^i] = 10^{-2} \times [0.75, 2.25, 2.25, 3.75]$. Therefore, we can assure that N inverted pendulum system is mean square stable if the uncertainty in the Markov transition probability matrix is within the bound such that $|\Delta p_{rs}^i| \leq \varepsilon_r^i$, $\forall r$ and $\forall i = 1, 2, \ldots, N$.

6.7 Concluding Remarks

This chapter studied the mean square stability of large-scale DNCSs in connection with random communication delays. Although the Markov jump linear system has been widely adopted to model the systems associated with random communication delays, it inevitably results in a huge number of modes for the large-scale DNCSs, making current stability analysis tools intractable. To avoid this scalability problem, we provided a new analysis framework, which incorporates a reduced mode model that scales linearly with respect to the number of agents. Additionally, we presented a new method to estimate stability bound for uncertain Markov transition probability matrix, in order to guarantee the stability of DNCSs. We showed that this method is less conservative than those proposed in the literature. The validity of the proposed methods were verified by the numerical example.

7. CONCLUSIONS

This dissertation investigated the switched system, particularly for the analysis and synthesis, and applications to large-scale asynchronous switched dynamical systems. For the uncertainty propagation of the state in the switched system, this dissertation developed a novel method, "split-and-merge" algorithm, which enables the performance and robustness analysis in both transient and steady-state time. The proposed method provides new tools for uncertainty quantification of the state, avoiding the computational intractability caused by exponential growth of the component-wise state PDFs. Also, the optimal switching synthesis method was developed for the general switched system having state uncertainty. With an implementation of the receding horizon framework, the proposed methods can be used to design the switching sequence for the optimal performance of the switched system. This synthesis methods can be applied to the optimal controller switching.

As case studies for the switched system, this dissertation introduced massively parallel asynchronous numerical algorithm, implemented on 1D heat equation, large-scale asynchronous distributed quadratic programming problem, and large-scale distributed networked control system with random communication delays. Since the asynchrony is a completely random event and thus results in uncertainty, a proper method is necessarily required to model such an asynchrony. Toward this end, this dissertation showed that the behavior of asynchrony in distributed and parallel computing can be naturally modeled by switched dynamical system. Then the analysis was carried out based on the switched system framework. Also, this switched system framework was applied to the large-scale distributed networked control systems with random communication delays, to develop the stability condition for such systems.

The major contributions of this dissertation can be summarized as follows:

1) For general stochastic switched systems, performance and the robustness analysis tools were developed with the Wasserstein metric. With a novel "Split-and-Merge" algorithm, the scalability issues in uncertainty propagation, due to the exponential increase in the number of modal PDFs, was circumvented. These results addressed both transient and steady-state behavior of stochastic jump linear systems and can be implemented in a computationally efficient manner. The new framework for the performance and the robustness analysis are applicable to general jump linear systems, that may not satisfy Markovian properties.

2) The new method for optimal switching synthesis was proposed with respect to general switched systems in conjunction with Gaussian initial state uncertainty. The Wasserstein metric that defines a distance between PDFs was adopted to measure both the performance and the stability of the switched systems. It was shown that the optimal performance of the system can be obtained by synthesizing switching laws via minimization of objective function while guaranteeing the mean square stability.

3) Through the switched system, the convergence, rate of convergence, and error probability were analyzed for asynchronous parallel numerical algorithms. The asynchronous scheme achieves better performance in terms of the total computation time, particularly when massively parallel computing is required. As a counter effect of implementing asynchronous algorithms, it causes uncertainty in the solution. To depict and analyze this uncertainty, the switched system was introduced. To cope with scalability issues associated with large mode numbers, new methods were developed with the Lyapunov method.

4) The convergence of asynchronous distributed QP problems was analyzed via dual decomposition technique. The behavior of asynchrony in distributed and parallel computing was investigated by the switched system framework. Since the switching mode number becomes large for massively asynchronous computing algorithm, we developed a new algorithm, which drastically decreases mode number. With a new algorithm proposed in this study, the mean square convergence can be tested without any computational intractability. In addition, the analytic expression for the rate of convergence was derived to compare the speed of convergence for three different schemes (e.g., synchronous, deterministic asynchronous, and stochastic asynchronous algorithms).

5) The computationally efficient test condition was devised for the mean square stability of large-scale DNCSs having random communication delays. Although, this system does not have direct connection with asynchronous dynamical systems, the characteristics of such random communication delays, effectively represented by Markov jump linear systems, can be regarded as asynchronous communication. To avoid the scalability problem that is encountered by large mode numbers, the reduced mode model was introduced, which scales linearly with respect to the number of agents. Additionally, a new method was developed to estimate stability bound for uncertain Markov transition probability matrix, in order to guarantee the stability of DNCSs. The proposed method was proved to be less conservative than those proposed in the literature.

REFERENCES

[1] Pedro Aguiar, Eric P Xing, Mário Figueiredo, Noah A Smith, and André Martins. An augmented lagrangian approach to constrained map inference. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 169–176, 2011.

[2] Simo Ali-Löytty. *On the convergence of the Gaussian mixture filter*. 2008.

[3] Stefan Arnborg, Derek G Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in ak-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[4] Jacques F Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[5] Sorin C Bengea and Raymond A DeCarlo. Optimal control of switching systems. *Automatica*, 41(1):11–27, 2005.

[6] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[7] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[8] Fayer F Boctor. Some efficient multi-heuristic procedures for resource-constrained project scheduling. *European Journal of Operational Research*, 49(1):3–13, 1990.

[9] Paolo Bolzern, Patrizio Colaneri, and Giuseppe De Nicolao. Markov jump linear systems with switching transition rates: mean square stability with dwell-time. *Automatica*, 46(6):1081–1088, 2010.

[10] El-Kébir Boukas. *Stochastic switching systems: Analysis and design*. Springer, 2006.

[11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[12] Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[13] Peter Buchholz, Markus Fischer, and Peter Kemper. Distributed steady state analysis using kronecker algebra. *Numerical Solutions of Markov Chains (NSMC99), Prensas Universitarias de Zaragoza, Zaragoza, Spain*, pages 76–95, 1999.

[14] Christos G Cassandras, David L Pepyne, and Yorai Wardi. Optimal control of a class of hybrid systems. *Automatic Control, IEEE Transactions on*, 46(3):398–415, 2001.

[15] Yair Censor and Stavros Andrea Zenios. Proximal minimization algorithm withd-functions. *Journal of Optimization Theory and Applications*, 73(3):451–464, 1992.

[16] Yuan-Chieh Cheng and Thomas G Robertazzi. Distributed computation with communication delay (distributed intelligent sensor networks). *Aerospace and Electronic Systems, IEEE Transactions on*, 24(6):700–712, 1988.

[17] Oswaldo LV Costa and Marcelo D Fragoso. Stability results for discrete-time linear systems with markovian jumping parameters. *Journal of Mathematical Analysis and Applications*, 179(1):154–178, 1993.

[18] Lorenzo Coviello, Paolo Minero, and Massimo Franceschetti. Stabilization over markov feedback channels. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 3776–3782. IEEE, 2011.

[19] Jamal Daafouz, Pierre Riedinger, and Claude Iung. Stability analysis and control synthesis for switched systems: a switched lyapunov function approach. *Automatic Control, IEEE Transactions on*, 47(11):1883–1887, 2002.

[20] Robert J Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965.

[21] George B Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

[22] Tuhin Das and Ranjan Mukherjee. Optimally switched linear systems. *Automatica*, 44(5):1437–1441, 2008.

[23] Carlos E de Souza. Robust stability and stabilization of uncertain discrete-time markovian jump linear systems. *Automatic Control, IEEE Transactions on*, 51(5):836–841, 2006.

[24] Oswaldo Luiz do Valle Costa, Marcelo Dutra Fragoso, and Ricardo Paulino Marques. *Discrete-time Markov Jump Linear Systems*. Springer, 2006.

[25] Diego A Donzis and Konduri Aditya. Asynchronous finite-difference schemes for partial differential equations. *Journal of Computational Physics*, 274:370–392, 2014.

[26] Jim Douglas and Henry H Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, pages 421–439, 1956.

[27] Jonathan Eckstein. Nonlinear proximal point algorithms using bregman functions, with applications to convex programming. *Mathematics of Operations Research*, 18(1):202–226, 1993.

[28] M Egerstedt, Y Wardi, and F Delmotte. Optimal control of switching times in switched dynamical systems. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, pages 2138–2143. IEEE, 2003.

[29] Hugh Everett III. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

[30] Zhe Fan, Feng Qiu, Arie Kaufman, and Suzanne Yoakum-Stover. Gpu cluster for high performance computing. In *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*, page 47. IEEE Computer Society, 2004.

[31] Lei Fang, Hai Lin, and Panos J Antsaklis. Stabilization and performance analysis for a class of switched systems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 3, pages 3265–3270. IEEE, 2004.

[32] Yuguang Fang and Kenneth A Loparo. Stabilization of continuous-time jump linear systems. *Automatic Control, IEEE Transactions on*, 47(10):1590–1603, 2002.

[33] Yuguang Fang and Kenneth A Loparo. Stochastic stability of jump linear systems. *Automatic Control, IEEE Transactions on*, 47(7):1204–1208, 2002.

[34] Xiangbo Feng, Kenneth A Loparo, Yuandong Ji, and Howard Jay Chizeck. Stochastic stability properties of jump linear systems. *Automatic Control,*

IEEE Transactions on, 37(1):38–53, 1992.

[35] Michel Fortin and Roland Glowinski. *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*. Elsevier, 2000.

[36] Masao Fukushima. Application of the alternating direction method of multipliers to separable convex programming problems. *Computational Optimization and Applications*, 1(1):93–111, 1992.

[37] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[38] Arthur M Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[39] Lewis Girod and Deborah Estrin. Robust range estimation using acoustic and multimodal sensing. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1312–1320. IEEE, 2001.

[40] Clark R Givens and Rae Michael Shortt. A class of wasserstein metrics for probability distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.

[41] Roland Glowinski and A Marroco. Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualite d'une classe de problemes de dirichlet non lineaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.

[42] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[43] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*, volume 2. Oxford University Press, 1992.

[44] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford University Press, 2001.

[45] Leonid Gurvits, Robert Shorten, and Oliver Mason. On the stability of switched positive linear systems. *Automatic Control, IEEE Transactions on*, 52(6):1099–1103, 2007.

[46] Sadri Hassani. *Mathematical Physics: A Modern Introduction to Its Foundations*. Springer Science & Business Media, 1999.

[47] Arash Hassibi, Stephen P Boyd, and Jonathan P How. Control of asynchronous dynamical systems with rate constraints on events. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 2, pages 1345–1351. IEEE, 1999.

[48] Sven Hedlund and Anders Rantzer. Optimal control of hybrid systems. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 4, pages 3972–3977. IEEE, 1999.

[49] Joao P Hespanha and A Stephen Morse. Stability of switched systems with average dwell-time. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 3, pages 2655–2660. IEEE, 1999.

[50] Magnus R Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[51] Zhenting Hou, Jiaowan Luo, Peng Shi, and Sing Kiong Nguang. Stochastic stability of ito differential equations with semi-markovian jump parameters. *IEEE Transactions on Automatic Control*, 51(8):1383, 2006.

[52] Y Ji. Stability and control of discrete-time jump linear systems. *Control-Theory and Advanced Technology*, 7(2):247–270, 1991.

[53] Mehmet Karan, Peng Shi, and C Yalçın Kaya. Transition probability bounds for the stochastic stability robustness of continuous-and discrete-time markovian jump linear systems. *Automatica*, 42(12):2159–2168, 2006.

[54] Nicholas Kottenstette and Joseph Porter. Digital passive attitude and altitude control schemes for quadrotor aircraft. In *Control and Automation, 2009. ICCA 2009. IEEE International Conference on*, pages 1761–1768. IEEE, 2009.

[55] Frank Kozin. A survey of stability of stochastic systems. *Automatica*, 5(1):95–112, 1969.

[56] Thomas J Laffey and Helena Šmigoc. Tensor conditions for the existence of a common solution to the lyapunov equation. *Linear Algebra and Its Applications*, 420(2):672–685, 2007.

[57] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.

[58] Ji-Woong Lee and Geir E Dullerud. Uniform stabilization of discrete-time switched and markovian jump linear systems. *Automatica*, 42(2):205–218, 2006.

[59] Kooktae Lee and Raktim Bhattacharya. Optimal switching synthesis for jump linear systems with gaussian initial state uncertainty. In *ASME 2014 Dynamic*

Systems and Control Conference, pages V002T24A003–V002T24A003. American Society of Mechanical Engineers, 2014.

[60] Kooktae Lee and Raktim Bhattacharya. On the convergence analysis of asynchronous distributed quadratic programming via dual decomposition. *arXiv preprint arXiv:1506.05485*, 2015.

[61] Kooktae Lee and Raktim Bhattacharya. Stability analysis of large-scale distributed networked control systems with random communication delays: A switched system approach. *arXiv preprint arXiv:1503.03047*, 2015.

[62] Kooktae Lee, Raktim Bhattacharya, and Vijay Gupta. A switcehd dynamical system framework for analysis of massively parallel asynchronous numerical algorithms. In *American Control Conference (ACC)*, page to appear. IEEE, 2015.

[63] Kooktae Lee, Abhishek Halder, and Raktim Bhattacharya. Probabilistic robustness analysis of stochastic jump linear systems. In *American Control Conference (ACC), 2014*, pages 2638–2643. IEEE, 2014.

[64] Kooktae Lee, Abhishek Halder, and Raktim Bhattacharya. Performance and robustness analysis of stochastic jump linear systems using wasserstein metric. *Automatica*, 51:341–347, 2015.

[65] Fanbiao Li, Ligang Wu, and Peng Shi. Stochastic stability of semi-markovian jump systems with mode-dependent delays. *International Journal of Robust and Nonlinear Control*, 2013.

[66] Daniel Liberzon. *Switching in systems and control*. Springer, 2003.

[67] Daniel Liberzon and A Stephen Morse. Basic problems in stability and design of switched systems. *Control Systems, IEEE*, 19(5):59–70, 1999.

[68] Hai Lin and Panos J Antsaklis. Stability and stabilizability of switched linear systems: a survey of recent results. *Automatic control, IEEE Transactions on*, 54(2):308–322, 2009.

[69] Hai Lin, Guisheng Zhai, Lei Fang, and Panos J Antsaklis. Stability and h performance preserving scheduling policy for networked control systems. In *Proc. 16th IFAC World Congress on Automatic Control*, 2005.

[70] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[71] Ji Liu, Stephen J Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *arXiv preprint arXiv:1311.1873*, 2013.

[72] Juan Liu, Maurice Chu, and James E Reich. Multitarget tracking in distributed sensor networks. *Signal Processing Magazine, IEEE*, 24(3):36–46, 2007.

[73] Ming Liu, Daniel WC Ho, and Yugang Niu. Stabilization of markovian jump linear system over networks with random communication delay. *Automatica*, 45(2):416–421, 2009.

[74] Michel Mariton. *Jump linear systems in automatic control*. CRC Press, 1990.

[75] KD Minto and R Ravi. New results on the multi-controller scheme for the reliable control of linear plants. In *American Control Conference, 1991*, pages 615–619. IEEE, 1991.

[76] Kumpati S Narendra and Jeyendran Balakrishnan. Improving transient response of adaptive control systems using multiple models and switching. *Automatic Control, IEEE Transactions on*, 39(9):1861–1866, 1994.

[77] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, 2008.

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

[79] John D Owens, Mike Houston, David Luebke, Simon Green, John E Stone, and James C Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899, 2008.

[80] Richard H Pletcher, John C Tannehill, and Dale Anderson. *Computational fluid mechanics and heat transfer*. CRC Press, 2012.

[81] Michael JD Powell. *A method for non-linear constraints in minimization problems*. UKAEA, 1967.

[82] James A Primbs, Vesna Nevistic, and John C Doyle. A receding horizon generalization of pointwise min-norm controllers. *Automatic Control, IEEE Transactions on*, 45(5):898–909, 2000.

[83] David W Scott. Multivariate density estimation. 1992.

[84] Pete Seiler and Raja Sengupta. Analysis of communication losses in vehicle control problems. In *American Control Conference, 2001. Proceedings of the 2001*, volume 2, pages 1491–1496. IEEE, 2001.

[85] Peng Shi, E-K Boukas, and Ramesh K Agarwal. Control of markovian jump discrete-time systems with norm bounded uncertainty and unknown delay. *Automatic Control, IEEE Transactions on*, 44(11):2139–2144, 1999.

[86] Yang Shi and Bo Yu. Output feedback stabilization of networked control systems with random delays modeled by markov chains. *Automatic Control, IEEE Transactions on*, 54(7):1668–1674, 2009.

[87] Robert Shorten, Kumpati S Narendra, and Oliver Mason. A result on common quadratic lyapunov functions. *Automatic Control, IEEE Transactions on*, 48(1):110–113, 2003.

[88] Jonathan E Spingarn. Applications of the method of partial inverses to convex programming: decomposition. *Mathematical Programming*, 32(2):199–223, 1985.

[89] Zhendong Sun and Shuzhi Sam Ge. Analysis and synthesis of switched linear control systems. *Automatica*, 41(2):181–195, 2005.

[90] David Sworder. Feedback control of a class of linear systems with jump parameters. *Automatic Control, IEEE Transactions on*, 14(1):9–14, 1969.

[91] D Michael Titterington, Adrian FM Smith, Udi E Makov, et al. *Statistical analysis of finite mixture distributions*, volume 7. Wiley New York, 1985.

[92] Paul Tseng. Alternating projection-proximal methods for convex programming and variational inequalities. *SIAM Journal on Optimization*, 7(4):951–965, 1997.

[93] Cédric Villani. *Topics in optimal transportation*, volume 58. AMS Bookstore, 2003.

[94] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[95] Gregory C Walsh, Hong Ye, and Linda G Bushnell. Stability analysis of networked control systems. *Control Systems Technology, IEEE Transactions on*, 10(3):438–446, 2002.

[96] Jing Wu and Tongwen Chen. Design of networked control systems with packet dropouts. *Automatic Control, IEEE Transactions on*, 52(7):1314–1319, 2007.

[97] Lin Xiao, Arash Hassibi, and Jonathan P How. Control with random communication delays via a discrete-time jump system approach. In *American Control Conference, 2000. Proceedings of the 2000*, volume 3, pages 2199–2204. IEEE, 2000.

[98] Junlin Xiong and James Lam. Stabilization of discrete-time markovian jump linear systems via time-delayed controllers. *Automatica*, 42(5):747–753, 2006.

[99] Junlin Xiong and James Lam. Stabilization of linear systems over networks with bounded packet loss. *Automatica*, 43(1):80–87, 2007.

[100] Xuping Xu and Panos J Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *Automatic Control, IEEE Transactions on*, 49(1):2–16, 2004.

[101] Fuwen Yang, Zidong Wang, YS Hung, and Mahbub Gani. H control for networked systems with random communication delays. *Automatic Control, IEEE Transactions on*, 51(3):511–518, 2006.

[102] John K Yook, Dawn M Tilbury, and Nandit R Soparkar. Trading computation for bandwidth: Reducing communication in distributed control systems using state estimators. *Control Systems Technology, IEEE Transactions on*, 10(4):503–518, 2002.

[103] Keyou You and Lihua Xie. Minimum data rate for mean square stabilizability of linear systems with markovian packet losses. *Automatic Control, IEEE Transactions on*, 56(4):772–785, 2011.

[104] Liqian Zhang, Yang Shi, Tongwen Chen, and Biao Huang. A new method for stabilization of networked control systems with random delays. *Automatic Control, IEEE Transactions on*, 50(8):1177–1181, 2005.

[105] Lixian Zhang, E-K Boukas, and James Lam. Analysis and synthesis of markov jump linear systems with time-varying delays and partially known transition probabilities. *Automatic Control, IEEE Transactions on*, 53(10):2458–2464, 2008.

[106] Lixian Zhang and El-Kébir Boukas. Stability and stabilization of markovian jump linear systems with partly unknown transition probabilities. *Automatica*, 45(2):463–468, 2009.

[107] Lixian Zhang and James Lam. Necessary and sufficient conditions for analysis and synthesis of markov jump linear systems with incomplete transition descriptions. *Automatic Control, IEEE Transactions on*, 55(7):1695–1701, 2010.

[108] Xudong Zhao, Shen Yin, Hongyi Li, and Ben Niu. Switching stabilization for a class of slowly switched systems. *Automatic Control, IEEE Transactions on*, 60(1):221–226, 2014.

APPENDIX A

PROOF FOR CHAPTER 2

A.1 Proof of Proposition 2.4

Proof From (3.7) and (2.6), we have

$$\begin{aligned} W^2 &= \int_{\mathbb{R}^n} \| x \|^2_{\ell_2(\mathbb{R}^n)} \varsigma(x)dx = \int_{\mathbb{R}^n} \| x \|^2_{\ell_2(\mathbb{R}^n)} \sum_{j=1}^m \pi_j \varsigma_j(x)dx \\ &= \sum_{j=1}^m \pi_j \int_{\mathbb{R}^n} \| x \|^2_{\ell_2(\mathbb{R}^n)} \varsigma_j(x)dx = \sum_{j=1}^m \pi_j W_j^2, \end{aligned} \tag{A.1}$$

where $W_j \triangleq W\left(\varsigma_j(x), \delta(x)\right)$.

Also, we can compute $\hat{W}^2 \triangleq W^2(\mathcal{N}(\hat{\mu}, \hat{\Sigma}), \delta(x))$ from Proposition 2.2 as follows.

$$\begin{aligned} \hat{W}^2 =& \| \hat{\mu} \|^2_{\ell_2(\mathbb{R}^n)} + \mathrm{tr}(\hat{\Sigma}) \\ \overset{(2.15)}{=}& \hat{\mu}^\top \hat{\mu} + \mathrm{tr}\left(\sum_{j=1}^m \pi_j \Big(\Sigma_j + (\mu_j - \hat{\mu})(\mu_j - \hat{\mu})^\top \Big) \right). \end{aligned} \tag{A.2}$$

Since $\mathrm{tr}(\cdot)$ is linear operator and $\sum_{j=1}^m \pi_j = 1$, (A.2) becomes

$$\begin{aligned} \hat{W}^2 = \hat{\mu}^\top \hat{\mu} + \sum_{j=1}^m \pi_j \mathrm{tr}\left(\Sigma_j\right) + \mathrm{tr}\left(\sum_{j=1}^m \pi_j \mu_j \mu_j^\top \right) - \\ \mathrm{tr}\left(\left(\sum_{j=1}^m \pi_j \mu_j \right) \hat{\mu}^\top \right) - \mathrm{tr}\left(\hat{\mu} \left(\sum_{j=1}^m \pi_j \mu_j \right)^\top \right) + \mathrm{tr}\left(\hat{\mu}\hat{\mu}^\top \right). \end{aligned} \tag{A.3}$$

Now, we recall from (2.15) that $\hat{\mu} = \sum_{j=1}^m \pi_j \mu_j$, and that $\hat{\mu}^\top \hat{\mu} = \mathrm{tr}\left(\hat{\mu}^\top \hat{\mu}\right) =$

$\text{tr}\left(\hat{\mu}\hat{\mu}^\top\right)$. Consequently, the first, fourth, fifth and sixth terms in (A.3) cancel out, resulting in

$$\begin{aligned}\hat{W}^2 &= \sum_{j=1}^{m} \pi_j \text{tr}\left(\Sigma_j\right) + \sum_{j=1}^{m} \pi_j \,\text{tr}\left(\mu_j \mu_j^\top\right) \\ &= \sum_{j=1}^{m} \pi_j \left(\parallel \mu_j \parallel^2_{\ell_2(\mathbb{R}^n)} + \text{tr}\left(\Sigma_j\right)\right) = \sum_{j=1}^{m} \pi_j W_j^2. \end{aligned} \tag{A.4}$$

From (A.1) and (A.4) $W^2 = \hat{W}^2$ for any time index k. Therefore,

$$\begin{aligned} W^2(k) = \hat{W}^2(k) &= \sum_{j=1}^{m} \pi_j(k) \left(\parallel \mu_j(k) \parallel^2_{\ell_2(\mathbb{R}^n)} + \text{tr}\big(\left(\Sigma_j(k)\right)\big)\right) \\ &= \text{tr}\left(\sum_{j=1}^{m} \pi_j(k) \Big(\mu_j(k) \mu_j(k)^\top + \Sigma_j(k) \Big) \right). \end{aligned} \tag{A.5}$$

Here, $\mu_j(k)$ and $\Sigma_j(k)$ are corresponding modal mean and covariance of the Gaussian mixture at time k, obtained from the synthetic Gaussian $\mathcal{N}(\hat{\mu}(k-1), \hat{\Sigma}(k-1))$ at time $k-1$, i.e.,

$$\mu_j(k) = A_j \hat{\mu}(k-1), \quad \Sigma_j(k) = A_j \hat{\Sigma}(k-1) A_j^\top. \tag{A.6}$$

Replacing $\mu_j(k)$ and $\Sigma_j(k)$ in (A.5) with (A.6) results in

$$\begin{aligned} W^2(k) &= \text{tr}\left(\sum_{j=1}^{m} \pi_j(k) A_j \underbrace{\Big(\hat{\mu}(k-1)\hat{\mu}(k-1)^\top + \hat{\Sigma}(k-1) \Big)}_{\triangleq \Phi(k-1)} A_j^\top \right) \\ &= \text{tr}\left(\left(\sum_{j=1}^{m} \pi_j(k) A_j^\top A_j \right)^\top \Phi(k-1) \right). \end{aligned} \tag{A.7}$$

Since the trace is invariant under cyclic permutation, the property $\text{tr}(ABC) =$

$\mathrm{tr}(CAB)$ was applied between first and second line of above equations. Moreover, using the trace property $\mathrm{tr}(X^\top Y) = \mathrm{vec}(X)^\top \mathrm{vec}(Y)$, (A.7) can be expressed as

$$W^2(k) = \mathrm{vec}\left(\sum_{j=1}^{m} \pi_j(k) A_j^\top I_n A_j\right)^\top \mathrm{vec}\Big(\Phi(k-1)\Big), \tag{A.8}$$

where I_n is $n \times n$ identity matrix.

Applying $\mathrm{vec}(ABC) = \left(C^\top \otimes A\right)\mathrm{vec}(B)$ to the first term of (A.8), we obtain

$$\begin{aligned} W^2(k) &= \left(\sum_{j=1}^{m} \pi_j(k) \left(A_j^\top \otimes A_j^\top\right) \mathrm{vec}\left(I_n\right)\right)^\top \mathrm{vec}\Big(\Phi(k-1)\Big) \\ &= \mathrm{vec}\left(I_n\right)^\top \left(\sum_{j=1}^{m} \pi_j(k) \left(A_j \otimes A_j\right)\right) \mathrm{vec}\Big(\Phi(k-1)\Big). \end{aligned} \tag{A.9}$$

Recalling (A.2), we have $W^2 = \hat{W}^2 = \| \hat{\mu} \|^2_{\ell_2(\mathbb{R}^n)} + \mathrm{tr}(\hat{\Sigma}) = \mathrm{tr}\left(\hat{\mu}\hat{\mu}^\top + \hat{\Sigma}\right) = \mathrm{tr}\left(I_n^\top \left(\hat{\mu}\hat{\mu}^\top + \hat{\Sigma}\right)\right)$. Again, from the trace property $\mathrm{tr}(X^\top Y) = \mathrm{vec}(X)^\top \mathrm{vec}(Y)$, above equation with time index k further becomes

$$W^2(k) = \hat{W}^2(k) = \mathrm{vec}(I_n)^\top \mathrm{vec}\Big(\Phi(k)\Big), \tag{A.10}$$

where $\Phi(k) \triangleq \hat{\mu}(k)\hat{\mu}(k)^\top + \hat{\Sigma}(k)$. Similarly, W^2 at $k-1$ becomes,

$$W^2(k-1) = \hat{W}^2(k-1) = \mathrm{vec}(I_n)^\top \mathrm{vec}\Big(\Phi(k-1)\Big). \tag{A.11}$$

Finally, from the recurrence relation between (A.9) and (A.11), we conclude that

$$\begin{aligned} W^2(k) = \hat{W}^2(k) &= \mathrm{vec}(I_n)^\top \Gamma(k) \mathrm{vec}\Big(\Phi(0)\Big) \\ &= \mathrm{vec}(I_n)^\top \Gamma(k) \mathrm{vec}\left(\hat{\mu}(0)\hat{\mu}(0)^\top + \hat{\Sigma}(0)\right), \end{aligned}$$

where $\Gamma(k) \triangleq \prod_{i=k}^{1} \left(\sum_{j=1}^{m} \pi_j(i)(A_j \otimes A_j) \right)$. $\square$