2011

Communication-constrained feedback stability and Multi-agent System consensusability in Networked Control Systems

Laurentiu Dan Marinovici
Louisiana State University and Agricultural and Mechanical College

COMMUNICATION-CONSTRAINED FEEDBACK STABILITY AND MULTI-AGENT SYSTEM CONSENSUSABILITY IN NETWORKED CONTROL SYSTEMS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Electrical and Computer Engineering

by
Laurentiu Dan Marinovici
Department of Electrical and Computer Engineering
B.Sc., "Gh. Asachi" Technical University, Iasi, Romania, 2000
M.Sc., Louisiana State University, 2007
August, 2011

Acknowledgements

I would like to express my never ending gratitude to my advisor and committee chair, Professor Kemin Zhou, for his ceaseless guidance and patience, challenging ideas, and his willingness to share his bottomless pool of knowledge. It was him who introduced me to the field of robust control, and constantly elaborated on the subject throughout the years of my studies.

Also, I am undoubtedly indebted to my co-advisor and committee co-chair, Professor Guoxiang Gu, for his topics suggestions, relevant technical discussions, and stimulating advice.

Without their help and guidance, this dissertation would have been next to impossible to write.

I extend my thanks to the late Professor Jorge Aravena, whose endless positive attitude and contagious enthusiasm taught me how to get more and more committed to my goals and dreams.

I would also like to thank all the professors in the Louisiana State University Departments of Electrical and Computer Engineering and Mathematics, whose classes I had the privilege of attending while completing my studies. They all helped me with broadening my knowledge horizon and made writing this thesis easier. I am grateful to the Electrical and Computer Engineering Department for the professional and financial support.

I give my regards to all my friends and colleagues who encouraged me not to give up when things seemed rough, and always offered supporting words and moral leverage.

Last, but not least, I give a very warm appreciative embrace to all my family members, who, though far away, understood, supported, and encouraged me to move forward. And also, I offer many thanks to my fiancée, Cristina, for her love and for patiently standing by me all these years.

Table of Contents

List of Tables

List of Figures

Notation and Symbols

$\mathbb{R}$	set of real numbers
$\mathbb{C}$	set of complex numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{Z}_+$	set of non-negative integers
$\in$	belong to
$\square$	end of proof
$:=$	defined as
$\triangleq$	denoted as
$\lvert\alpha\rvert$	absolute value of $\alpha \in \mathbb{R}$ or $\alpha \in \mathbb{C}$
I_n	$n \times n$ identity matrix
$[a_{i,j}]$	a matrix with $a_{i,j}$ as the element on the i-th row and j-th column
$\mathrm{diag}(a_1, \ldots, a_n)$	an $n \times n$ diagonal matrix with a_i as the i-th element on the diagonal
A^{T}	transpose of matrix A
A^*	transpose conjugate of matrix A
A^{-1}	inverse of matrix A
$\det(A)$	determinant of matrix A
$\lambda_i(A)$	i-th eigenvalue of matrix A
$\sigma_i(A)$	i-th singular value of matrix A
$\bar{\sigma}(A)$	largest singular value of matrix A
$\otimes$	Kronecker product operator
$\angle$	angle
$G(s)$	transfer function (continuous time)
$G(z)$	transfer function (discrete time)
$\left[\begin{array}{c\|c} \mathrm{A} & \mathrm{B} \\ \hline \mathrm{C} & \mathrm{D} \end{array}\right]$	shorthand for state space realization $C(sI - A)^{-1}B + D$ (continuous time) or $C(zI - A)^{-1}B + D$ (discrete time)
$\mathcal{F}_l(M, Q)$	lower LFT
$\mathcal{F}_u(M, Q)$	upper LFT

Acronyms

DARE discrete algebraic Riccati equation

FI Full Information

LDTI Linear Discrete Time Invariant

LFT Linear Fractional Transformation

LQR Linear Quadratic Regulator

LTI Linear Time Invariant

MAS Multi-agent System

MIMO Multiple-Input Multiple-Output

NCS Networked Control System

SISO Single-Input Single-Output

UAV unmanned air vehicle

UGV unmanned ground vehicle

UUV unmanned underwater vehicle

Abstract

With the advances in wireless communication, the topic of Networked Control Systems (NCSs) has become an interesting research subject. Moreover, the advantages they offer convinced companies to implement and use data networks for remote industrial control and process automation. Data networks prove to be very efficient for controlling distributed systems, which would otherwise require complex wiring connections on large or inaccessible areas. In addition, they are easier to maintain and more cost efficient. Unfortunately, stability and performance control is always going to be affected by network and communication issues, such as band-limited channels, quantization errors, sampling, delays, packet dropouts or system architecture.

The first part of his research aims to study the effects of both input and output quantization on an NCS. Both input and output quantization errors are going to be modeled as sector bounded multiplicative uncertainties, the main goal being the minimization of the quantization density, while maintaining feedback stability. Modeling quantization errors as uncertainties allows for robust optimal control strategies to be applied in order to study the accepted uncertainty levels, which are directly related to the quantization levels. A new feedback law is proposed that will improve closed-loop system stability by increasing the upper bound of allowed uncertainty, and thus allowing the use of a coarser quantizer.

Another aspect of NCS deals with coordination of the independent agents within a Multi-agent System (MAS). This research addresses the consensus problem for a set of discrete-time agents communicating through a network with directed information flow. It examines the combined effect of agent dynamics and network topology on agents' consensusability. Given a particular consensus protocol, a sufficient condition is given for agents to be consensusable. This condition requires the eigenvalues of the digraph modeling the network topology to be outer bounded by a fan-shaped area determined by the Mahler measure of the agents' dynamics matrix.

Chapter 1
Introduction

1.1 Motivations and Challenges of NCS and MAS

Due to an accelerating technological merging of communications, control, and computing, researchers in various fields have been interested in the rapid and constantly exciting developments and the technological challenges of the Networked Control Systems (NCSs). Typically, these systems consist of the system to be controlled and of actuators, sensors, and controllers which are operated through a communication network. In most of the cases, the NCSs are spatially distributed, might function in an asynchronous mode, while still being coordinated such that they achieve some desired performances.

Spatially distributed components for control systems are not considered novelty since they have been around for decades in chemical process plants, refineries, power plants, and airplanes. Years ago, such industrial equipment had its components hard-wired, and the information from all sensors was brought to a dispatcher, which would monitor conditions, analyze data, and take decisions on how the system was to be controlled. Actuators like valves, motors, etc. were then used to implement the control policies. Nowadays, computer routines run on remotely located microprocessors, controlling the order in which input and output devices obtain access to the processed data, and sending the data via shared digital networks or wireless connections. These changes were imminent, since wiring hardware costs went up while IT devices got cheaper, and also since physically introducing new components into some systems, as the needs change, proved to be difficult. All these reasons lead to a continuously growing interest in networked control systems, which, in turn, raised up essentially new questions in communications, information processing, and control, dealing with the relationship between network operation and the quality of the overall system's behavior. For instance, researchers are studying the connection between closed-loop stability and communication constraints on the feedback channels. Attention

has been focused on the minimum transmission bit rate that would guarantee stability of the feedback loop. The nature of the wireless communication links between sensors, actuators, and controllers, and their limitations (packet rates, sampling, network delay, packet dropouts) also brought questions and challenges in the estimation and controller synthesis area.

It is important to notice that networked control systems research is mainly at the crossroad of three research areas: control systems, communication networks and information theory, and computer science. While many theoretical results have been obtained separately in each of these fields focusing on individual problems, merging them would be of great benefit to the NCSs research community. Usually, control theory considers the links between the interconnected dynamical systems as "ideal channels", in order to focus on issues related to control area. In contrast, communication theory ignores most of the control problems, so that it can focus on the transmission of information over "imperfect channels". Since, as shown in Figure 1.1, the NCSs are, in general, spatially distributed systems in which sensors, actuators, and controllers communicate through a shared band-limited network, combining these frameworks proved to be imperative in NCSs modeling.

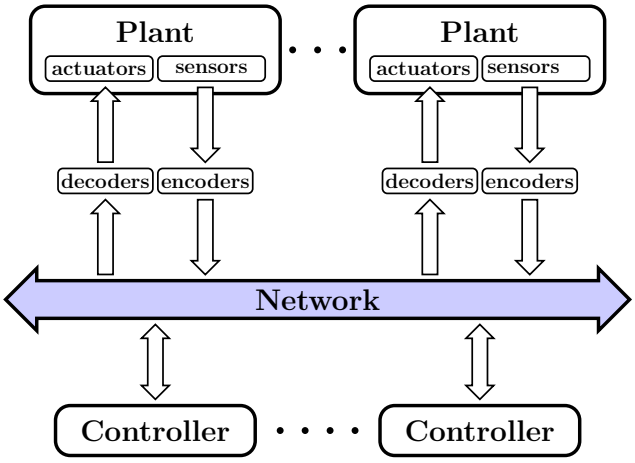


Figure 1.1: General NCS architecture.

Several key issues that make networked control systems distinct from the other control systems are detailed in the following paragraphs.

Band-Limited Channels. A set of significant constraints set upon NCSs are due to the incapacity of communication networks to carry an infinite amount of information per unit of time. Starting with Shannon's results on the *maximum* amount of information that can be reliably transmitted on a communication channel with a specified bandwidth, researchers in the communication field developed a series of important results. Some of these results have been used by control theory to solve the problem of *minimum bit rate* necessary to stabilize a linear system through feedback over a finite capacity channel [15], [32], [42].

Sampling and Delay. Though periodic sampling of a continuous signal has been extensively studied in the digital control area, the process becomes significantly more complicated once the signal needs to be further encoded, transmitted over a network and decoded by a receiver. The overall network delay, which mainly consists of network access delays and transmission delays, plays a key role in the system. It can be highly variable due to network conditions, such as congestion and channel capacity. Even though communication researchers do not see them as potential pitfalls, network delays are given serious thoughts in the control community, since they can affect both performance and stability of the system. Open-loop control, such as on-off relay systems, for instance, are not significantly influenced by network delays. Nevertheless, real life time sensitive high performance applications, like telerobotics and telesurgery, cannot be adequately controlled using an open-loop structure. They require measurement data to be sent across the network in order to minimize the output error. Either constant or time varying, the network delays will degrade performance and stability. Therefore, advanced methodologies are necessary to handle their effects.

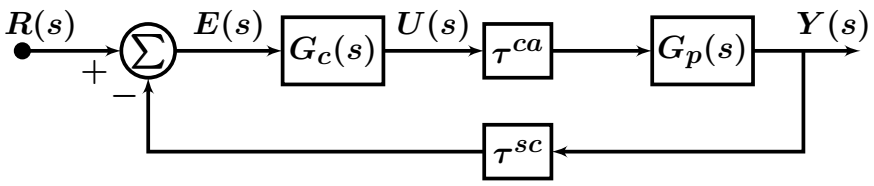


Figure 1.2: System deterioration caused by delays - closed-loop control structure.

A very simple example to simulate network delays, can be implemented by introducing some delays in a classical control structure shown in Figure 1.2. Following the example in [43], let the plant be a DC motor with transfer function

$$G_p(s) = \frac{2029.826}{(s + 26.29)(s + 2.296)}.$$

The controller was chosen to be a proportional-integral (PI) controller

$$G_c(s) = \frac{\beta K_P(s + \frac{K_I}{K_P})}{s},$$

where $K_P = 0.1701$ is the proportional gain, $K_I = 0.378$ is the integral gain, and β is a parameter to change both K_P and K_I, while keeping the ratio between these gains constant. Closed-loop is affected by delays τ^{ca} between controller and actuator, and τ^{sc} from sensor to controller. The example will consider $\tau^{ca} = \tau^{sc} = \frac{\tau}{2}$, where τ is the total closed-loop communication delay. Closed-loop step responses in Figure 1.3 show that longer delays will substantially degrade performances like overshoot and settling time.

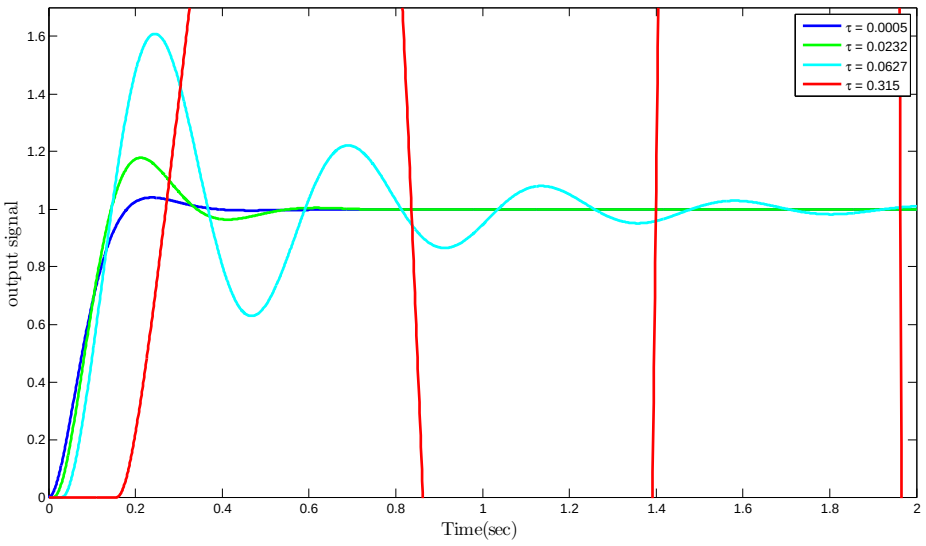


Figure 1.3: System deterioration caused by delays - closed-loop step responses for various values of closed-loop delay τ, and $\beta = 1$.

Figure 1.4 depicts the primary branches of the root locus with respect to β for different

delays, and focus on the stability region. It is obvious that, the longer the delay the smaller the stability margin, and thus, the smaller the range of stabilizing values for the closed-loop parameter β. It should be pointed out that stability analysis techniques are subject to network architectures and protocols.

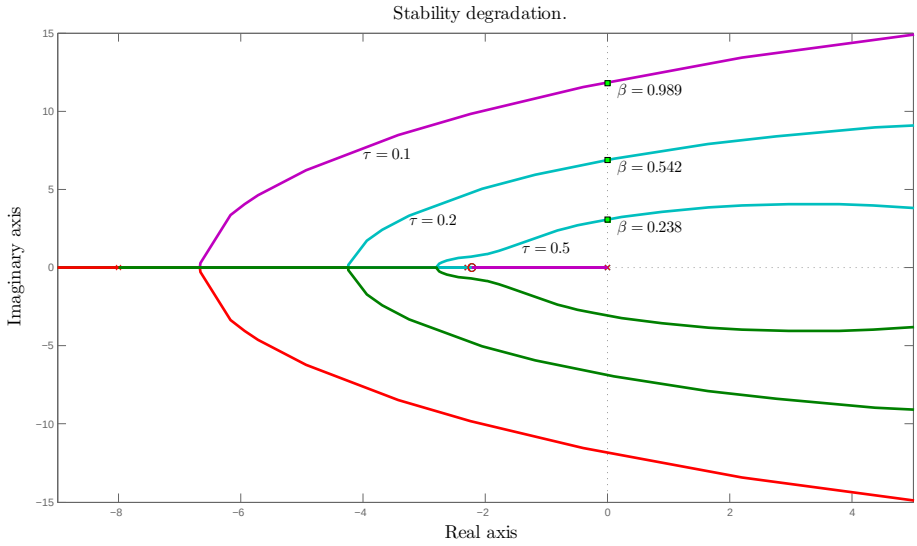


Figure 1.4: System deterioration caused by delays - primary branches of the root locus with respect to β for various values of closed-loop delay.

Packet Dropout. While transiting the network, data might be lost. The so-called *packet dropouts* could result from transmission errors in physical links, such as switches or routers (which are obviously more common in wireless networks than in wired ones), or from buffer overflows caused by data congestion. Communication experts would solve these incidents by retransmitting the lost data. Control researchers might think of a trade-off. Because, even though some signals coming from sensors might be lost while crossing the network, some NCSs could operate satisfactorily. Consequently, retransmission may not be the case for these systems, since it would imply additional delays, and, therefore, severe decrease in performance might occur.

Systems Architecture. The general NCS structure in Figure 1.1 is most of the times simplified under specific assumptions to make the analysis easier. For instance, some controllers are collocated with the actuators, and thus, are able to directly communicate.

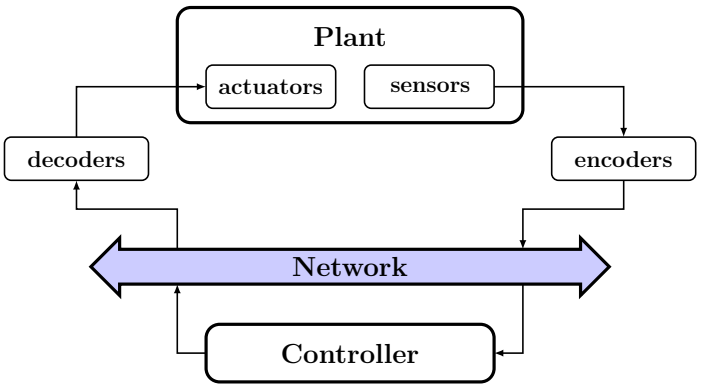


Figure 1.5: Single-loop NCS architecture.

Also, although significantly simpler than the multiple-loop structure in Figure 1.1, a single loop structure like the one in Figure 1.5 is still characterized by the same bandwidth limitations, communication delays, and packet dropouts.

Besides communication-constrained feedback control, a rather new aspect of NCS research is interested in distributed coordination of networks of mobile robot agents. This has lead to a more general area of interest, that of Multi-agent Systems (MASs). The reason for which such a field would raise the researchers' enthusiasm is the myriad applications that would benefit from it, such as formation control, flocking, scheduling and planning, distributed control, condition monitoring, to name just a few. The main issue in a network of agents is that, while each agent performs the task that it has been designed for, it has to act aware of its environment changes that might occur because of the other agents' actions. Therefore, communicating with the other agents in order to be able to readjust its behavior accordingly is very important. As pointed out in [6], there have been studied many strategies to solve this problem, depending on the desired final outcome of the MAS. Some of these outcomes and the proposed strategies are briefly discussed in the following paragraphs.

Maintain Rigid Formations. Starting with the simple problem of two mobile agents that are required to keep a constant distance between them while tending to their tasks, the idea of developing and controlling agent rigid formations on a larger scale triggered

the search for effective strategies to solve the problem. One of these strategies is the so-called *leader-follower* strategy. Typically, what happens is that each robot tries to regulate its distance with respect to a limited set of neighboring agents, which most of the times consists of only one or two more robots. In this small subset of agents, one is designated as the *leader*, and the others are the *followers*, which will constantly measure and try to readjust the value of the interagent distance. Moreover, an MAS formation is many times required to be *stably rigid*, meaning that the MAS should be able to overcome any small perturbation that might disturb the agents arrangement. In this case, each robot should restore its directed distances from the neighboring agents, such that the MAS can return to the formation that preceded the occurrence of perturbation.

Consensus Problems. Unlike the *leader-follower* strategy, the idea behind the consensus problems is to develop a certain consensus law, which will eventually determine all agents to move in a prescribed direction, even without a centralized controller to coordinate their movements, and regardless of the fact that by changing location each agent changes its closest neighbors.

Shaping Formation Movement. Other applications, such as controlling a group of satellites to cooperatively take images from space, not only require the agents to reach a relative distance from each other, but also position themselves on a particular curve and then move at constant speed.

Coverage Problems. Very similar to consensus problems, the coverage problems aim at controlling the agents in a distributed manner, so that they position themselves to cover the action area according to a given distribution.

1.2 Dissertation Contributions

Of all the aspects of NCS and MAS mentioned so far, this research focuses on the communication-constrained feedback control due to transmission channel bandwidth limitations, and on the consensus problem for a class of discrete-time MAS.

In the case of distributed systems, each component will only have access to a small

portion of the total communication bandwidth available to the network. Therefore, even though the network has a large number of bits allocated for communication, a considerable number of components sharing that bandwidth could raise the possibility of large quantization errors. In turn, these errors can drastically influence the system stability control, due to the low resolution of the transmitted data.

The first part of this research aims to develop some theoretical stability results concerning Linear Time Invariant (LTI) systems under a feedback law that uses only a finite number of fixed control values and a finite number of measurements levels. In other words, both the inputs and the outputs of the plant are quantized, which automatically leads to quantization of the system's states.

In [15], the authors prove that for a single-input linear discrete time invariant system, the least dense, or coarsest, quantizer that quadratically stabilizes it is the logarithmic quantizer. Moreover, the density of the quantizer is computed using a special case of the Linear Quadratic Regulator (LQR) method. An expression for the optimal logarithmic quantizer is also provided, which depends exclusively on the system unstable eigenvalues.

The idea that the logarithmic quantizer can be bounded by a sector, which depends entirely on its density, inspired the study in [16] to use the *sector bound* method to analyze the quantized state feedback problem. Thus, it reveals that the quantized feedback stabilization problem is strongly connected to the quadratic stabilization problem with sector bound uncertainties. The main conclusion of [16] is that feedback control problems can be translated into robust control problems by simply converting the quantization errors into sector bound uncertainties. Hence, by solving the equivalent $\mathcal{H}_\infty$ optimization problems, it is straight forward to obtain complete solutions to quadratic stabilization problem for Multiple-Input Multiple-Output (MIMO) quantized systems with either quantized state feedback or quantized output feedback.

While [16] deals with the cases in which either the control signal or the measurement is quantized, the proposed sector bound approach is extended in [10] to take on the case of

a SISO linear output feedback system affected by quantization at both input and output. Once again, by converting the quantized feedback control problem into a robust control problem, the objective of minimizing the quantity of information needed to be transmitted while stability is preserved is easily achieved and a bound for admissible quantization densities is provided.

To take it a step further, this research improves the quantization density bound by proposing a new control law. Since so far, the quantized input signal was considered unknown due to uncertainty, the approaches for feedback control with quantized input did not consider the possibility of making it available to the controller. In reality, this signal is known, and it is of great advantage to feed it back to the controller. Thus, the quantization effect is taken into account by the controller when adjusting the control input. Not only does the new proposed feedback law include the levels of the quantized output, but it also informs the controller about the possible lost information in the control input due to quantization.

First, the state feedback is studied with the control law augmented with the quantized values of the input. Since the quantization is modeled using the sector bound approach, the state feedback control problem becomes a Full Information (FI) control problem. Standard $\mathcal{H}_\infty$ control is then used to design the quantized state feedback controller. In the case when the states are not available, they can be estimated from the measurements in order to be able to design a state feedback law that is also augmented with the quantized input values. In this case, an observer-based controller is implemented to solve the problem.

The most challenging part comes with introducing quantization at both input and output simultaneously. Intuitively, the case of input-output feedback, that is when the controller is supplied with more information by including the quantized input in the control law besides the quantized output measurements, should be more efficient than the common output feedback law, and provide a larger upper bound for the quantization density. This research gives an analytical solution to the problem, by constructing the controllers for

both cases and comparing the parameters involved in the corresponding solutions. The conclusions are also supported by a series of simulation results.

The part of the thesis discussing the MASs investigates the consensusability problem for discrete-time multi agent dynamic systems, and how to synthesize the control law to ensure the agents reach consensus. Consensus problem for MASs has received a lot of attention lately in the case of both continuous-time and discrete-time systems. Particularly, in [30], the existence of consensus protocols for continuous-time LTI MASs is studied, and then strengthened by the formulation of a necessary and sufficient condition to achieve consensusability of this type of agents with respect to a set of admissible consensus protocols. An intriguing result detailed in [28] shows that the consensusability of continuous-time LTI MASs under directed communication topology is dependent on the ratio of the largest eigenvalue to the second smallest eigenvalue of the Laplacian of the connected graph. The same problem has been investigated for the case of discrete-time LTI MASs under undirected communication topology in [49], and emphasizes once again the importance of the eigenvalue ratio introduced in [28].

In this dissertation, the focus is on discrete-time LTI MASs whose communication topologies are modeled by directed graphs. A consensusability condition for both SISO and MIMO discrete-time dynamic agents is derived and an explicit synthesis procedure for designing the distributed consensus feedback control law is provided, such that, the agents asymptotically reach the prescribed consensus. The main contribution lies in adding an extra condition to the eigenvalue ratio condition, that would result in a strong sufficient consensusability condition. This extra condition regards the argument of the largest and second smallest eigenvalues of the Laplacian of the connected graph. Together, these two conditions that depend on the agent dynamics, delimit a fan shaped region where the Laplacian eigenvalues should reside in order for the MAS to reach consensus. Thus, the communication topology can be somehow easily set up by choosing the Laplacian eigenvalues to belong to the designed region.

1.3 Dissertation Structure

The rest of the dissertation is structured as follows.

Chapter 2 presents some preliminary results in the fields of linear algebra, discrete-time signals and systems, and robust control. Notions like vector and matrix norms, Kronecker product, signals and systems norms, robust stability, Linear Fractional Transformation (LFT), and small gain theorem are reviewed to ensure a better understanding of the problems discussed in the coming chapters.

Chapter 3 starts with introducing the closed-loop feedback issues due to communication constraints between the controlled system and the designed controller. In particular, the problem of quantized signals as a consequence of limited-bandwidth communication channels is presented, followed by some of the solutions that had been considered so far to find the largest upper bound for the quantization levels that would ensure stability. Then, a new control law is proposed to improve this upper bound. The results are further on backed up by comparing the controllers synthesized according to the old and new approaches, and by a series of simulations.

Chapter 4 tackles the consensusability problem of MASs. After an overview of the area of MASs, including some important definitions, a summary of the main characteristics of a directed graphs is given. Next sections describe the main subject of the chapter, that of discrete-time LTI MASs reaching consensus. The sufficient consensusability condition is studied by investigating the gain and phase margins under state feedback control for both single-input and multiple-input MASs under directed communication topologies.

The dissertation is concluded in Chapter 5, which also lists some ideas about future possible research topics.

Chapter 2
Preliminaries

This chapter reviews some of the basic concepts of linear algebra, signals and systems and their norms, LFT, and robust stability conditions, that are going to be used throughout this thesis.

2.1 Linear Algebra

While many of the basic linear algebra concepts and facts are very well known and detailed in most of the books covering the subject, including references such as [25, 27, 36], some of them are not too commonly used, and therefore easily disregarded. Hence, a reminder of some of these notions is presented in the following subsections.

2.1.1 Eigenvalues and Eigenvectors

For a square matrix $A \in \mathbb{C}^{n\times n}$, the *eigenvalues* are defined as the solutions to the equation $\det(\lambda I_n - A) = 0$. The set of all roots $\{\lambda_i\}_{i=1}^n$ satisfying this equation is called the *spectrum of* A, and, hence, the *spectral radius* of matrix A is defined as

$$\rho(A) := \max_{1\le i\le n} |\lambda_i|,$$

where $|\cdot|$ denotes the absolute value function.

If λ is an eigenvalue of a matrix A, then any nonzero vector $v \in \mathbb{C}^n$ satisfying

$$Av = \lambda v$$

is called the *right eigenvector* of A. Similarly, a nonzero vector u is referred to as the *left eigenvector* of A if

$$u^*A = \lambda u^*.$$

2.1.2 Matrix Inversion Formula

The following two identities, known as variants of the Woodbury identity, proved to be helpful in some of the later proofs.

$$(A + CBC^{\mathrm{T}})^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^{\mathrm{T}}A^{-1}C)^{-1}C^{\mathrm{T}}A^{-1}$$
$$(A + UBV)^{-1} = A^{-1} - A^{-1}U(B^{-1} + VA^{-1}U)^{-1}VA^{-1}.$$

Also, a particular case, called the Kailath variant, is also useful and shown below:

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}.$$

2.1.3 Vector Norms and Matrix Norms

Consider a vector $x \in \mathbb{C}^n$. The p-norm of x is defined as

$$\|x\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}, \text{ for } 1 \leq p \leq \infty.$$

The three most commonly used norms are, therefore, expressed as

$$\|x\|_1 := \sum_{i=1}^{n} |x_i|,$$
$$\|x\|_2 := \sqrt{\sum_{i=1}^{n} |x_i|^2},$$
$$\|x\|_\infty := \max_{1 \leq i \leq n} |x_i|.$$

For a matrix $A \in \mathbb{C}^{m \times n}$, the matrix norm induced by a vector p-norm is defined as

$$\|A\|_p := \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}.$$

For $p = 1,\, 2, \infty$, the related induced matrix norms are given by

$$\begin{aligned}
\|A\|_1 &:= \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}| \text{ (column sum)},\\
\|A\|_2 &:= \sqrt{\lambda_{\max}(A^*A)},\\
\|A\|_\infty &:= \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}| \text{ (row sum)}.
\end{aligned}$$

Since $\|A\|_p$ is induced from a vector p-norm, A can be considered to be a mapping from a vector space $\mathbb{C}^n$ equipped with a vector norm $\|\cdot\|_p$ to another vector space $\mathbb{C}^m$ equipped with a vector norm $\|\cdot\|_p$. Thus, in systems theory, the induced norms are viewed as input/output amplification gains.

2.1.4 Kronecker Product

Given two matrices $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$, the *Kronecker product* of A and B is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \in \mathbb{C}^{mp\times nq}.$$

Properties of the Kronecker product include

$$\begin{aligned}
A \otimes (B + C) &= A \otimes B + A \otimes C\\
A \otimes (B \otimes C) &= (A \otimes B) \otimes C\\
(\alpha_A A) \otimes (\alpha_B B) &= \alpha_A \alpha_B (A \otimes B)\\
(A \otimes B)^{\mathrm{T}} &= A^{\mathrm{T}} \otimes B^{\mathrm{T}}\\
(A \otimes B)(C \otimes D) &= (AC) \otimes (BD)\\
(A \otimes B)^{-1} &= A^{-1} \otimes B^{-1}.
\end{aligned}$$

2.2 Discrete-time Systems and Signals

This section reviews fundamental concepts of discrete-time systems and signals, focusing primarily on the way the strength of a system or signal is measured. Different norms are defined to measure the energy of a signal or a certain system gain to use as performance index of the control system.

2.2.1 Discrete-time Linear Dynamical System Models

The state-space description of a finite dimensional Linear Discrete Time Invariant (LDTI) dynamical system is given by a set of difference equations with constant coefficients:

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k),\ x(k_0) = x_0 \\ y(k) &= Cx(k) + Du(k), \end{cases}$$

where $x(k) \in \mathbb{R}^n$ is the *state* vector, $x(k_0)$ is the *initial condition* of the system, $u(k) \in \mathbb{R}^m$ is the *input* vector, $y(k) \in \mathbb{R}^p$ is the *output* vector, $A \in \mathbb{R}^{n\times n}$ is the *system dynamics matrix*, $B \in \mathbb{R}^{n\times m}$ is the *system input matrix*, $C \in \mathbb{R}^{p\times n}$ is the *system output matrix*, and $D \in \mathbb{R}^{p\times m}$ is the *system direct connection matrix*. If the system has one input ($m = 1$) and one output ($p = 1$), the system is called Single-Input Single-Output (SISO) system. Otherwise, it is know as a MIMO system.

An LDTI system is said to be asymptotically stable if and only if the eigenvalues of the system dynamics matrix A are less than unity in their magnitude, i.e. $|\lambda_i| < 1,\ i = 1, 2, \ , \ldots, n$. In this case, matrix A is also known as a *stability matrix*.

The systems dynamics can also be modeled using the *transfer matrix* from input u to output y as in

$$Y(z) = G(z)U(z),$$

where $U(z)$ and $Y(z)$ are the $\mathcal{Z}$-transform of $u(k)$ and $y(k)$, respectively, with zero initial

conditions. Thus, it can be written

$$G(z) = C(zI - A)^{-1}B + D \triangleq \left[\begin{array}{c|c} \text{A} & \text{B} \\ \hline \text{C} & \text{D} \end{array}\right].$$

2.2.2 Discrete-time Signal and System Norms

To quantify the concept of strength or energy of a signal, the mathematical notion of the norm is used for both continuous-time and discrete-time signals. The most important types of norms that could be defined for signals are discussed in the following paragraphs for discrete-time cases.

Let $x(k),\ k \in \mathbb{Z}_+$ be a discrete-time signal. The notion of the strength of $x(k)$ can be generalized through the definition of p-norm associated with the $l_p[0, \infty]$ space for $1 \leq p < \infty$:

$$\|x\|_p = \left(\sum_{k=1}^{\infty} |x(k)|^p\right)^{1/p}.$$

In particular, and with different physical meanings, the following signal norms can be defined:

- 1-norm of $x(k)$, also known as *the action of the signal*

$$\|x\|_1 = \sum_{k=1}^{\infty} |x(k)|$$

- 2-norm of $x(k)$, also known as *square root of the energy of the signal*

$$\|x\|_2 = \sqrt{\sum_{k=1}^{\infty} |x(k)|^2}$$

- ∞-norm of $x(k)$, also known as *the amplitude of the signal*

$$\|x\|_\infty = \sup_{k\in\mathbb{Z}_+} |x(k)|,$$

  which is the least upper bound of $|x(k)|$.

Since a transfer matrix models the transfer from an input signal to an output signal, it could be considered as an operator from the input space to the output space, on which a norm is induced. Thus, by knowing the norm of a stable dynamical system, the size of its output could be easily determined for a given input. As for signals, there are several types of norms that could be defined for systems. Nevertheless, only the $\mathcal{H}_\infty$ norm is discussed, since it is the one the makes the tool for stability studies in this research.

Let $l_2[0,\, \infty)$ be the set of all real square summable sequences, i.e.

$$l_2[0,\, \infty) = \{x : \|x\|_2^2 < \infty\}.$$

For a mapping $G : l_2[0,\, \infty) \mapsto l_2[0,\, \infty)$, the $\mathcal{H}_\infty$ norm of G is defined as

$$\|G\|_\infty := \sup_{x\neq 0\in l_2[0,\,\infty)} \frac{\|Gx\|_2}{\|x\|_2},$$

symbolizing the maximum possible energy amplification or reduction. In particular, if $G(z)$ denotes the proper and real rational stable transfer matrix of an LDTI dynamic system, the $\mathcal{H}_\infty$ norm is defined as

$$\|G(z)\|_\infty := \sup_{|z|>1} \bar{\sigma}[G(z)],$$

where $\bar{\sigma}[G(z)]$ is the maximum singular value of $G(z)$.

2.3 Robust Control

The theorem below is useful in testing robust stability conditions under distinct suppositions on the type of model uncertainty. The notion *model uncertainties* refers to the

inevitable errors that occur while trying to describe the real life behavior of a system through a mathematical model. With $P(z)$ representing the *nominal plant model*, the uncertain models are generally given in one of the following forms:

1. additive form:

$$P_\Delta(z) = P(z) + W_1(z)\Delta(z)W_2(z),\ \bar{\sigma}(\Delta(z)) < 1,\ \forall\, |z| > 1,$$

   where the modeling error $\Delta(z)$ is scaled using the stable weighting transfer functions $W_1(z)$ and $W_2(z)$ to better characterize the frequency structure of the uncertainty;

2. multiplicative form:

$$P_\Delta(z) = (I + W_1(z)\Delta(z)W_2(z))P(z).$$

Following the terminology in [51], given Π the set of uncertain models, suppose $P(z) \in \Pi$ is the nominal model and $K(z)$ is the resulting controller. The closed-loop system is said to be

- *nominally stable* if $K(z)$ internally stabilizes the nominal model $P(z)$,
- *robustly stable* if $K(z)$ internally stabilizes all the models in Π.

If in the LFT in Figure 2.1, $G(z)$ is partitioned as

$$G(z) = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \in \mathbb{C}^{(p_1+p_2)\times(q_1+q_2)},$$

then the *lower LFT* with respect to $K(z)$ is defined as

$$\mathcal{F}_l(G(z), K(z)) := G_{11}(z) + G_{12}(z)K(z)(I - G_{22}(z)K(z))^{-1}G_{21}(z),$$

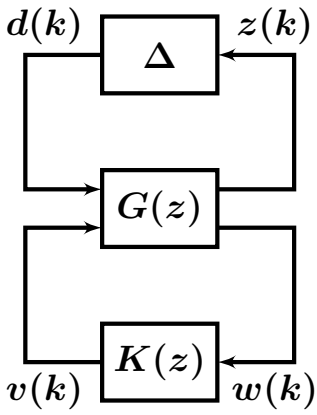


Figure 2.1: Linear fractional transformation.

provided the inverse of $(I - G_{22}(z)K(z))$ exists. Also, the *upper LFT* with respect to Δ is given by

$$\mathcal{F}_u(G(z), \Delta) := G_{22}(z) + G_{21}(z)\Delta(I - G_{11}(z)\Delta)^{-1}G_{12}(z),$$

provided $(I - G_{11}(z)\Delta)^{-1}$ exists.

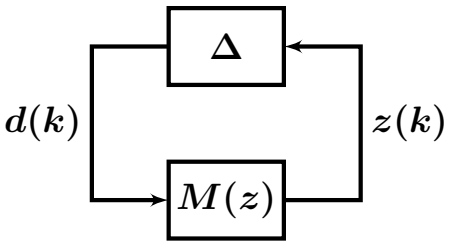


Figure 2.2: Small gain theorem.

Theorem 2.3.1 ([51] Small Gain Theorem)**.** *Suppose $M(z)$ is a rational proper transfer matrix, that could model a stable linear closed-loop system consisting of the plant, controller and various weights, and let $\gamma > 0$. Then the LFT in Figure 2.2 is well defined and internally stable for all Δ, and*

(a) $\|\Delta\|_\infty \leq \dfrac{1}{\gamma}$ *if and only if* $\|M(z)\|_\infty < \gamma$*;*

(b) $\|\Delta\|_\infty < \dfrac{1}{\gamma}$ *if and only if* $\|M(z)\|_\infty \leq \gamma$*.*

When the uncertainty block Δ is in a diagonal form comprising multiple uncertainty blocks, the small gain condition in Theorem 2.3.1 could turn out to be a conservative

sufficient condition for robust stability. Therefore, in the case of three or less uncertainty blocks, a reasonable approach would be to redraw the structure in Figure 2.2 as in Figure 2.3, where W_l and W_r are constant scaling matrices such that $W_r\Delta = \Delta W_l$. Then the small gain theorem can be extended to time-varying structured uncertainty. Consequently, the robust stability holds and $\|\Delta\|_\infty < \frac{1}{\gamma}$ if and only if $\|W_l M(z) W_r^{-1}\|_\infty \leq \gamma$. It follows that robust stability could be, in this case, tested by doing a search for the scaling matrices that would entail the small gain condition to be non-conservative.

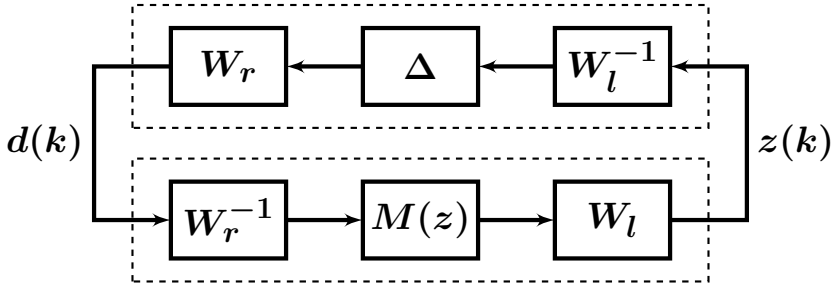


Figure 2.3: Small gain theorem - structured time-varying uncertainty case.

Chapter 3
Communication-constrained Feedback Stability in NCS

For years, control and communications researchers have worked on their own problems independently. Communications theory mostly cares about reliably transmitting the information from sender to receiver, with none or less concern on a particular purpose of the transmitted data, or whether it would finally be fed back to the source. On the other hand, control theory focuses on using the information in a feedback loop to ensure stability and desired performances, while assuming the communication links do not significantly influence these performances.

It is advisable to treat communication and control as two separate problems for engineering systems with large bandwidth, in order to simplify the analysis and design of the overall system. Nevertheless, due to recent advances in technology, applications like mobile telephony, sensor networks, micro-electromechanical systems, or industrial control networks, have required a different approach. The main purpose of these applications is to control a series of dynamical systems using multiple actuators and sensors, which exchange information over a digital communication network.

For such distributed systems, out of the total communication capacity of the network only a small quota is allocated to each component. Therefore, even though the total capacity in bits per second might be large, there still exists the possibility of large quantization errors that infringe upon control performance, due to the low resolution of the transmitted data.

Over the decades, there has been a fairly large amount of research concerning quantization errors. Back in the 1970s, quantization was modeled as an extra additive white noise, which allowed results from stochastic control to be applied [12]. Research has proven, though, that while reasonable for high resolution quantizers, this approach fails when the

quantizer has a coarse resolution and the open-loop dynamics are unstable. It has been recently discovered, for example in [5], [44], that there exists a critical positive data rate below which an unstable plant cannot be stabilized by any quantization and control scheme. This important result suggests that control performance is strongly and negatively affected by a low communication capacity. Therefore, a more thorough analysis is required, in which communication and control aspects should be considered jointly rather than independently.

The simplest network topology used to study the networked control systems is depicted in Figure 1.5. It consists of one dynamical system and its corresponding controller connected through a feedback network with a specified data rate in bits per unit time. Having set the closed loop structure, the most essential question related to it is whether or not there exists a lower limit for the feedback data rate above which a stabilizable controller can be designed. The problem is very similar to Shannon's source coding theory that determines the smallest data rate above which a given random process can be reliably communicated, that is, with arbitrarily small probability of error [11], [39]. Nonetheless, the most significant difference consists in the fact that for control systems, data is not just transmitted between two points, but also used in the feedback loop.

Once the stability problem has been overcome, the next issue of great concern is the trade-off between the communication rate and the optimal control performance that could be achieved. This research idea is the control theory counterpart of Shannon's rate-distortion theory for digital communications [8], [40].

The main objective of this research is to develop some theoretical stability results concerning LDTI systems using only a finite number of fixed control values and a finite number of measurements levels. In other words, both the inputs and the outputs of the plant are quantized, which automatically leads to quantization of the system's states.

3.1 Previous Approaches and Strategies

3.1.1 Stabilization Using Quantized State Feedback

In [15], unlike the approach in this research, quantization is considered as a useful process. Understanding how to systematically quantize information while preserving stability and desired performance without treating quantization as noise or uncertainty is the main purpose of [15]. The simplest case considered in [15] is the quadratic stabilization of the LDTI system

$$x(k+1) = Ax(k) + Bu(k) \tag{3.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, $x \in \mathbb{R}^{n\times 1}$ is the state, and $u \in \mathbb{R}$ is the control input. It is assumed that A is unstable and (A, B) is stabilizable. The quantized state feedback is modeled by

$$\begin{aligned} u(k) &= f(v(k)) \\ v(k) &= g(x(k)) \end{aligned} \tag{3.2}$$

where $g(\cdot)$ represents the *unquantized feedback law*, and $f(\cdot)$ is a static (memoryless), time-invariant (fixed quantization levels), and symmetric quantizer ($f(-v) = -f(v)$).

The collection $\{v(k)\}_{k\in\mathbb{Z}}$ is encoded into a set of distinct quantized levels

$$\mathcal{U} = \{\pm u_i, i = \pm 1, \pm 2, \dots\} \cup \{0\}, \tag{3.3}$$

each of the quantization level u_i corresponding to a segment V_i such that the quantizer maps the entire segment to the same quantization level.

The density of a quantizer $f(\cdot)$ is defined as:

$$n_f = \limsup_{\varepsilon\to 0} \frac{\#g[\varepsilon]}{-\ln \varepsilon} \tag{3.4}$$

where $\#g[\varepsilon]$ denotes the number of quantization levels in the interval $[\varepsilon, 1/\varepsilon]$. A *coarse*, or least dense, quantizer has a large space between levels, which implies a small n_f.

A *logarithmic* quantizer has the set of quantized levels defined by

$$\mathcal{U} = \{\pm u_i : u_i = \rho^i u_0, i = \pm 1, \pm 2, \dots\} \cup \{u_0\} \cup \{0\}, 0 < \rho < 1, u_0 > 0, \tag{3.5}$$

and the associated quantizing function given by:

$$f(v) = \begin{cases} u_i, & \text{if } \frac{1}{1+\delta} u_i < v \leq \frac{1}{1-\delta} u_i,\ v > 0 \\ 0, & \text{if } v = 0 \\ -f(-v), & \text{if } v < 0 \end{cases} \tag{3.6}$$

where

$$\delta = \frac{1-\rho}{1+\rho}. \tag{3.7}$$

Applying the definition in (3.4), the density of a logarithmic quantizer is $n_f = \dfrac{2}{\ln \frac{1}{\rho}}$. This entails that the smaller the parameter ρ, the smaller the density n_f, and therefore, in specialized literature, ρ is referred to as the quantization density instead of n_f. The *logarithmic quantizer* is graphically exemplified in Figure 3.1.

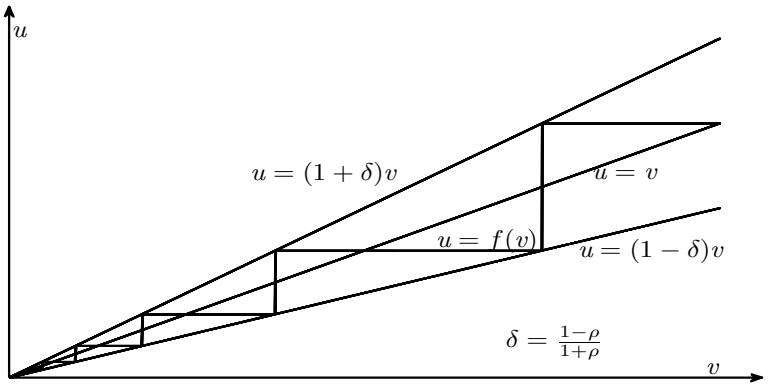


Figure 3.1: Logarithmic quantization.

In [15], the quadratic stabilization problem is studied. For this purpose, a quadratic Lyapunov function $V(x) = x^T P x$, $P = P^T > 0$ is used to evaluate the stability of the

feedback system. In other words, the quantizer must satisfy:

$$\begin{aligned} \nabla V(x) &= V(x(k+1)) - V(x(k)) \\ &= V(Ax + Bf(g(x))) - V(x) < 0, \ \forall x \neq 0. \end{aligned} \tag{3.8}$$

The quantizer that minimizes n_f subject to (3.8) is called the *coarsest* quantizer. But, since the constraint in (3.8) is a strict inequality, the exact coarsest quantizer cannot be achieved.

The density of the quantizer is strictly dependent on the choices for $V(x)$ (or P), $f(\cdot)$, and $g(\cdot)$. Therefore, the fundamental question that arises is what the coarsest density, ρ_{inf}, could be among all possible P and $g(\cdot)$. It has been proven in [15] that the best choice would be a logarithmic quantizer with the density given by

$$\rho_{\text{inf}} = \frac{\prod_i |\lambda_i^u| - 1}{\prod_i |\lambda_i^u| + 1} \tag{3.9}$$

where λ_i^u denote the unstable eigenvalues of the system matrix A.

As depicted in Figure 3.1, the logarithmic quantizer $f(v)$ can be bounded by a sector $(1 + \Delta)v$, $\Delta \in [-\delta, \delta]$, described by only one parameter δ, which relates to the quantizer density by (3.7). This idea inspired the study in [16] that uses the *sector bound* method to analyze the quantized state feedback problem for the system (3.1). This study reveals that there is a strong connection between the quantized state feedback stabilization problem and a state feedback quadratic stabilization problem with sector bound uncertainty. This newly discovered connection entails an alternative proof for the coarsest quantization density in (3.9), and it is summarized in the following theorem [16].

Theorem 3.1.1. *The following results hold.*

1. *If the system (3.1) is quadratically stabilizable via quantized state feedback (3.2) -*

(3.3), then the coarsest quantization density can be approached by taking a logarithmic quantizer and a linear unquantized feedback law.

2. *Given a logarithmic quantizer with quantization density ρ, the system (3.1) is quadratically stabilizable via quantized linear state feedback if and only if the following uncertain system:*

$$x(k+1) = Ax(k) + B(1+\Delta)v(k),\ \Delta \in [-\delta, \delta] \tag{3.10}$$

is quadratically stabilizable via linear state feedback, where δ and ρ are related by (3.7).

3. *The largest sector bound for (3.10) to be quadratically stabilizable via linear state feedback is given by*

$$\delta_{\sup} = \frac{1}{\prod_i |\lambda_i^u|}. \tag{3.11}$$

Consequently, the coarsest quantization density $\rho_{\inf}$ for (3.1) is given by (3.9).

It is well known from [35], [13], and [47] that to quadratically stabilize system (3.10) using a linear state feedback

$$g(x) = Kx, \tag{3.12}$$

the uncertainty bound should be upper bounded by

$$\delta_{\sup} = \frac{1}{\inf_K \|G_c(z)\|_\infty}, \tag{3.13}$$

where

$$G_c(z) = K(zI - A - BK)^{-1}B. \tag{3.14}$$

Therefore, it is more appealing to look at the coarsest quantization problem as an $\mathcal{H}_\infty$ optimization problem (3.13), because they share the same optimal control gain.

3.1.2 Stabilization Using Quantized Output Feedback

The next step was to apply the technique for state feedback to quantized output feedback for the system

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{cases} \tag{3.15}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, and $C \in \mathbb{R}^{1\times n}$. [16] deals with the cases in which either the control signal or the measurement is quantized, while [10] tackles the problem of simultaneously applying quantization to both input and output.

When only the control signal is quantized, it turns out that, under the assumption that the pair (A, C) is observable, the coarsest quantization density achievable by output feedback is the same as the one obtained when state feedback is used. Moreover, the output feedback controller can be chosen as an observer-based controller

$$\begin{cases} x_c(k+1) &= Ax_c(k) + Bu(k) + L(y(k) - Cx_c(k)) \\ v(k) &= Kx_c(k) \\ u(k) &= f(v(k)), \end{cases} \tag{3.16}$$

where $f(\cdot)$ is the quantizer, K is the state feedback gain, and L is any gain such that (3.16) is a dead-beat observer.

In the case when only measurement is quantized, the controller takes the form

$$\begin{cases} x_c(k+1) &= A_c x_c(k) + B_c f(y(k)) \\ u(k) &= C_c x_c(k) + D_c f(y(k)) \end{cases} \tag{3.17}$$

where $f(\cdot)$ is the quantizer. A generalized version of the state feedback case can be used to solve the problem of finding the coarsest quantizer for quadratic stabilization of the closed loop system. The result is summed up in the following theorem [16].

Theorem 3.1.2. *Consider the system (3.15). For a given quantization density $\rho > 0$,*

the system is quadratically stabilizable via a quantized controller (3.17) if and only if the following auxiliary system:

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ v(k) &= (1+\Delta)Cx(k),\ |\Delta| \le \delta \end{cases} \tag{3.18}$$

is quadratically stabilizable via

$$\begin{cases} x_c(k+1) &= A_c x_c(k) + B_c v(k) \\ u(k) &= C_c x_c(k) + D_c v(k) \end{cases} \tag{3.19}$$

where δ and ρ are related by (3.7). The largest sector bound $\delta_{\sup}$ (which gives $\rho_{\inf}$) is given by

$$\delta_{\sup} = \frac{1}{\inf_{\bar{K}} \|\bar{G}_c(z)\|_\infty} \tag{3.20}$$

where

$$\begin{aligned} \bar{K}(z) &= C_c(zI - A_c)^{-1}B_c + D_c, \\ G(z) &= C(zI - A)^{-1}B \\ \bar{G}_c(z) &= (1 - \bar{K}(z)G(z))^{-1}\bar{K}(z)G(z) \end{aligned} \tag{3.21}$$

Further, if $G(z)$ has relative degree equal to 1 and no unstable zeros, then the coarsest quantization density for quantized state feedback can be reached via quantized output feedback.

The key conclusion of [16] is that quantized feedback control problems can be transformed into robust control problems by simply converting quantization errors into sector bound uncertainties. Thus, for SISO and MIMO quantized systems, quadratic stabilization problem (with either quantized state feedback or quantized output feedback) has complete solutions obtained by solving the equivalent $\mathcal{H}_\infty$ optimization problems.

The sector bound approach in [16] is extended in [10] to assist with the situation in which a SISO linear output feedback system is simultaneously affected by quantization at both input and output. This is very common in practical situation these days, because the feedback information (control signal and measurements) is exchanged by the control system components, such as sensors, and actuators, through a shared communication channel. Once again, by using the sector bound approach, the quantized feedback control problem is converted into a robust control problem. Thus, the objective of minimizing the quantity of information needed to be transmitted while stability is preserved is easily achieved and a bound for admissible quantization densities is provided. Consider the system (3.15) is controlled by a dynamic controller

$$\begin{cases} \xi(k+1) &= A_c\xi(k) + B_c v(k) \\ w(k) &= C_c\xi(k) + D_c v(k). \end{cases} \tag{3.22}$$

The quantizers are modeled by

$$\begin{cases} u(k) &= Q_1(w(k)), \text{ input quantizer} \\ v(k) &= Q_2(y(k)), \text{ output quantizer,} \end{cases} \tag{3.23}$$

where $Q_1(\cdot)$ and $Q_2(\cdot)$ are logarithmic quantizers with quantization levels ρ_1, and ρ_2, respectively. With controller (3.22), the closed-loop system can be modeled by

$$\begin{cases} x(k+1) &= Ax(k) + BQ_1(C_c\xi(k) + D_cQ_2(Cx(k))) \\ \xi(k+1) &= A_c\xi(k) + B_cQ_2(Cx(k)). \end{cases} \tag{3.24}$$

Applying the bounded sector approach to both quantizers and extending the results in [16] to the double quantization problem, it follows in [10] that the quadratic stability of the closed loop system (3.15), (3.22), and (3.23) is equivalent to the quadratic stability of the

auxiliary system

$$\begin{cases} x(k+1) &= Ax(k) + B(1+\Delta_1)w(k) \\ \xi(k+1) &= A_c\xi(k) + B_c(1+\Delta_2)y(k) \\ y(k) &= Cx(k) \\ w(k) &= C_c\xi(k) + D_c(1+\Delta_2)y(k) \end{cases} \tag{3.25}$$

where $|\Delta_1| \leq \delta_1$, $|\Delta_2| \leq \delta_2$, and the parameters δ_i are related to the quantization densities ρ_i through $\delta_i = \dfrac{1-\rho_i}{1+\rho_i}$, $i = 1, 2$.

Using standard linear fractional transformation [51], and defining

$$\begin{gathered} \widehat{A} = \begin{bmatrix} A + BD_cC & BC_c \\ B_cC & A_c \end{bmatrix}, \ \widehat{B} = \begin{bmatrix} B & BD_c \\ 0 & B_c \end{bmatrix}, \\ \widehat{C} = \begin{bmatrix} D_cC & C_c \\ C & 0 \end{bmatrix}, \ \widehat{D} = \begin{bmatrix} 0 & D_c \\ 0 & 0 \end{bmatrix}, \\ q(k) = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \ p(k) = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \\ q_1 = w(k),\ q_2 = y(k),\ p_1 = \Delta_1 q_1,\ p_2 = \Delta_2 q_2, \end{gathered} \tag{3.26}$$

the closed-loop system (3.25) can be rewritten as

$$\begin{cases} \bar{x}(k+1) &= \widehat{A}\bar{x}(k) + \widehat{B}p(k) \\ q(k) &= \widehat{C}\bar{x}(k) + \widehat{D}p(k) \\ p(k) &= \widehat{\Delta}q(k),\ \widehat{\Delta} = \mathrm{diag}\{\Delta_1,\ \Delta_2\}. \end{cases} \tag{3.27}$$

Denote by

$$\widehat{G}(z) = \widehat{C}(zI - \widehat{A})^{-1}\widehat{B} + \widehat{D} \tag{3.28}$$

the transfer function matrix from $p(k)$ to $q(k)$ for the open-loop system in (3.27). Then, by the small gain theorem, the closed-loop system is quadratically stable if the following

condition holds

$$\|\widehat{G}(z)\|_\infty \|\widehat{\Delta}\|_2 < 1. \tag{3.29}$$

The small-gain condition turns out to be both necessary and sufficient to ensure the quadratic stability of system (3.27) in the case of a single uncertainty block [14]. Due to the conservative nature of the small-gain condition in the matter of multiple uncertainty blocks, the condition is scaled to become

$$\|T\widehat{G}(z)T^{-1}\|_\infty \|\widehat{\Delta}\|_2 < 1, \tag{3.30}$$

where T is any invertible diagonal matrix [38]. Without loss of generality, T can be chosen as $T = \text{diag}\{1, \tau\}$, $\tau > 0$.

In consideration of the above statements, the quadratic stability of the closed-loop system (3.15), (3.22), and (3.23) is determined by the following theorem [10].

Theorem 3.1.3. *Consider the system (3.15) and quantizers as in (3.23) with given densities ρ_1 and ρ_2. This system is quadratically stabilizable via the controller (3.22) if and only if the auxiliary system*

$$\begin{cases} x(k+1) &= Ax(k) + B(1+\Delta_1)w(k) \\ v(k) &= (1+\Delta_2)Cx(k),\ |\Delta_i| \le \delta_i, i = 1,2 \end{cases} \tag{3.31}$$

is quadratically stabilizable via the controller (3.22). This, in turn, is guaranteed when $\delta_i < \widehat{\delta}_{\text{sup}}$, $i = 1, 2$, where

$$\widehat{\delta}_{\text{sup}} = \frac{1}{\inf\limits_{K,T} \|T\widehat{G}(z)T^{-1}\|_\infty} \tag{3.32}$$

with $\widehat{G}(z)$ as given in (3.28), T is a diagonal and invertible matrix and

$$K(z) = C_c \left(zI - A_c\right)^{-1} B_c + D_c.$$

3.2 Current Approaches and Strategies

3.2.1 State Feedback

So far, the approaches for state feedback control with logarithmically quantized input did not consider the quantized control signal $u(k)$ as known, and thus, available to the controller. But, since the quantized input is known, it could be of great advantage to feed it back to the controller. The quantizing effect can thus be taken into account by the controller when adjusting the control input. This could also be desirable in case rigorous system performance needs to be studied later. The new proposed feedback law is shown in Figure 3.2, where the quantization has been modeled using the sector bound approach. When the input quantization process is modeled as in Figure 3.2, the control input becomes

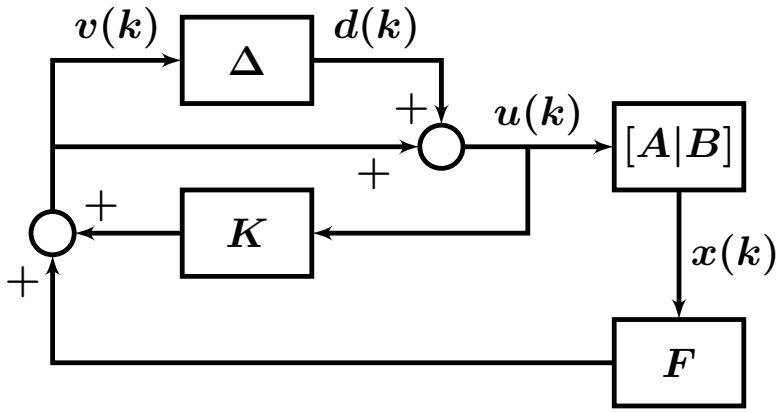


Figure 3.2: Illustration of the quantized feedback system under state feedback.

$$u(k) = v(k) + d(k) = (1 + \Delta)v(k), \tag{3.33}$$

and thus the systems equations are

$$x(k+1) = Ax(k) + Bd(k) + Bv(k), \tag{3.34}$$

with the control law

$$v(k) = Fx(k) + Ku(k). \tag{3.35}$$

Substituting (3.33) in (3.34) implies $v(k) = Fx(k) + Kv(k) + Kd(k)$, which yields $v(k) = (I-K)^{-1}Fx(k) + (I-K)^{-1}Kd(k)$. Denote $F_1 = (I-K)^{-1}F$ and $F_2 = (I-K)^{-1}K$ to write

$$v(k) = \begin{bmatrix} F_1 & F_2 \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix}. \tag{3.36}$$

It is assumed that K has no unitary eigenvalues. Also, since the quantization is modeled using the sector bound approach, Δ is a bounded uncertainty, and therefore, $d(k)$ can be looked at as an energy bounded disturbance. Hence, the state feedback control problem becomes a FI control problem. Specifically, this means that, if $T_{vd}(z)$ denotes the closed loop transfer matrix from disturbance $d(t)$ to control input $v(t)$ with the state-space realization

$$\begin{cases} x(k+1) &= Ax(k) + Bd(k) + Bv(k), \\ v(k) &= F_1x(k) + F_2d(k), \end{cases} \tag{3.37}$$

the goal is to design an FI control law that will stabilize the closed loop system and minimize $\|T_{vd}(z)\|_\infty$.

Standard $\mathcal{H}_\infty$ control ensures that $\|T_{vd}(z)\|_\infty < \gamma$ for some $\gamma > 0$ if and only if there exists a solution $X \geq 0$ to the discrete algebraic Riccati equation (DARE)

$$X = A^*X[I + (1-\gamma^{-2})BB^*X]^{-1}A, \tag{3.38}$$

and

$$\gamma^2 I - B^*X(I + BB^*X)^{-1}B > 0. \tag{3.39}$$

Since $X \geq 0$, the matrix inequality holds for any $\gamma > 1$. Thus, it can be concluded that the coarsest quantization density is 1, the largest possible. If $X \geq 0$ is a stabilizing solution to

the DARE (3.38), then the optimal FI control gains are given by

$$\begin{bmatrix} F_1 & F_2 \end{bmatrix} = -(I + B^*XB)^{-1}B^*X \begin{bmatrix} A & B \end{bmatrix}. \tag{3.40}$$

With this controller, the corresponding closed loop transfer matrix is given by

$$T_{vd}(z) = \left[\begin{array}{c|c} A + BF_1 & B(I + F_2) \\ \hline F_1 & F_2 \end{array}\right]. \tag{3.41}$$

It is noted that $I + F_2 = (I + B^*XB)^{-1}$, and therefore, invertible. By the definition $F_2 = (I - K)^{-1}K$, the controller components can be obtained as

$$\begin{aligned} K &= (I + F_2)^{-1}F_2 = -B^*XB, \\ F &= (I + F_2)^{-1}F_1 = -B^*XA. \end{aligned} \tag{3.42}$$

3.2.2 Output Feedback

If the states are not available and only the outputs $y(k)$ in (3.16) are measured, the system's transfer matrix becomes $P(z) = C(zI - A)^{-1}B$. Moreover, estimated states are needed in order to be able to design a state feedback control law. Let $\widehat{x}(k)$ denote the estimated state vector, and $e_x(k) = \widehat{x}(k) - x(k)$ the estimation error vector. Thus, the control law becomes

$$v(k) = F_1\widehat{x}(k) + F_2d(k) = F_1x(k) + F_2d(k) + F_1e_x(k). \tag{3.43}$$

Since $d(k) = u(k) - v(k)$ and both $u(k)$ and $v(k)$ are known to the controller, $d(k)$ is also accessible to the controller. This implies

$$v(k) = F_1\widehat{x}(k) + F_2[u(k) - v(k)] = (I + F_2)^{-1}F_1\widehat{x}(k) + (I + F_2)^{-1}F_2u(k), \tag{3.44}$$

where $I + F_2 = (I + B^*XB)^{-1}$, and therefore invertible.

An observer-based controller is employed in the following rationale. Thus, assuming the matrix L is such that $A + LC$ is a stability matrix, the estimated state vector is given by the standard observer

$$\begin{aligned} \widehat{x}(k+1) &= A\widehat{x}(k) + L[C\widehat{x}(k) - y(k)] + Bu(k) \\ &= (A + BF_1 + LC)\widehat{x}(k) - Ly(k) + B(I + F_2)d(k). \end{aligned} \tag{3.45}$$

Putting together (3.43) and (3.45), the observer-based controller has the transfer function

$$\begin{aligned} K(z) = \begin{bmatrix} K_1(z) & K_2(z) \end{bmatrix} &= \left[\begin{array}{c|cc} A + BF_1 + LC & B(I + F_2) & -L \\ \hline F_1 & F_2 & 0 \end{array}\right] \\ &= \left[\begin{array}{c|cc} \widetilde{A} & \widetilde{B}_1 & \widetilde{B}_2 \\ \hline \widetilde{C} & \widetilde{D}_1 & \widetilde{D}_2 \end{array}\right], \end{aligned} \tag{3.46}$$

and the closed loop structure is depicted in Figure 3.3.

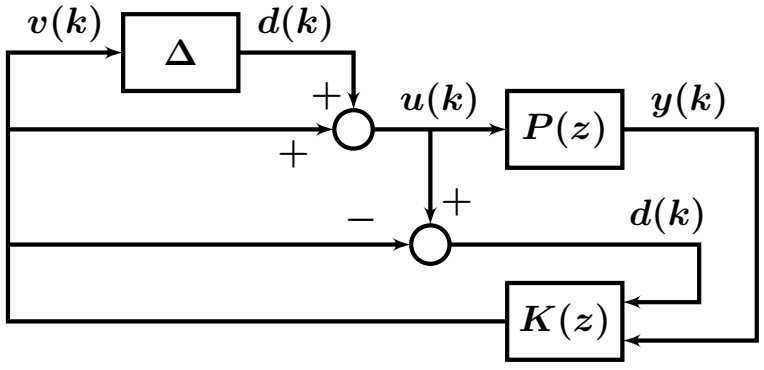


Figure 3.3: Illustration of the quantized feedback system under output feedback.

By (3.34) and (3.43), there holds

$$x(k+1) = Ax(k) + BF_1\widehat{x}(k) + B(I + F_2)d(k). \tag{3.47}$$

Taking the difference between equations (3.45) and (3.47), and recalling the state estimation error $e_x(k) = \widehat{x}(k) - x(k)$, the error dynamics are characterized by

$$e_x(k+1) = (A + LC)e_x(k), \tag{3.48}$$

which represents the unreachable subsystem of the closed loop system. Equation (3.47) can be rewritten as

$$x(k+1) = (A + BF_1)x(k) + BF_1 e_x(k) + B(I + F_2)d(k). \tag{3.49}$$

Thus, the closed loop state space model becomes

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ e_x(k+1) \end{bmatrix} &= \begin{bmatrix} A + BF_1 & BF_1 \\ 0 & A + LC \end{bmatrix} \begin{bmatrix} x(k) \\ e_x(k) \end{bmatrix} + \begin{bmatrix} B(I + F_2) \\ 0 \end{bmatrix} d(k), \\ v(k) &= \begin{bmatrix} F_1 & F_2 \end{bmatrix} \begin{bmatrix} x(k) \\ e_x(k) \end{bmatrix} + F_2 d(k), \end{aligned} \tag{3.50}$$

and the goal is to design a stabilizing L such that $\|T_{vd}(z)\|_\infty < \gamma$, where

$$T_{vd}(z) = \left[\begin{array}{cc|c} A + BF_1 & BF_1 & B(I + F_2) \\ 0 & A + LC & 0 \\ \hline F_1 & F_1 & F_2 \end{array}\right]. \tag{3.51}$$

After the unreachable modes $e_x(k)$ are eliminated, $T_{vd}(z) = \left[\begin{array}{c|c} A + BF_1 & B(I + F_2) \\ \hline F_1 & F_2 \end{array}\right]$, which is exactly the same state space realization as in (3.41). Therefore the problem is equivalent to the FI control problem previously discussed, and the same $\mathcal{H}_\infty$ norm is achieved.

If the control law is implemented like in the state feedback case, using the estimated state and the plant quantized input, (3.43) and (3.45) become

$$\begin{aligned} \widehat{x}(k+1) &= A\widehat{x}(k) + L[C\widehat{x}(k) - y(k)] + Bu(k) \\ &= (A+LC)\widehat{x}(k) + Bu(k) - Ly(k), \\ v(k) &= F\widehat{x}(k) + Ku(k). \end{aligned} \tag{3.52}$$

This is in fact the standard state observer in Figure 3.4, where

$$\widehat{K}(z) = \begin{bmatrix} \widehat{K}_1(z) & \widehat{K}_2(z) \end{bmatrix} = \left[\begin{array}{c|cc} A+LC & B & -L \\ \hline F & K & 0 \end{array}\right] = \left[\begin{array}{c|cc} \widehat{A} & \widehat{B}_1 & \widehat{B}_2 \\ \hline \widehat{C} & \widehat{D}_1 & \widehat{D}_2 \end{array}\right]. \tag{3.53}$$

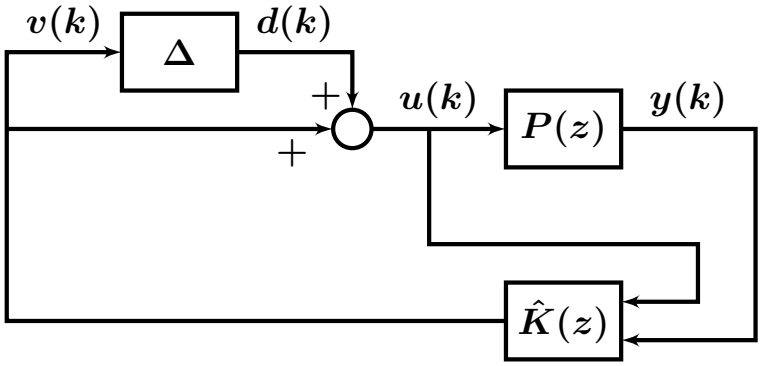


Figure 3.4: Equivalent block diagram for quantized feedback system under output feedback.

It can be verified that the feedback controller $\widehat{K}(z)$ is given by

$$\widehat{K}(z) = \left(I + K(z) \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} K(z). \tag{3.54}$$

The following are detailed steps of the proof of equation (3.54). From (3.46),

$$\begin{aligned} v(z) = K(z)\begin{bmatrix} d(z) \\ y(z) \end{bmatrix} &= K_1(z)d(z) + K_2(z)y(z) = K_1(z)(u(z) - v(z)) + K_2(z)y(z) \\ &\Downarrow \\ (I + K_1(z))v(z) &= K_1(z)u(z) + K_2(z)y(z) = K(z)\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} \\ &\Downarrow \\ v(z) &= (I + K_1(z))^{-1}K(z)\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} \end{aligned}$$

Since $K_1(z) = \begin{bmatrix} K_1(z) & K_2(z) \end{bmatrix}\begin{bmatrix} I \\ 0 \end{bmatrix} = K(z)\begin{bmatrix} I \\ 0 \end{bmatrix}$, it results that

$$\begin{aligned} v(z) &= \left(I + K(z)\begin{bmatrix} I \\ 0 \end{bmatrix}\right)^{-1} K(z)\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} \\ &\Downarrow \\ v(z) &= \widehat{K}(z)\begin{bmatrix} u(z) \\ y(z) \end{bmatrix} \end{aligned}$$

Also,

$$\begin{aligned} v(k) &= F_1\widehat{x}(k) + F_2 d(k) = F_1\widehat{x}(k) + F_2 u(k) - F_2 v(k) \\ &\Downarrow \\ (I + F_2)v(k) &= F_1\widehat{x}(k) + F_2 u(k) \\ &\Downarrow \\ v(k) &= (I + F_2)^{-1}F_1\widehat{x}(k) + (I + F_2)^{-1}F_2 u(k) \end{aligned}$$

But, $v(k) = F\widehat{x}(k) + Ku(k)$, and therefore $K = (I + F_2)^{-1}F_2$, and $F = (I + F_2)^{-1}F_1$ as in (3.42).

If the closed loop structure in Figure 3.4 is transformed into the LFT in Figure 3.5, standard $\mathcal{H}_\infty$ control could be carried out on $G(z)$. With $d(k)$ as the disturbance input, $v(k)$ as the control input, $[y(k),\, u(k)]^{\mathrm{T}}$ as measurements outputs, and $v(k)$ as the output to be controlled and minimized, a realization for $G(z)$ would be given by

$$G(z) = \left[\begin{array}{c|c:c} A & B & B \\ \hline 0 & 0 & I \\ \hdashline C & 0 & 0 \\ 0 & I & I \end{array}\right]. \tag{3.55}$$

Nevertheless, the above given observer-based solution was given instead of an $\mathcal{H}_\infty$ approach

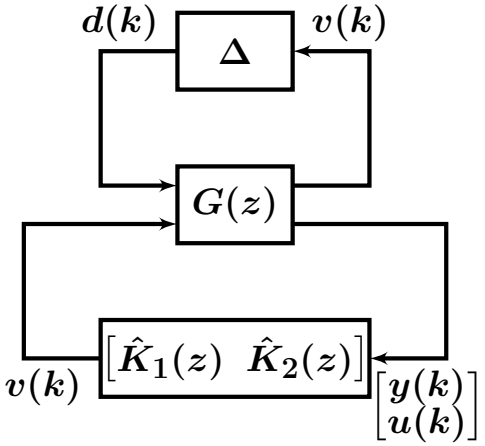


Figure 3.5: Closed loop in LFT form.

in order to avoid unnecessary complications. Also, the $\mathcal{H}_\infty$ control method is applied in the following section, to solve the rather complicated case of an LDTI system affected by quantization at both input and output simultaneously.

3.3 Stabilization with Quantization at Both Input and Output

The next step is to consider the output feedback control for the plant $P(z) = C(zI - A)^{-1}B$ with logarithmic quantization at both input and output. In this setting, the feedback controller $K(z)$ will be driven by either the quantized measurements $y_q(k)$ alone, or by the quantized control inputs $u(k)$ and the quantized measurements $y_q(k)$ simultaneously.

As stated in Theorem 3.1.1 from [16], analyzing stability of an NCS with logarithmic quantizer is equivalent to analyzing the robust stability of a feedback system with sector bounded uncertainties modeled in multiplicative form. Hence, stability analysis and controller synthesis are focused on uncertain dynamic systems with both input and output $\mathcal{H}_\infty$-norm bounded multiplicative uncertainties. The quantization errors are treated as time-varying sector bounded uncertainties, and integrated in the system's model as shown in equation (3.56).

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) = Ax(k) + B[I + \Delta_1(k)]v(k) \\ y_q(k) &= [I + \Delta_2(k)]y(k) = [I + \Delta_2(k)]Cx(k), \end{cases} \tag{3.56}$$

where, $y(k) = Cx(k)$ denotes the plant output prior to quantization. The general LFT form used to model the closed-loop structure is shown in Figure 3.6, with signals $z(k)$, $d(k)$, and $w(k)$ chosen according to the analyzed control system. The typical $\mathcal{H}_\infty$ control

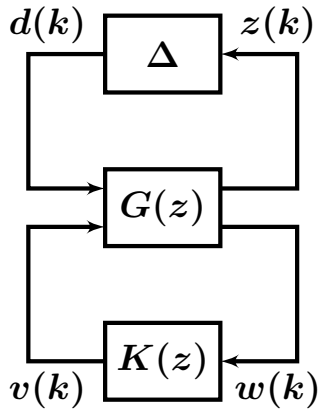


Figure 3.6: Closed loop structure in LFT form.

problem in [51] is employed in order to synthesize the controller that would allow for the largest uncertainty bound. By the sector bound approach, define the uncertainties bounds as $\|\Delta_1\|_\infty \leq \delta_u$, and $\|\Delta_2\|_\infty \leq \delta_y$. The uncertainty block in Figure 3.6 has the diagonal form $\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}$. In order to solve the $\mathcal{H}_\infty$ control and robust stability problems, the μ-synthesis method presented in [51] is applied.

Let $\|T(z)\|_\infty = \sup_{|z|>1} \bar{\sigma}\,[T(z)]$, with $\bar{\sigma}\,[T(z)]$ being the maximum singular value of any proper and real rational stable transfer matrix $T(z)$, and $\|\Delta\|_\infty = \sup_{f \neq 0 \in l_2[0,\infty)} \dfrac{\|\Delta f\|_2}{\|f\|_2}$. Define $\widehat{G}(z) = \mathcal{F}_l(G(z), K(z))$ as the lower LFT with respect to K of the system in Figure 3.6. Since Δ is a time-varying two-block-diagonal uncertainty, to study the robust stability of the closed-loop system in Figure 3.6, the extended version of the small gain theorem in [51], summarized at the end of Chapter 2, is employed. Hence, let the scaling matrices W_l and W_r be chosen as $\begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix}$. Without loss of generality, W_l and W_r are picked as compatibly dimensioned diagonal matrices $\begin{bmatrix} \tau I & 0 \\ 0 & I \end{bmatrix}$, $\tau > 0$. By the small gain theory in [51], the following lemma can be stated.

Lemma 3.3.1. *The closed-loop system is robustly stable for all* $\|\Delta\|_\infty < \dfrac{1}{\gamma}$ *if and only if*

$$\|W_l \widehat{G}(z) W_r^{-1}\|_\infty \leq \gamma. \tag{3.57}$$

A reasonable approach to deal with the controller synthesis problem is to use an algorithm similar to the so-called *D-K iteration* algorithm to iteratively solve

$$\min_{\tau>0} \min_{K} \|W_l \mathcal{F}_l(G(z), K(z)) W_r^{-1}\|_\infty$$

for τ and K. In short, for a fixed τ in the scaling matrices W_l and W_r, the standard $\mathcal{H}_\infty$ optimization problem is solved to obtain $\min_K \|W_l \mathcal{F}_l(G(z), K(z))W_r^{-1}\|_\infty$. Then, for a given stabilizing controller K, the second convex optimization problem $\min_{\tau>0} \|W_l \mathcal{F}_l(G(z), K(z))W_r^{-1}\|_\infty$ is worked out.

3.3.1 Controller Synthesis - Problem Formulation

Consider the system depicted in the block diagram in Figure 3.6 as being the LFT form of the feedback structure to be analyzed. Since the plant $P(z) = C(zI - A)^{-1}B$ is considered to be real-rational and proper, $G(z)$ is also going to be real-rational and proper. Moreover, its state-space realization will be stabilizable and detectable, due to the fact that the pairs (A, B), and (C, A) are assumed to be stabilizable, and detectable, respectively. Clearly, the main issue that needs to be taken care of is stability. The $\mathcal{H}_\infty$ method aims to synthesize a real-rational and proper controller $K(z)$ that would not only stabilize the closed-loop system, but also minimize $\|W_l \mathcal{F}_l(G(z), K(z))W_r^{-1}\|_\infty$. This would result in finding the largest admissible bound γ^{-1} for the uncertainty block $\Delta = \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix}$. Since the time-varying sector bounded uncertainties model the input and output quantization errors, their magnitude cannot exceed 1, and therefore it is assumed that $\gamma^{-1} < 1 (\gamma > 1)$.

3.3.2 Controller Synthesis - Simple Output Feedback Case

The first feedback loop with both input and output quantization modeled as multiplicative uncertainty is depicted in Figure 3.7. The output feedback control law depends solely on the quantized output of the system, and hence expressed as

$$v(k) = K y_q(k). \tag{3.58}$$

To arrange the system in Figure 3.7 in the LFT form in Figure 3.6, for $P(z) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]$, with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times q}$, and $C \in \mathbb{R}^{p\times n}$, denote the signals $z(k) = \begin{bmatrix} v(k) \\ y(k) \end{bmatrix}$,

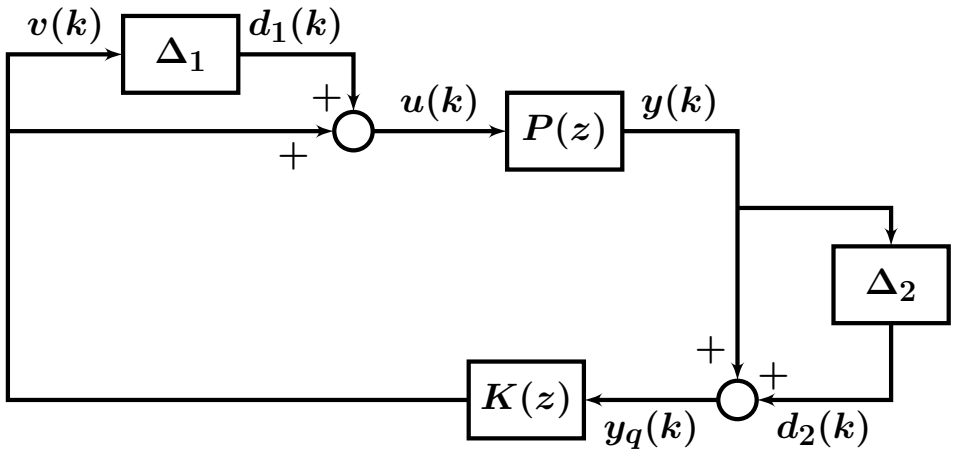


Figure 3.7: Feedback control system involving quantized plant input and output modeled as multiplicative uncertainties.

$w(k) = y_q(k)$, and $d(k) = \begin{bmatrix} d_1(k) \\ d_2(k) \end{bmatrix}$. The LFT for this case is depicted in Figure 3.8, and the equation that describes the dynamics becomes

$$\left[\begin{array}{c} v(k) \\ y(k) \\ \hdashline y_q(k) \end{array}\right] = \left[\begin{array}{cc:c} 0 & 0 & I_q \\ P(z) & 0 & P(z) \\ \hdashline P(z) & I_p & P(z) \end{array}\right] \left[\begin{array}{c} d_1(k) \\ d_2(k) \\ \hdashline v(k) \end{array}\right] = \left[\begin{array}{c|cc:c} A & B & 0 & B \\ \hline 0 & 0 & 0 & I_q \\ C & 0 & 0 & 0 \\ \hdashline C & 0 & I_p & 0 \end{array}\right] \left[\begin{array}{c} d_1(k) \\ d_2(k) \\ \hdashline v(k) \end{array}\right]. \tag{3.59}$$

In this first case, $G(z) = \left[\begin{array}{cc:c} 0 & 0 & I_q \\ P(z) & 0 & P(z) \\ \hdashline P(z) & I_p & P(z) \end{array}\right]$, and therefore the scaling matrices are chosen to be $W_l = \begin{bmatrix} \tau I_q & 0 \\ 0 & I_p \end{bmatrix}$, and $W_r = \begin{bmatrix} \tau I_q & 0 \\ 0 & I_p \end{bmatrix}$, $\tau > 0$. A necessary and sufficient condition for the existence of a stabilizing controller $K(z)$ such that $\|W_l \mathcal{F}_l(G(z), K(z)) W_r^{-1}\|_\infty < \gamma$ for a given $\gamma > 1$ is expressed in Theorem 3.3.1 below.

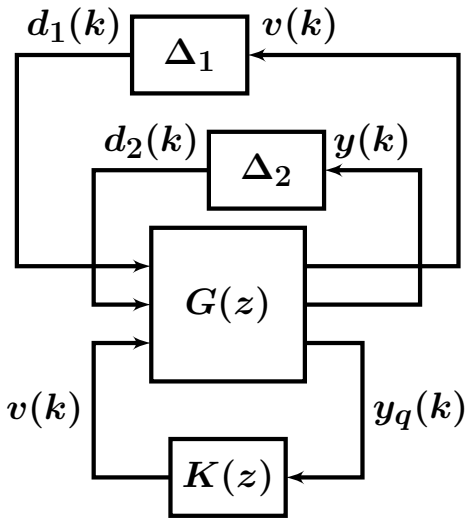


Figure 3.8: LFT for structure in Figure 3.7.

Theorem 3.3.1. *There exist $\tau > 0$ and a stabilizing controller K such that*

$$\|W_l \mathcal{F}_l(G(z), K) W_r^{-1}\|_\infty < \gamma$$

if and only if $\rho(X_\infty Y_\infty) < \gamma^2$, where $\rho(X_\infty Y_\infty)$ is the spectral radius of the product $X_\infty Y_\infty$, and X_∞ and Y_∞ are the solutions to the DAREs below

$$X_\infty = A^* X_\infty \left[I + (1 - \gamma^{-2}) B B^* X_\infty\right]^{-1} A + \tau^{-2} C^* C, \tag{3.60}$$

$$Y_\infty = A Y_\infty \left[I + (1 - \gamma^{-2}) \tau^{-2} C^* C Y_\infty\right]^{-1} A^* + B B^*. \tag{3.61}$$

Proof. Since $\gamma > 1$, the DARE in equations (3.60) and (3.61) will always have positive semi-definite solutions. Therefore, according to the $\mathcal{H}_\infty$ control theory, the only condition that needs to be satisfied to ensure the existence of a stabilizing controller is $\rho(X_\infty Y_\infty) < \gamma^2$. □

3.3.3 Controller Synthesis - Input-Output Feedback Case

The same analysis is performed on the structure in Figure 3.9, which is characterized by an augmented control law that, not only considers the quantized measurements, but

also the quantized input, as seen in equation (3.62).

$$v(k) = K_1 u(k) + K_2 y_q(k) = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} u(k) \\ y_q(k) \end{bmatrix} = K \begin{bmatrix} u(k) \\ y_q(k) \end{bmatrix}. \tag{3.62}$$

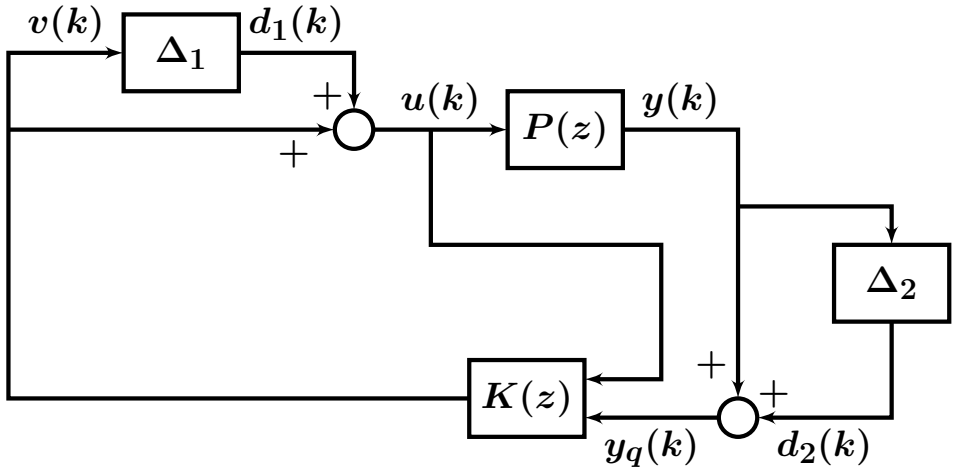


Figure 3.9: Feedback control system involving quantized plant input and output modeled as multiplicative uncertainties, and input-output feedback law.

After rearranging the system in Figure 3.9 to fit the LFT form in Figure 3.6, with $z(k) = \begin{bmatrix} v(k) \\ y(k) \end{bmatrix}$, $w(k) = \begin{bmatrix} u(k) \\ y_q(k) \end{bmatrix}$, and $d(k) = \begin{bmatrix} d_1(k) \\ d_2(k) \end{bmatrix}$, the LFT in Figure 3.10 is obtained and the equation describing its dynamics is

$$\left[\begin{array}{c} v(k) \\ y(k) \\ \hdashline u(k) \\ y_q(k) \end{array}\right] = \left[\begin{array}{cc:c} 0 & 0 & I_q \\ P(z) & 0 & P(z) \\ \hdashline I_q & 0 & I_q \\ P(z) & I_p & P(z) \end{array}\right] \left[\begin{array}{c} d_1(k) \\ d_2(k) \\ \hdashline v(k) \end{array}\right] = \left[\begin{array}{c|c|c:c} A & B & 0 & B \\ \hline 0 & 0 & 0 & I_q \\ \hline C & 0 & 0 & 0 \\ \hdashline 0 & I_q & 0 & I_q \\ \hline C & 0 & I_p & 0 \end{array}\right] \left[\begin{array}{c} d_1(k) \\ d_2(k) \\ \hdashline v(k) \end{array}\right]. \tag{3.63}$$

For this case $G(z) = \left[\begin{array}{cc:c} 0 & 0 & I_q \\ P(z) & 0 & P(z) \\ \hdashline I_q & 0 & I_q \\ P(z) & I_p & P(z) \end{array}\right]$, and thus the scaling matrices become $W_l = \begin{bmatrix} \tau I_q & 0 \\ 0 & I_p \end{bmatrix}$, and $W_r = \begin{bmatrix} \tau I_q & 0 \\ 0 & I_p \end{bmatrix}$, $\tau > 0$. Once again, after transforming the discrete time problem into its continuous time counterpart using a bilinear transformation, the controller synthesis problem is solved as presented in Theorem 3.3.2.

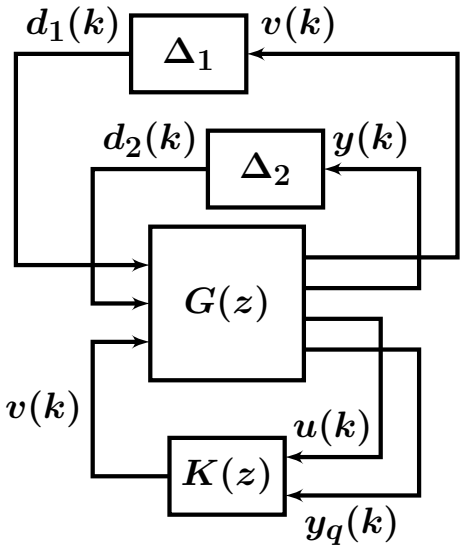


Figure 3.10: LFT for structure in Figure 3.9.

Theorem 3.3.2. *There exist $\tau > 0$ and a stabilizing controller K such that*

$$\|W_l \mathcal{F}_l(G(z), K) W_r^{-1}\|_\infty < \gamma$$

if and only if $\rho(X_\infty Y_\infty) < \gamma^2$, where $\rho(X_\infty Y_\infty)$ is the spectral radius of the product $X_\infty Y_\infty$,

and X_∞ and Y_∞ are the solutions to the DAREs below

$$X_\infty = A^* X_\infty \left[I + (1-\gamma^{-2}) B B^* X_\infty\right]^{-1} A + \tau^{-2} C^* C, \tag{3.64}$$

$$Y_\infty = A Y_\infty \left[I + (1-\gamma^{-2}) \tau^{-2} C^* C Y_\infty\right]^{-1} A^*. \tag{3.65}$$

Proof. The proof follows the same argument as the proof for Theorem 3.3.1. □

Remark 3.3.1. If plant $P(z) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]$ is stable, i.e. all eigenvalues of A are inside the unit circle, then the solution to equation (3.65) is given by $Y_\infty = 0$. Subsequently, the coupling condition becomes $\rho(X_\infty Y_\infty) = 0$. This, in turn, implies that $\gamma \to 0$, which allows for a larger and larger upper bound for the uncertainties. Since the main concern of this study is stability, it can be concluded that if A is a stability matrix, the system can be left to evolve in open-loop.

Denote the solutions to equations (3.60), (3.61), (3.64), and (3.65) as $\widehat{X}_\infty$, $\widehat{Y}_\infty$, $\widetilde{X}_\infty$, and $\widetilde{Y}_\infty$, respectively. According to [26] and [51], since $\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]$ is a minimal realization of $P(z)$, these solutions exist and are hermitian and positive semi-definite. Moreover, since equations (3.60) and (3.64) are the same for both control structures, $\widehat{X}_\infty = \widetilde{X}_\infty \geq 0$.

Moreover, from equations (3.64) and (3.65), it results that $\widehat{Y}_\infty \geq \widetilde{Y}_\infty$. With $\widehat{X}_\infty = \widetilde{X}_\infty \geq 0$, it follows that $\rho(\widehat{X}_\infty \widehat{Y}_\infty) \geq \rho(\widetilde{X}_\infty \widetilde{Y}_\infty)$. The results of the $\mathcal{H}_\infty$ optimization problem in Theorems 3.3.1 and 3.3.2 ensure that, for given $\widehat{\gamma} > 1$ and $\widetilde{\gamma} > 1$, $\rho(\widehat{X}_\infty \widehat{Y}_\infty) < \widehat{\gamma}^2$, and $\rho(\widetilde{X}_\infty \widetilde{Y}_\infty) < \widetilde{\gamma}^2$, respectively. It follows that $\widehat{\gamma} \geq \widetilde{\gamma}$. In conclusion, $\widehat{\gamma}^{-1} \leq \widetilde{\gamma}^{-1}$, which states that the amount of admissible uncertainty in the case of the simple output feedback structure is less than how much uncertainty the second structure could tolerate. This analytical result is in agreement with the intuitive conclusion that the controller in the second case should be able to handle more uncertainties, since it is not only receiving

information about the uncertain output, but also about the uncertain input.

3.3.4 Comparative Simulation Results

This section presents the algorithm used to solve several examples whose results are intended to back up the intuitive and theoretical conclusions drawn in the previous section.

However, though this research focuses on discrete time systems, the controller synthesis theorem is stated for continuous time domain, based on the $\mathcal{H}_\infty$ control theory developed in [51]. The reason is that the final results that are of interest are similar in both continuous and discrete time domains. Indeed, the analyzed upper bounds represent the infinity norm of a closed-loop system, and the infinity norm of a given discrete problem is preserved under a bilinear transformation in the continuous domain, and vice-versa. Thus, using the bilinear transformation, also known as the trapezoidal integration method or Tustin transformation method

$$z = \frac{1 + \frac{sT_s}{2}}{1 - \frac{sT_s}{2}} \Leftrightarrow s = \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}},$$

where T_s is the sampling time, the corresponding Laplace transform is given by

$$G(s) = G(z)\Big|_{z=\frac{1+\frac{sT_s}{2}}{1-\frac{sT_s}{2}}}.$$

It follows that

$$\|W_l \mathcal{F}_l(G(z), K_d) W_r^{-1}\|_\infty = \|W_l \mathcal{F}_l(G(s), K_c) W_r^{-1}\|_\infty,$$

with K_d and K_c being the controllers for discrete, and continuous cases, respectively.

The algorithm that is coded in the MATLAB function in Appendix C performs a two-parameter minimization procedure sequentially. First, with W_l and W_r fixed by choosing a $\tau > 0$, a minimization is carried on over K, then over τ with K being fixed. The process is repeated until a certain desired tolerance is reached. The following steps summarize some of the synthesis algorithm details:

(i) Fix an initial estimate of the scaling matrices W_l and W_r by setting up an estimate

for $\tau > 0$.

(ii) Select an initial set of positive values for τ among which the first search is going to be performed.

(iii) For each τ in the set in step (ii), construct a state-space model for the scaled system

$$\widetilde{W}_l G(z) \widetilde{W}_r^{-1} = \begin{bmatrix} W_l & 0 \\ 0 & I \end{bmatrix} G(z) \begin{bmatrix} W_r^{-1} & 0 \\ 0 & I \end{bmatrix}.$$

(iv) Solve an $\mathcal{H}_\infty$-optimization problem to minimize

$$\|\mathcal{F}_l(\widetilde{W}_l G(z) \widetilde{W}_r^{-1}, K)\|_\infty = \|W_l \mathcal{F}_l(G(z), K) W_r^{-1}\|_\infty$$

over all stabilizing K's, and denote the minimizing controller by $\widehat{K}$.

(v) Find τ that minimizes

$$\|W_l \mathcal{F}_l(G(z), \widehat{K}) W_r^{-1}\|_\infty = \|\mathcal{F}(\widetilde{W}_l G(z) \widetilde{W}_r^{-1}, \widehat{K})\|_\infty,$$

and denote it by $\widehat{\tau}$.

(vi) Compare $\widehat{\tau}$ with the previous estimate τ. The algorithm stops if the two estimates are close, or replaces τ with $\widehat{\tau}$ and goes back to step (ii), otherwise.

To ensure the search for τ is always performed within a set of equidistant points, the algorithm will only consider working with a scaling variable $\varepsilon \in (0, 1)$. Values greater than 1 for τ will be nonetheless taken into consideration based on the following remarks.

1. For $\tau = \varepsilon \in (0, 1)$, the scaling matrices can be rewritten in terms of ε without any

major differences.

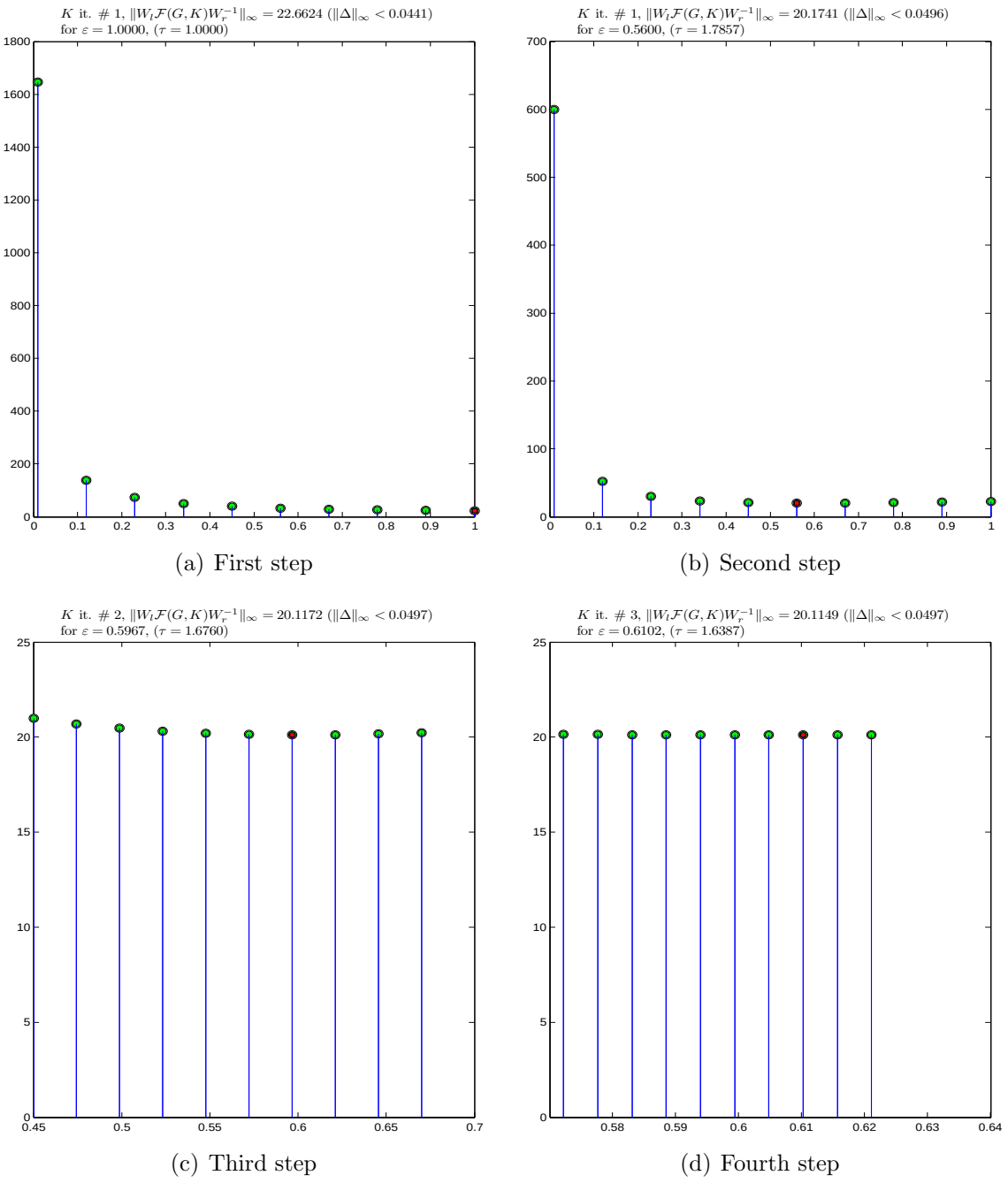
$$W_l = \begin{bmatrix} \tau I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \varepsilon I & 0 \\ 0 & I \end{bmatrix},$$
$$W_r^{-1} = \begin{bmatrix} \tau^{-1} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \varepsilon^{-1} I & 0 \\ 0 & I \end{bmatrix}.$$

2. For $\tau > 1$, we can rewrite $\tau = \dfrac{1}{\varepsilon} = \varepsilon^{-1}$. The scaling matrices can thus be rewritten in terms of ε as follows

$$W_l = \begin{bmatrix} \tau I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \varepsilon^{-1} I & 0 \\ 0 & I \end{bmatrix},$$
$$W_r^{-1} = \begin{bmatrix} \tau^{-1} I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \varepsilon I & 0 \\ 0 & I \end{bmatrix}.$$

The algorithm procedure is exemplified in the set of 4 graphs in Figure 3.11, for the case of the unstable plant $P(z) = \dfrac{z-3}{z(z-2)}$. The search for $\tau > 0$ that would minimize the norm $\|W_l \mathcal{F}_l(G(z), \widehat{K}) W_r^{-1}\|_\infty$ starts in the interval (0, 1], as seen in Figure 3.11(a). Since the minimum norm is found to be for $\tau = 1$, the search continues for values $\tau > 1$, but still inside the (0, 1] interval through the change of variable $\tau = \varepsilon^{-1} > 1$, as depicted in Figures 3.11(b), 3.11(c), and 3.11(d).

After having run the algorithm on several cases characterized by different plant models, the results were compiled in Table 3.1. Once again, it can be seen by the examples covering the unstable cases, that the upper limits for the uncertainty gain are greater in the case when extra information is added to the feedback control law (column 5) than when controller is synthesized based solely on output measurements (column 3). Regarding the cases when the plant is stable, illustrated in the second part of Table 3.1, the conclusions



(a) First step

(b) Second step

(c) Third step

(d) Fourth step

Figure 3.11: Detailed steps of the search algorithm for a plant $P(z) = \dfrac{z-3}{z(z-2)}$ controlled using the simple output feedback structure.

are similar to those in Remark 3.3.1. When the plant is stable, there is no difference in the maximum amount of admissible uncertainty between the two studied structures. Moreover, the upper bounds for the uncertainty gains in the case of stable plants are considerably

larger than the ones obtained for the unstable plants, and correspond to the same value of $\tau > 0$.

Table 3.1: Comparative simulation results for the uncertainty upper bounds.

$P(z)$	simple output feedback		input-output feedback	
	τ	$\widehat{\gamma}^{-1}$	τ	$\widetilde{\gamma}^{-1}$
$\frac{1}{z-2}$	0.5478	0.2588	0.01	0.4989
$\frac{1}{z(z-2)}$	0.6415	0.1209	0.01	0.2499
$\frac{z-3}{z(z-2)}$	1.6387	0.0497	0.01	0.1
$\frac{1}{z(z-1.4)}$	1.187	0.2497	0.01	0.5089
$\frac{z-2}{z(z-1.6)}$	1.399	0.0566	0.01	0.1136
$\frac{z-1.5}{z(z-1.7)}$	0.8248	0.0379	0.01	0.0759
$\frac{1}{z-1.1}$	1.7476	0.5395	0.01	0.9078
$\frac{1}{z-0.8}$	100	20.0005	100	20.0236
$\frac{z-0.8}{z(z-0.3)}$	100	72.2223	100	72.2291
$\frac{z-0.3}{z(z-0.1)}$	100	84.6154	100	84.6213

To back up these results, another experiment has been conducted. A MATLAB® Simulink model shown in Figure 3.12 was set up to include both feedback structures in Figures 3.7 and 3.9. The purpose was to monitor the time-domain behavior of the output signal when random time-variant uncertainties were applied at both input and output. The unforced time responses of the closed-loop systems with the plant model given by $P(z) = \dfrac{z-3}{z(z-2)}$ is studied in 3 different scenarios. The first case uses randomly generated uncertainties in the interval $(-\widetilde{\gamma}^{-1}, \widetilde{\gamma}^{-1}) = (-0.1, 0.1)$. It can be seen from Figure 3.13 that the outputs are stable in both cases, despite the fact that the uncertainties exceed

the limits for the closed-loop structure with simple feedback. This may be due to the fact that, once the uncertainties fall in the interval $(-\widehat{\gamma}^{-1}, \widehat{\gamma}^{-1}) = (-0.05, 0.05)$, the controller manages to restore the stable behavior. Unlike this first scenario, the next two consider the uncertainties to be randomly generated outside the bound for the first structure, but still within the bound of the second feedback structure. The results in Figure 3.14 are for positive values of uncertainties, while the ones in Figure 3.15 deal with negative uncertainties. As expected, since the uncertainty gain is constantly outside the stability range for the first

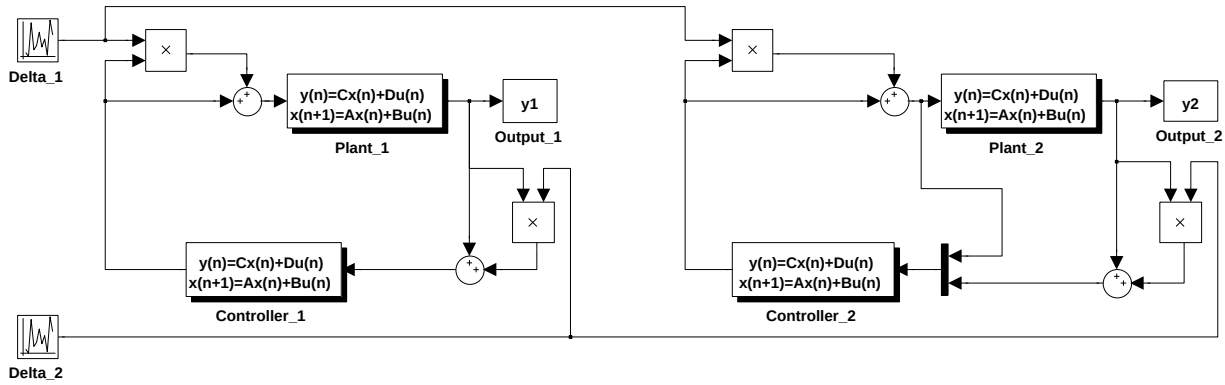


Figure 3.12: MATLAB®/Simulink diagram to simulate system's behavior with input and output uncertainties.

structure, the designed controller will not be able to sustain the closed-loop stability. On the contrary, the controller designed for the second structure manages to bring the output signal to a steady-state.

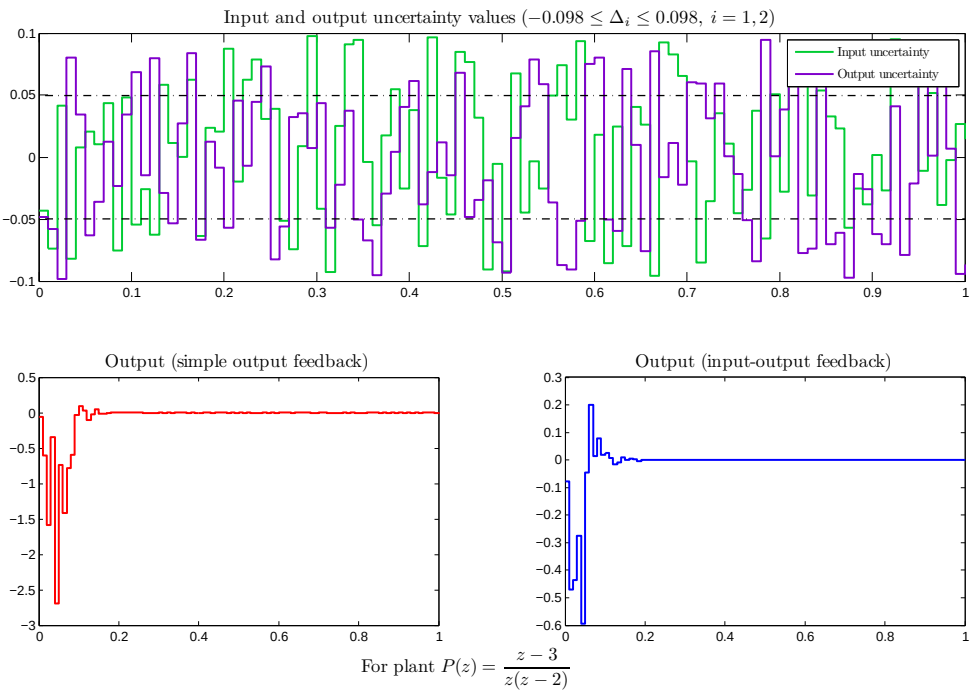


Figure 3.13: Time responses for the uncertain structures in Figures 3.7 and 3.9 for plant given by $P(z) = \dfrac{z-3}{z(z-2)}$ and uncertainties in $(-0.1,\, 0.1)$.

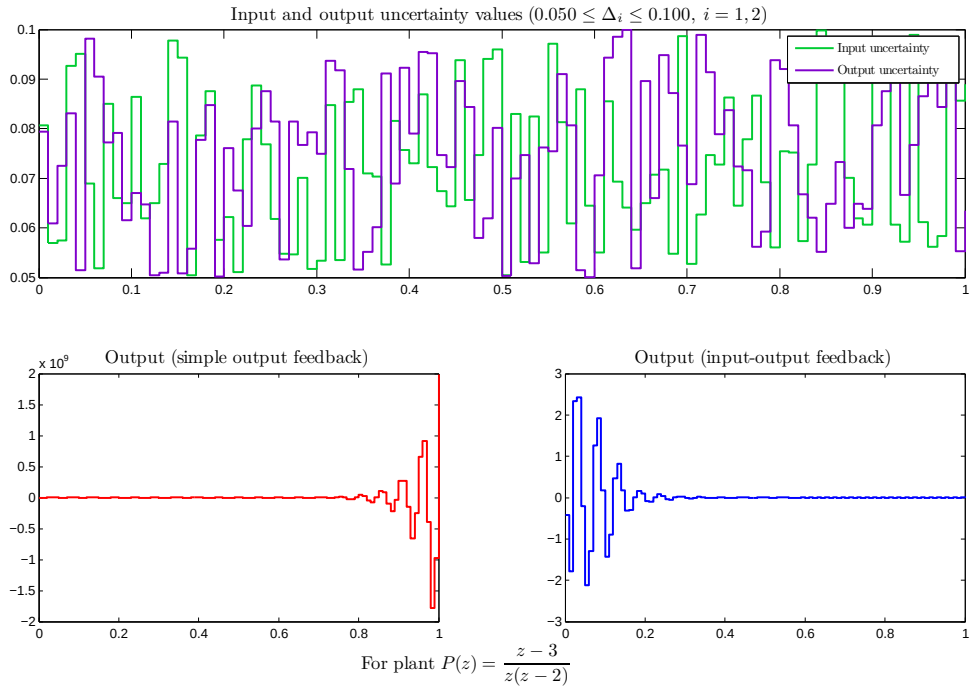


Figure 3.14: Time responses for the uncertain structures in Figures 3.7 and 3.9 for plant given by $P(z) = \dfrac{z-3}{z(z-2)}$ and uncertainties in $(0.05,\, 0.1)$.

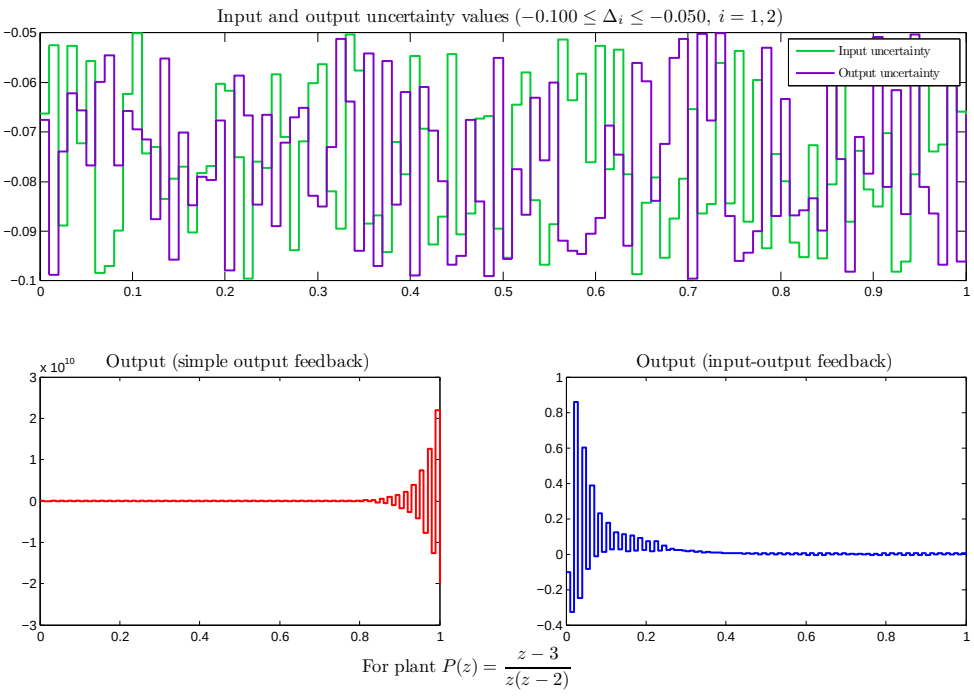


Figure 3.15: Time responses for the uncertain structures in Figures 3.7 and 3.9 for plant given by $P(z) = \dfrac{z-3}{z(z-2)}$ and uncertainties in $(-0.1,\ -0.05)$.

Chapter 4
Consensusability of Multi-agent Systems

The discussions in Chapter 3 dealt with issues related to communication-constrained feedback channels in networked control systems. Another networked control problem that has received a considerable amount of attention among many research communities considers the distributed coordination of a set of autonomous or semiautonomous agents, such as unmanned air vehicles (UAVs), unmanned ground vehicles (UGVs), or unmanned underwater vehicles (UUVs). Applications of such problem include formation control, flocking, and synchronization of chaotic oscillators. Though it is still important for the control information flow to run through the communication channels at a fast rate, the patterns of this information flow matter most for these systems and applications. The agents of this kind of systems are networked together through feedback of local information coming from neighboring agents. They all perform individual tasks while interacting with each other with the intention of reaching a prescribed consensus asymptotically.

The main contribution of this chapter is to address the consensus problems for a set of discrete-time agents with directed network information flow. The information exchange among agents is viewed as a communication graph and, therefore, modeled using algebraic graph theory, to describe how much information each agent has access to at a specific moment in time. Directed graphs prove to be very helpful with modeling network characteristics, like "agent i receives information from agent j", for example. Moreover, the graph Laplace matrix offers a way to quantify the agents performance to reach an agreement.

4.1 Multi-agent Systems: Overview

It has been more than a decade since the term Multi-agent System (MAS) has been introduced in the computer science community. It didn't take long for the concept to become popular in other fields of research and industry. MAS technology is being built today

to help in the fields of power systems ([31]) with diagnostics, condition monitoring, powers system restoration, market simulation, network control and automation, or in control area, to take care of sensor networks, distributed computation, cooperative control of unmanned vehicles, attitude alignment of satellite clusters, to name just a few. For all these and many other applications, an MAS would allow to have a main problem divided into subproblems that would be distributed to distinct problem solving agents designed to have their own interests and goals. Though the necessity to research the field of MASs is easy to understand, their advantages, the novelties they are bringing, and the problems they are more suitable for, are still issues that make MAS technology a well debated topic among researchers and industrial partners.

As mentioned in [41], starting as a subfield of artificial intelligence, MASs attempt to provide the necessary background to construct complex systems that include multiple agents and the mechanisms that would help with synchronizing their behaviors. Thus, there is no surprise to see that most of the MAS terminology, concepts, methodologies are defined using terms from artificial intelligence and computer science fields. The terminology and definitions to follow are mainly taken from [45] and [31].

4.1.1 Agent, Intelligent Agent, and Multi-agent System

According to Wooldridge in [45], "an *agent* is a computer system that is *situated* in some *environment*, and that is capable of *autonomous action* in this environment in order to meet its design objectives." The environment represents the external surroundings of the agent. To perform its tasks, an agent might have to observe the environment, which would be usually done through sensors in the case of physical environment (e.g. industrial processes), or through messaging and program calling in the case of computing environment (e.g. data and computing resources). In the industrial processes field, for instance, a *control system* can be considered an agent. Thermostats, for example, can be viewed as an agent that automatically responds to the temperature changes in the surrounding space. The temperature is sensed using sensors, and switches are activated to control the equipment.

Of course, control systems viewed as agents can be more elaborate, as in the case of independent space probes, unmanned aircrafts, nuclear reactor control systems, etc.

Nevertheless, as noted in [31], from an engineering point of view, this definition is debatable, because it is not quite clear what differentiates agents from the existing hardware and software systems. A more detailed description of the agent notion was needed to ensure the line between agent systems approaches and current engineering approaches is not easy to cross. Therefore, Wooldridge revised the concept of *agent* and turned it into the new concept of an *intelligent agent*, by adding three extra features that will individualize agents from standard hardware and software systems. These characteristics are:

- Reactivity. Any change in the surrounding environment is promptly perceived by an intelligent agent, and determines it to immediately take action towards fulfilling the goals it has been designed for.

- Pro-activeness. An intelligent agent behaves constantly guided by the goals it needs to achieve. In the words of Wooldridge, it presents a "goal-directed behavior by taking the initiative", meaning that it continuously changes its actions in order to meet the requirements.

- Social ability. An intelligent agent has the ability to interact with other intelligent agents coexisting in the same environment to better reach the purpose for which it has been developed. The interaction between intelligent agents implies more than just simple data transfer from one to another, which most software and hardware systems already do. Intelligent agents negotiate actions and work in a collaborative manner, so that they reach a consensus that would allow all of them to achieve their objectives.

It is now easy to define a Multi-agent System (MAS) as a system consisting of two or more agents or intelligent agents. What is really critical to acknowledge is that while each agent of an MAS has a very well defined goal, there is no global system objective. The intention

for which the MAS has been designed can only be achieved by involving several intelligent agents, each developed to achieve their own local goals corresponding to component parts of that intention. The agent receives messages signaling the designer's intentions and the other agents' actions. Its autonomy allows it to decide upon fulfilling the requests, the task priority, and what further actions need to be scheduled if necessary.

Since there is hardly a unanimous decision on a general definition of agents, it is left to the designer's latitude to choose whether or not the agents in the MAS have the ability to communicate, depending on the specific field of interest. For the sake of this research and its applications, the MASs taken into account support communication between agents, that is used with the main intention of reaching a consensus among the agents.

4.1.2 Consensus of Multi-agent Networked Systems

Once again, with their developments in distributed computing, the computer science community brought up an interesting subject: *consensus* in a distributed network [29]. In general, by consensus it should be understood that the agents within a network asymptotically come to an accord with respect to a specific entity of interest that depends the initial states of all agents [34].

In the field of systems control theory, distributed computation and systems have been playing an important role. Thus, a lot of consensus problems arose throughout the years. For instance, the systems in charge of watching the altimeters on board of an aircraft need to eventually reach an agreement about the altitude value. Or, some agents might run distinct fault diagnosis procedures on one of the systems components, and the individual conclusions could be later combined into a common decision regarding the state of that particular component.

4.2 Graph Theory

4.2.1 Directed Graphs

The communication between the N agents of a multi-agent system can be represented by a *directed graph* (or *digraph*) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is a finite non-empty set of *nodes*, and $\mathcal{E} = \{e_1, e_2, \ldots, e_M\} \subset \mathcal{V} \times \mathcal{V}$ is a set of ordered pairs of distinct nodes called *directed edges* or *arcs*. An arc $(v_i,\, v_j) \in \mathcal{E}$ of a digraph is simply denoted by $\overrightarrow{v_i v_j}$, and is said to *go out of* v_i and *go into* v_j, or plainly to *go from* v_i *to* v_j. Matrix $\mathcal{A} = [\mathsf{a}_{i,j}]_{i,j=1,1}^{N,N}$ of non-negative elements is called the *weighted adjacency matrix* and consists of values representing the coupling strength between neighboring agents or information received by agent j from agent i. The weights in matrix $\mathcal{A}$ satisfy the following properties:

- $\mathsf{a}_{i,j} > 0$ if $(e_i,\, e_j) \in \mathcal{E}$;
- $\mathsf{a}_{i,i} = 0$, that is self-edges are not allowed or $(e_i,\, e_i) \notin \mathcal{E}$.

The neighborhood of a node v_i is the set of all the nodes v_j, $j \neq i$ directly connected to it, i.e. $\mathcal{N}_i = \{j \,|\, v_i v_j \in \mathcal{E},\, i,\, j = 1, \ldots, N\}$. A $v_1 - v_K$ *walk* connects nodes v_1 and v_K through a sequence of nodes and edges $v_1,\, e_1,\, v_2,\, e_2,\, \ldots,\, e_{K-1},\, v_K$, in which each edge $e_{j-1} = v_{j-1} v_j$, $j = 2, \ldots, K$. A walk in which no node is repeated is called a *path*. A communication digraph $\mathcal{G}$ is then said to be *strongly connected* if there exists a path connecting any pair $(v_i,\, v_j)$ of nodes. For a graph to be *complete*, each node has to be adjacent to all of the others. That is, v_i is directly connected to v_j for all $i \neq j$. A digraph contains a *directed spanning tree*, if one of its nodes v_i has the property of being the *root*, i.e. for each node v_j different than v_i, there is a unique directed path from v_i to v_j. In the case of a weighted directed graph, the *out-degree* of a node $v_i \in \mathcal{V}$ is defined as the sum of the weights of all the arcs radiating from it, according to

$$deg_i^+ = \sum_{j=1}^{N} \mathsf{a}_{i,j},\; i = 1, \ldots, N.$$

On the other hand, its *in-degree* represents the sum of the weights of all arcs going into it,

$$deg_i^- = \sum_{j=1}^{N} \mathsf{a}_{j,i},\ i = 1, \ldots, N.$$

Let $\mathcal{D} = diag(deg_i^+ : v_i \in \mathcal{V})$ be the *diagonal matrix* of graph $\mathcal{G}$. Then, the difference

$$\mathcal{L}_{\mathcal{G}} = [l_{i,j}]_{i,j=1,1}^{N,N} = \mathcal{D} - \mathcal{A} = \begin{bmatrix} \sum_{j=1}^{N} \mathsf{a}_{1,j} & -\mathsf{a}_{1,2} & -\mathsf{a}_{1,3} & \ldots & -\mathsf{a}_{1,N} \\ -\mathsf{a}_{2,1} & \sum_{j=1}^{N} \mathsf{a}_{2,j} & -\mathsf{a}_{2,3} & \ldots & -\mathsf{a}_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathsf{a}_{N,1} & -\mathsf{a}_{N,2} & -\mathsf{a}_{N,3} & \ldots & \sum_{j=1}^{N} \mathsf{a}_{N,j} \end{bmatrix} \tag{4.1}$$

is the *Laplace matrix*, or *Laplacian*, of communication graph $\mathcal{G}$. Thus, $l_{i,i} > 0$, $l_{i,j} \leq 0$, $\forall i \neq j$, and $\sum_{j=1}^{N} l_{i,j} = 0$ for each i.

Lemma 4.2.1. *[37], [17], [7] The Laplacian $\mathcal{L}_{\mathcal{G}}$ of a digraph $\mathcal{G}$ has at least one zero eigenvalue and all the non-zero eigenvalues are in the open right half plane, and can be arranged in non-decreasing order of their magnitude $0 \leq |\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_N|$. Furthermore, $\lambda_1 = 0$ is a simple eigenvalue, if and only if the digraph $\mathcal{G}$ has a spanning tree.*

Lemma 4.2.2. *[28], [46], [9], [7] If $\mathcal{G}$ has a directed spanning tree, then there exists a permutation matrix $\mathcal{P}$ that would transform $\mathcal{L}$ into its Frobenius normal form*

$$\mathcal{P}\mathcal{L}\mathcal{P}^T = \begin{bmatrix} \mathcal{L}_1 & \mathcal{L}_{12} & \cdots & \mathcal{L}_{1k} \\ 0 & \mathcal{L}_2 & \cdots & \mathcal{L}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{L}_k \end{bmatrix},$$

where $\mathcal{L}_i$, $i = 1, \ldots, N$ are square irreducible matrices, that is their corresponding digraphs are strongly connected. Hence, decomposing the Laplace matrix into its Frobenius normal form is equivalent to decomposing the digraph into its maximally strongly connected

subgraphs.

4.2.2 The Laplace Matrix Spectrum

Consider a digraph $\mathcal{G}$ that has a spanning tree, and let the non-zero eigenvalues of its Laplacian matrix be $\lambda_i = r_i e^{j\theta_i}$, with $r_i = |\lambda_i|$, $\theta_i = \angle\lambda_i$ for $i = 2, \ldots, N$. Write $\lambda_i = (1 + \delta_i)c$, for some $c \in \mathbb{R}$, and $|\delta_i| \leq \delta$, for $i = 2, \ldots, N$ and some $\delta > 0$. Then,

$$\delta_i = \frac{\lambda_i}{c} - 1 \Leftrightarrow |\delta_i| = \left|\frac{\lambda_i}{c} - 1\right| \leq \delta. \tag{4.2}$$

If $\{\lambda_i\}_{i=2}^N$ are densely populated in an area, then the radius can be calculated according to

$$\delta = \min_{c\in\mathbb{R}} \max_{2\leq i\leq N} \left|\frac{\lambda_i}{c} - 1\right| = \min_{c\in\mathbb{R}} \max_{2\leq i\leq N} \left|\frac{\lambda_i - c}{c}\right|. \tag{4.3}$$

Assume they are compact in a fan shaped region, like the shaded area in Figure 4.1, with $r_2 \leq r_i \leq r_N$ and $-\theta \leq \theta_i \leq \theta$. Without loss of generality, assume $\theta_2 = \theta_N = \theta$ by the dense distribution of the eigenvalues on the fan shaped region.

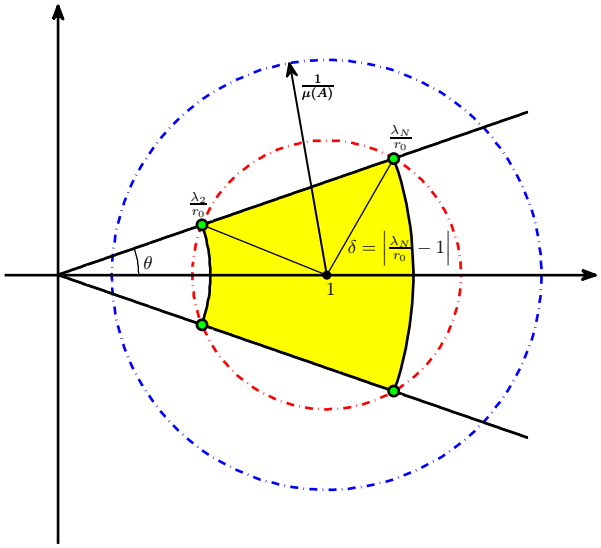


Figure 4.1: Laplace matrix spectrum region.

Then, the center of the set of $\{\lambda_i\}_{i=2}^{N}$, denoted by r_0, satisfies

$$\delta = \left|\frac{\lambda_2}{r_0} - 1\right| = \left|\frac{\lambda_N}{r_0} - 1\right|.$$

Hence, the next set of equations gives a solution for the center r_0 in terms of the largest and the second smallest eigenvalue magnitudes:

$$\begin{aligned}
\delta = \left|\frac{r_2 e^{j\theta}}{r_0} - 1\right| &= \left|\frac{r_N e^{j\theta}}{r_0} - 1\right| \Leftrightarrow \\
\left|r_2 e^{j\theta} - r_0\right| &= \left|r_N e^{j\theta} - r_0\right| \Leftrightarrow \\
\left|r_2(\cos\theta + j\sin\theta) - r_0\right|^2 &= \left|r_N(\cos\theta + j\sin\theta) - r_0\right|^2 \Leftrightarrow \\
(r_2\cos\theta - r_0)^2 + r_2^2\sin^2\theta &= (r_N\cos\theta - r_0)^2 + r_N^2\sin^2\theta \Leftrightarrow \\
r_2^2 - 2r_2 r_0\cos\theta &= r_N^2 - 2r_N r_0\cos\theta \Leftrightarrow \\
r_0 &= \frac{r_N + r_2}{2\cos\theta} = \frac{|\lambda_N| + |\lambda_2|}{2\cos\theta}.
\end{aligned} \tag{4.4}$$

Thus, the radius of the set becomes

$$\begin{aligned}
\delta &= \left|\frac{2r_N e^{j\theta}\cos\theta}{r_N + r_2} - 1\right| = \left|\frac{2r_N e^{j\theta}\cos\theta - (r_N + r_2)}{r_N + r_2}\right| \\
&= \left|\frac{2r_N\cos\theta - (r_N + r_2)e^{-j\theta}}{r_N + r_2}\right| = \left|\frac{2r_N\cos\theta - (r_N + r_2)\cos\theta + j(r_N + r_2)\sin\theta}{r_N + r_2}\right| \\
&= \left|\frac{(r_N - r_2)\cos\theta + j(r_N + r_2)\sin\theta}{r_N + r_2}\right| = \left|\frac{r_N - r_2}{r_N + r_2}\cos\theta + j\sin\theta\right| \\
&= \sqrt{\left(\frac{r_N - r_2}{r_N + r_2}\right)^2\cos^2\theta + \sin^2\theta}.
\end{aligned} \tag{4.5}$$

Since $0 < \dfrac{r_N - r_2}{r_N + r_2} < 1$, from equation (4.5), the following inequality can be written:

$$\begin{aligned}
\delta = \sqrt{\left(\frac{r_N - r_2}{r_N + r_2}\right)^2\cos^2\theta + \sin^2\theta} &> \sqrt{\left(\frac{r_N - r_2}{r_N + r_2}\right)^2\cos^2\theta + \left(\frac{r_N - r_2}{r_N + r_2}\right)^2\sin^2\theta} \\
&= \frac{r_N - r_2}{r_N + r_2} = \frac{|\lambda_N| - |\lambda_2|}{|\lambda_N| + |\lambda_2|}.
\end{aligned} \tag{4.6}$$

If the digraph $\mathcal{G}$, whose Laplace matrix is given by $\mathcal{L}$, models the communication network topology of an MAS, finding the fan-shaped area for the Laplace matrix spectrum, which would ensure agents consensus, becomes a simultaneous gain and phase margin problem.

4.3 Consensus Control of Discrete-time Multi-agent Systems

Consider a set of N discrete-time dynamic agents, each described by the same state-space model

$$\begin{cases} x_i(k+1) &= Ax_i(k) + Bu_i(k), \\ y_i(k) &= Cx_i(k),\ i = 1,\ 2,\ \ldots,\ N, \end{cases} \tag{4.7}$$

where $x_i(k) \in \mathbb{R}^n$ is the agent's state vector, $u_i(k) \in \mathbb{R}^m$ is the control input, and $y_i(k) \in \mathbb{R}^p$ is the measured output. Thus all agents in the system have the same plant model given by the transfer matrix $P(z) = C(zI - A)^{-1}B$. The communication network topology is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. Consensus control of such an MAS intends to design a distributed feedback control protocol based on local information, such that

$$\lim_{k\to\infty} \|x_i(k) - x_j(k)\| = 0,\ \forall i, j = 1, \ldots, N. \tag{4.8}$$

So far, research has been focused on the state feedback protocol ($C = I$) given by

$$u_i(k) = K \sum_{j=1}^{N} \mathsf{a}_{i,j}[x_j(k) - x_i(k)],\ i = 1, \ldots, N, \tag{4.9}$$

where $K \in \mathbb{R}^{m\times n}$ is a constant state feedback gain, and $\mathcal{A} = [\mathsf{a}_{i,j}]_{i,j=1,1}^{N,N}$ represents the adjacency matrix of the digraph $\mathcal{G}$ describing the network communication topology. Consensusability is reached for the multi-agent system, if there exists a control protocol as in equation (4.9) that will lead to (4.8).

Lemma 4.3.1. *[28], [30], [49] In the case of state feedback control, the discrete-time agents modeled by (4.7) are consensusable under the protocol (4.9), if and only if there exists*

a constant control gain $K \in \mathbb{R}^{m\times n}$ *such that* $A - \lambda_i BK$ *is a stability matrix for any* $i = 2, 3, \ldots, N$.

Proof. Using the control protocol in equation (4.9), the state equation in (4.7) for each agent can be rewritten as

$$x_i(k+1) = Ax_i(k) + BK\sum_{j=1}^{N} \mathsf{a}_{i,j}[x_j(k) - x_i(k)] = Ax_i(k) - \sum_{j=1}^{N} l_{i,j}BKx_j(k). \tag{4.10}$$

Stacking all the agents' states in one vector $X(k) = \begin{bmatrix} x_1^{\mathrm{T}}(k) & x_2^{\mathrm{T}}(k) & \ldots & x_N^{\mathrm{T}}(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{nN\times 1}$, the overall state dynamics is given by

$$\begin{aligned} X(k+1) = &\begin{bmatrix} A & 0 & \ldots & 0 \\ 0 & A & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A \end{bmatrix} X(k) \\ &+ \begin{bmatrix} \left(\sum_{j=1}^{N} \mathsf{a}_{1,j}\right) BK & -\mathsf{a}_{1,2}BK & \ldots & -\mathsf{a}_{1,N}BK \\ -\mathsf{a}_{2,1}BK & \left(\sum_{j=1}^{N} \mathsf{a}_{2,j}\right) BK & \ldots & -\mathsf{a}_{1,N}BK \\ \vdots & \vdots & \ddots & \vdots \\ -\mathsf{a}_{N,1}BK & -\mathsf{a}_{N,2}BK & \ldots & \left(\sum_{j=1}^{N} \mathsf{a}_{N,j}\right) BK \end{bmatrix} X(k). \end{aligned}$$

Invoking the definition of Kronecker product, the above equation can be expressed as

$$X(k+1) = \left[I_N \otimes A - \mathcal{L} \otimes BK\right] X(K), \tag{4.11}$$

where $\mathcal{L} \in \mathbb{R}^{N\times N}$ is the Laplacian matrix of the digraph $\mathcal{G}$. It is said that protocol (4.9) solves the consensus problem if the states of system (4.10) satisfy

$$x_i(k) \overset{k\to\infty}{\longrightarrow} x_j(k),\ \forall i, j = 1, 2, \ldots, N. \tag{4.12}$$

Consider $r^{\mathrm{T}} = \begin{bmatrix} r_1 & r_2 & \dots & r_N \end{bmatrix} \in \mathbb{R}^{1\times N}$ to be the left eigenvector of the Laplacian matrix $\mathcal{L}$ associated with the zero eigenvalue ($r^{\mathrm{T}}\mathcal{L} = \mathbf{0}$), satisfying $r^{\mathrm{T}}\mathbf{1} = 1$. Following the rationale in [28], define the *disagreement vector* as

$$\varpi(k) = X(k) - \left[\left(\mathbf{1}r^{\mathrm{T}}\right) \otimes I_n\right] X(k) \in \mathbb{R}^{nN\times 1}, \tag{4.13}$$

satisfying

$$\begin{aligned}\left(r^{\mathrm{T}} \otimes I_n\right) \varpi(k) &= \left(r^{\mathrm{T}} \otimes I_n\right) X(k) - \left(r^{\mathrm{T}} \otimes I_n\right) \left[\left(\mathbf{1}r^{\mathrm{T}}\right) \otimes I_n\right] X(k) \\ &= \left(r^{\mathrm{T}} \otimes I_n\right) X(k) - \left(r^{\mathrm{T}}\mathbf{1}r^{\mathrm{T}} \otimes I_n\right) X(k) \\ &= 0. \end{aligned} \tag{4.14}$$

Based on the dynamics in equation (4.11), the disagreement dynamics becomes

$$\varpi(k+1) = \left(I_N \otimes A - \mathcal{L} \otimes BK\right) X(k) - \left[\left(\mathbf{1}r^{\mathrm{T}}\right) \otimes I_n\right] \left(I_N \otimes A - \mathcal{L} \otimes BK\right) X(k),$$

which, using equations (D.1) and (D.2) in Appendix D, can be rearranged as below:

$$\begin{aligned}\varpi(k+1) &= \left(I_N \otimes A - \mathcal{L} \otimes BK\right) X(k) - \left(I_N \otimes A - \mathcal{L} \otimes BK\right) \left[\left(\mathbf{1}r^{\mathrm{T}}\right) \otimes I_n\right] X(k) \\ &= \left(I_N \otimes A - \mathcal{L} \otimes BK\right) \left[X(k) - \left[\left(\mathbf{1}r^{\mathrm{T}}\right) \otimes I_n\right] X(k)\right] \\ &= \left(I_N \otimes A - \mathcal{L} \otimes BK\right) \varpi(k). \end{aligned} \tag{4.15}$$

In the following paragraphs, it is argued that solving the agents consensusability problem is equivalent to solving the asymptotic stability problem for the disagreement dynamics.

From equation (4.13), the disagreement dynamics is rewritten as

$$\begin{aligned}\varpi(k) &= \left[I_N \otimes I_n - \left(\mathbf{1}r^{\mathrm{T}}\right) \otimes I_n\right] X(k) = \left[\left(I_N - \mathbf{1}r^{\mathrm{T}}\right) \otimes I_n\right] X(k) \\ &= \left(\widehat{M} \otimes I_n\right) X(k), \end{aligned} \tag{4.16}$$

with

$$\widehat{M} = I_N - \mathbf{1}r^{\mathrm{T}} = \begin{bmatrix} 1 - r_1 & -r_2 & \dots & -r_N \\ -r_1 & 1 - r_2 & \dots & -r_N \\ \vdots & \vdots & \ddots & \vdots \\ -r_1 & -r_2 & \dots & 1 - r_N \end{bmatrix}.$$

Given the definition of the left eigenvector r for the Laplacian matrix, $\widehat{M}$ can be diagonalised using the transformation matrix $\widehat{M}_T = \begin{bmatrix} 1 & 0 & \dots & -1 \\ 0 & 1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$ as shown below

$$\widehat{M}_T \widehat{M} \widehat{M}_T^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ -r_1 & -r_2 & \dots & -r_{N-1} & 0 \end{bmatrix}.$$

Thus, it is easy to see that 0 is a simple eigenvalue for $\widehat{M}$ and 1 is an eigenvalue of multiplicity $N - 1$. Moreover, $\mathbf{1}$ is the right eigenvector corresponding to the 0 eigenvalue. Therefore, according to equation (4.16), $\varpi(k) = 0$ if and only if $x_1(k) = x_2(k) = \cdots = x_N(k)$. In other words, the consensus problem is solved if and only if $\varpi(k) \overset{k\to\infty}{\longrightarrow} 0$.

Next step is to review the asymptotic stability of the disagreement dynamics in (4.15).

First, define matrices $\mathcal{Y} \in \mathbb{R}^{N\times(N-1)}$, $\mathcal{W} \in \mathbb{R}^{(N-1)\times N}$, $\mathcal{T} \in \mathbb{R}^{N\times N}$, and $\mathcal{H} \in \mathbb{R}^{(N-1)\times(N-1)}$, satisfying the following properties

$$\mathcal{T} = \begin{bmatrix} \mathbf{1} & \mathcal{Y} \end{bmatrix}, \; \mathcal{T}^{-1} = \begin{bmatrix} r^{\mathrm{T}} \\ \mathcal{W} \end{bmatrix},$$
$$\mathcal{T}^{-1}\mathcal{L}\mathcal{T} = \mathcal{J} = \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{H} \end{bmatrix}, \; \mathcal{H} = \begin{bmatrix} \lambda_2 & * & * & \dots & * \\ 0 & \lambda_3 & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_N \end{bmatrix}, \tag{4.17}$$

where $\lambda_i, \; i = 2, \dots, N$ are the non-zero eigenvalues of the Laplacian matrix L. State transformation $\varepsilon(k) = (\mathcal{T}^{-1} \otimes I_n)\, \varpi(k)$, with $\varepsilon(k) = \begin{bmatrix} \varepsilon_1^{\mathrm{T}} & \varepsilon_2^{\mathrm{T}} & \dots & \varepsilon_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, leads to the new dynamics

$$\varepsilon(k+1) = (I_n \otimes A - \mathcal{J} \otimes BK)\, \varepsilon(k), \tag{4.18}$$

based on equations (D.3), (D.4), (D.5) in Appendix D.

It is important to notice that, since $\varepsilon(k) = \left(\begin{bmatrix} r^{\mathrm{T}} \\ \mathcal{W} \end{bmatrix} \otimes I_n \right) \varpi(k)$ and with equations (D.6) and (D.7) in Appendix D,

$$\varepsilon_1(k) = (r^{\mathrm{T}} \otimes I_n)\varpi(k) = (r^{\mathrm{T}} \otimes I_n)(\widehat{M} \otimes I_n)X(k) \equiv 0. \tag{4.19}$$

The dynamics in (4.18) is expanded as

$$\varepsilon(k+1) = \begin{bmatrix} A - 0 \cdot BK & 0 & 0 & \dots 0 & \\ 0 & A - \lambda_2 BK & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A - \lambda_N BK \end{bmatrix} \varepsilon(k). \tag{4.20}$$

Thus, asymptotic stability of $\varepsilon_i(k),\ i = 2, \ldots, N$, i.e. $\varepsilon_i(k) \overset{k\to\infty}{\longrightarrow} 0$, is ensured if and only if the subsystems

$$\varepsilon_i(k+1) = (A - \lambda_i BK)\varepsilon_i(k),\ i = 2, \ldots, N, \tag{4.21}$$

are asymptotically stable, which is equivalent with $A - \lambda_i BK,\ i = 2, \ldots, N$ being stability matrices. This concludes the proof. □

Lemma 4.3.2. *[28] Let the set of N discrete-time dynamic agents described by equation (4.7) be interconnected through a network with a communication topology $\mathcal{G}$ with a directed spanning tree. If the distributed feedback control protocol in (4.9) satisfies Lemma 4.3.1, then*

$$x_i(k) \overset{k\to\infty}{\longrightarrow} \left(r^{\mathrm{T}} \otimes A^k\right) \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_N(0) \end{bmatrix},\ i = 1, 2, \ldots, N, \tag{4.22}$$

where $r \in \mathbb{R}^{N\times 1}$ is the left eigenvector of the digraph Laplacian corresponding to the zero eigenvalue ($r^{\mathrm{T}}\mathcal{L} = \mathbf{0}$), also satisfying $r^{\mathrm{T}}\mathbf{1} = 0$.

Proof. The solution to the agents dynamics defined in compact form in equation (4.11) is given by

$$X(k) = \left[I_N \otimes A - \mathcal{L} \otimes BK\right]^k X(0), \tag{4.23}$$

which, by using equations (D.3) and (D.4) in Appendix D, becomes

$$\begin{aligned} X(k) &= \left[\left(\mathcal{T}^{-1} \otimes I_n\right)^{-1} \left[I_N \otimes A - \mathcal{J} \otimes BK\right] \left(\mathcal{T}^{-1} \otimes I_n\right)\right]^k X(0) \\ &= \left(\mathcal{T}^{-1} \otimes I_n\right)^{-1} \left[I_N \otimes A - \mathcal{J} \otimes BK\right]^k \left(\mathcal{T}^{-1} \otimes I_n\right) X(0). \end{aligned} \tag{4.24}$$

Since $I_N \otimes A - \mathcal{J} \otimes BK = \begin{bmatrix} A & 0 \\ 0 & I_N \otimes A - \mathcal{H} \otimes BK \end{bmatrix}$ and $\left(\mathcal{T}^{-1} \otimes I_n\right)^{-1} = \mathcal{T} \otimes I_n$, it follows

that

$$X(k) = (\mathcal{T} \otimes I_n) \begin{bmatrix} A^k & 0 \\ 0 & [I_N \otimes A - \mathcal{H} \otimes BK]^k \end{bmatrix} \left(\mathcal{T}^{-1} \otimes I_n\right) X(0). \tag{4.25}$$

By Lemma 4.3.1, $I_N \otimes A - \mathcal{H} \otimes BK$ is a stability matrix, and therefore,

$$[I_N \otimes A - \mathcal{H} \otimes BK]^k \overset{k\to\infty}{\longrightarrow} 0.$$

Consequently,

$$X(k) \overset{k\to\infty}{\longrightarrow} (\mathcal{T} \otimes I_n) \begin{bmatrix} A^k & 0 \\ 0 & 0 \end{bmatrix} \left(\mathcal{T}^{-1} \otimes I_n\right). \tag{4.26}$$

Using the definitions of $\mathcal{T}$ and $\mathcal{T}^{-1}$ in equation (4.17), it can be written that

$$\begin{aligned} (\mathcal{T} \otimes I_n) \begin{bmatrix} A^k & 0 \\ 0 & 0 \end{bmatrix} \left(\mathcal{T}^{-1} \otimes I_n\right) &= (\mathbf{1} \otimes I_n) A^k (r^{\mathrm{T}} \otimes I_n) = (\mathbf{1} \otimes A^k)(r^{\mathrm{T}} \otimes I_n) \\ &= (\mathbf{1} \cdot r^{\mathrm{T}}) \otimes (A^k \otimes I_n) = (\mathbf{1} \cdot r^{\mathrm{T}}) \otimes A^k. \end{aligned} \tag{4.27}$$

From equations (4.26) and (4.27), it follows that

$$X(k) \overset{k\to\infty}{\longrightarrow} \left[(\mathbf{1} \cdot r^{\mathrm{T}}) \otimes A^k\right] X(0), \tag{4.28}$$

which implies that, for each of the N agents, equation (4.22) is valid. □

Remark 4.3.1. The conclusions of Lemmas 4.3.1 and 4.3.2 lead to some interesting observations regarding the consensus values with respect to the agents dynamics. Thus, if matrix A is a stability matrix, i.e. it has all eigenvalues inside the unit circle, then the consensus value reached by the agents is going to be 0. If matrix A has all eigenvalues outside the unit circle, the consensus is going to be reached asymptotically at infinity. The case when matrix A has eigenvalues on the unit circle proves to be the critical case for consensus of agents in (4.7) under the protocol (4.9) to happen at a constant value.

4.3.1 Single Input Systems

According to Lemma 4.3.1, consensusability is related to the eigenvalues of $A-\lambda_i BK$. Since the controller gain K is real (which is the case with physical systems), it could be considered that, for each component of the MAS, the feedback protocol gain is given by $\lambda_i K = |\lambda_i| K e^{j\theta_i}$. Under this alternate paradigm, the concepts of gain and phase margin for reaching consensusability are reinterpreted based on the following arguments. The gain margin is correlated to the range $|\lambda_i|$ can be adjusted supposing $\theta_i = 0$, for the agent to be consensusable. Likewise, the phase margin corresponds to the range that θ_i could be modified for a given $|\lambda_i|$, such that the agent remains consensusable. This perspective allows to look at the angle θ in Figure 4.1 as the phase margin, while the ratio $\frac{|\lambda_N|}{|\lambda_2|}$ is going to be considered the gain margin. Thus, in order to ensure a better design of the communication graph Laplacian, it is of paramount importance to find upper limits for these gain and phase margins based upon the agent model.

With $\lambda_i = (1+\delta_i)c$, $c \in \mathbb{R}$, $|\delta_i| \leq \delta$, $i = 2, \ldots, N$, stability of $A - \lambda_i BK$ is equivalent to $\det(zI - A + (1+\delta_i)cBK) \neq 0,\ \forall |z| \geq 1,\ \forall i = 2, \ldots, N$. This can be looked at as studying the robust stability of a plant $P_i(z) = (zI - A)^{-1}Bc(1+\delta_i)$ with multiplicative uncertainty. Typical $\mathcal{H}_\infty$ robust control results are employed to find upper bounds for the gain and phase margins.

Denote the complementary sensitivity function under state feedback by $T_0(z) = K(zI - A + BK)^{-1}B$, K^{T}, $B \in \mathbb{R}^{n\times 1}$. Define the $\mathcal{H}_\infty$ norm as $\|T_0(z)\|_\infty := \sup_{|z|>1} \bar{\sigma}(T_0(z))$, with $\bar{\sigma}(T_0(z))$ the largest singular value of $T_0(z)$. Let the Mahler measure of the $n \times n$ system matrix A be given by

$$\mu(A) = \prod_{i=1}^{n} \max\{1, |\lambda_i(A)|\}. \tag{4.29}$$

Lemma 4.3.3. *[4], [18], [21] If the pair* $(A,\ B)$ *is stabilizable, then*

$$\gamma_{\mathrm{opt}} = \inf_{K \in \mathbb{R}^{1\times n}} \|T_0(z)\|_\infty = \mu(A),$$

and for each $\gamma > \gamma_{\text{opt}}$*, there exists a stabilizing state feedback control gain such that* $\|T_0(z)\|_\infty < \gamma$*. Moreover,* $K = \left[1 + (1-\gamma^{-2})B^*XB\right]^{-1} B^*XA$*, where* $X \geq 0$ *is the stabilizing solution of the DARE*

$$X = A^*X\left[I + (1-\gamma^{-2})BB^*X\right]^{-1}A,\ B^*XB < \gamma^2. \tag{4.30}$$

Theorem 4.3.1. *Given a digraph* $\mathcal{G}$ *modeling the network topology of the discrete-time multi-agent system (4.7), with* $(A,\ B)$ *a stabilizable pair and* $m = 1$*, the system is consensusable under protocol (4.9) if*

$$\theta < \arcsin\sqrt{\frac{\left(\frac{r_N}{r_2}+1\right)^2}{4\frac{r_N}{r_2}\mu^2(A)} - \frac{\left(\frac{r_N}{r_2}-1\right)^2}{4\frac{r_N}{r_2}}}, \tag{4.31}$$

where r_2 *and* r_N *are the magnitudes of the second smallest and the largest eigenvalues of the digraph Laplacian.*

Proof. According to Lemma 4.2.1, the consensusability problem is solvable if and only if $A - \lambda_i BK$ is a stability matrix, where $\lambda_i,\ i = 2, \ldots, N$ are the complex eigenvalues for the Laplacian matrix $\mathcal{L}$ of the digraph $\mathcal{G}$. This condition is equivalent to

$$\det(zI - A + \lambda_i BK) \neq 0,\ \forall\, |z| \geq 1. \tag{4.32}$$

By the argument used in root locus method, the roots of $\det(zI - A + \lambda_i BK)$ are the same as for $\det\left(I + \lambda_i BK(zI-A)^{-1}\right)$. Consequently, condition in (4.32) is equivalent to

$$\det\left(I + \lambda_i BK(zI-A)^{-1}\right) = 1 + \lambda_i K(zI-A)^{-1}B \neq 0,\ \forall\, |z| \geq 1. \tag{4.33}$$

Since, for $C = I$, the agent transfer function is given by $P(z) = (zI-A)^{-1}B$, the condition

in equation (4.33) can be written as

$$1 + \lambda_i KP(z) \neq 0, \ \forall \, |z| \geq 1. \tag{4.34}$$

With $\lambda_i = (1 + \delta_i)c$, for some $c \in \mathbb{R}$, and $|\delta_i| \leq \delta$, for $i = 2, \ldots, N$, and some $\delta > 0$, the above statement is equivalent to

$$1 + (1 + \delta_i)cKP(z) = 1 + cKP(z) + \delta_i cKP(z) \neq 0, \ \forall \, |z| \geq 1. \tag{4.35}$$

Furthermore, the consensusability condition can be expressed as

$$1 + \delta_i \frac{cKP(z)}{1 + cKP(z)} \neq 0, \ \forall \, |z| \geq 1. \tag{4.36}$$

In light of Lemma 4.3.3, there exists a stabilizing state feedback controller gain K such that the complementary sensitivity function $T_c(z) = cK(zI - A + cBK)^{-1}B$ under state feedback for the nominal part of the uncertain agent $P_i(z)$ satisfies $\|T_c(z)\|_\infty < \gamma$ for $\gamma > \gamma_{\text{opt}} = \mu(A)$.

Since the system is single input, that is $B \in \mathbb{R}^{n \times 1}$, the complementary sensitivity function can be written as

$$T_c(z) = \frac{cK(zI - A)^{-1}B}{1 + cK(zI - A)^{-1}B} = \frac{cKP(z)}{1 + cKP(z)}. \tag{4.37}$$

Thus, the consensusability condition in equation (4.36) becomes

$$1 + \delta_i T_c(z) \neq 0, \ \forall \, |z| \geq 1. \tag{4.38}$$

With $0 < \theta < \frac{\pi}{2}$, the phase condition in equation (4.31) becomes:

$$|\sin\theta| < \sqrt{\frac{(r_N + r_2)^2}{4r_N r_2 \mu^2(A)} - \frac{(r_N - r_2)^2}{4r_N r_2}} \Leftrightarrow \sin^2\theta < \frac{(r_N + r_2)^2}{4r_N r_2 \mu^2(A)} - \frac{r_N^2 + r_2^2}{4r_N r_2} + \frac{1}{2}. \quad (4.39)$$

From equations (4.5) and (4.39), it follows that

$$\frac{r_N^2 + r_2^2 + 2r_N r_2(2\sin^2\theta - 1)}{(r_N + r_2)^2} = \frac{r_N^2 + r_2^2 + 2r_N r_2(\sin^2\theta - \cos^2\theta)}{(r_N + r_2)^2} = \delta^2 < \mu^{-2}(A) \quad (4.40)$$

which implies

$$\delta < \mu^{-1}(A). \quad (4.41)$$

Since $\delta_i \leq \delta$, equation (4.41) ensures the consensusability condition in equation (4.38) is satisfied. Moreover, together with equation (4.6), the condition in equation (4.41) implies:

$$\begin{aligned} &\frac{1}{\mu(A)} > \frac{|\lambda_N| - |\lambda_2|}{|\lambda_N| + |\lambda_2|} \Leftrightarrow \\ &\mu(A) < \frac{\left|\frac{\lambda_N}{\lambda_2}\right| + 1}{\left|\frac{\lambda_N}{\lambda_2}\right| - 1} \Leftrightarrow \\ &\left|\frac{\lambda_N}{\lambda_2}\right| < \frac{\mu(A) + 1}{\mu(A) - 1}. \end{aligned} \quad (4.42)$$

□

Theorem 4.3.1 provides a sufficient condition for reaching consensus among a set of discrete-time agents. It relates the gain and phase margin to an important measure of the agents stability, that is the Mahler measure. As expected and also shown in Figure 4.2, according to the conditions in Theorem 4.3.1, the consensusability region gets tighter and tighter as the agents are less and less stable.

Example 4.3.1. To prove the efficiency of assigning the Laplacian eigenvalues according to the inequalities in equations (4.31) and (4.42), consider the system described by equation

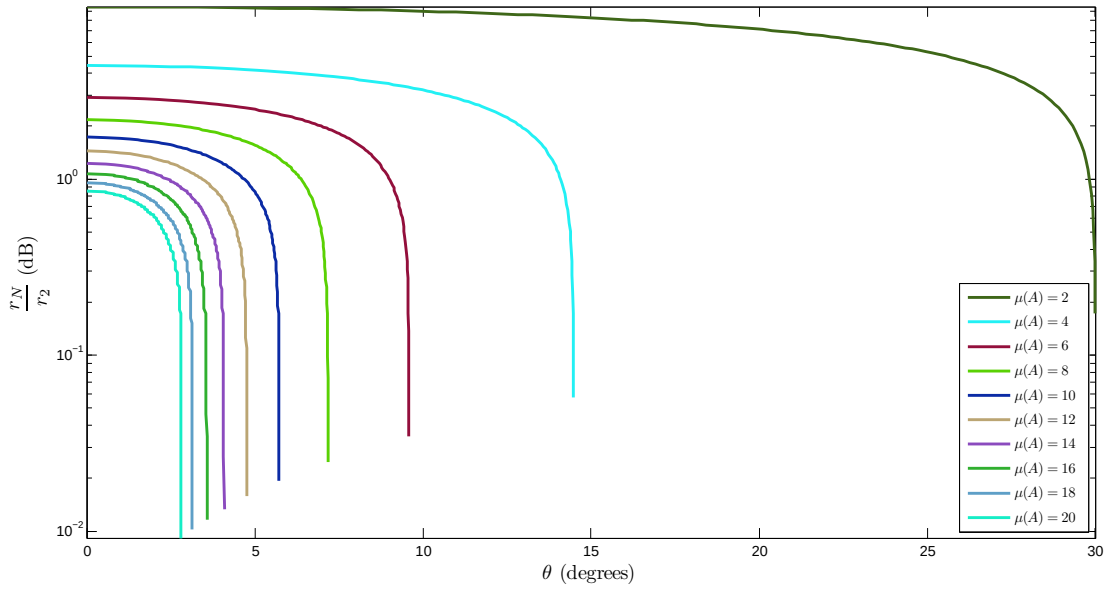


Figure 4.2: Consensusability region with respect to the system Mahler measure $\mu(A)$.

(4.7), for which the system matrices are given by:

$$A = \begin{bmatrix} 1.5 & 0.2 & 0 \\ -1.2 & 0 & 1.8 \\ 1.1 & 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 1.2 \\ -1.5 \\ 0.4 \end{bmatrix}, \; C = I_3. \tag{4.43}$$

The MAS consists of $N = 5$ agents, with each agent dynamics characterized by matrices in equation (4.43). The topology of the network connecting the 5 agents is a digraph whose weighted adjacency matrix is computed based on the following arguments.

The eigenvalues of matrix A are 1.5142, and $-0.0071 \pm 0.5113j$, with absolute values 1.5142, and 0.5114, respectively. Therefore, due to one eigenvalue outside the unit circle, each agent has an unstable behavior. The Mahler measure of matrix A is $\mu(A) = 1.5142$. To ensure consensusability of agents, the radius δ of the fan region where the eigenvalues of the digraph Laplacian should reside, must satisfy $\delta < \dfrac{1}{\mu(A)} = 0.66$, as depicted in Figure 4.3. According to equation (4.42), the non-zero smallest and largest eigenvalues of the Laplacian should satisfy $\dfrac{|\lambda_4|}{|\lambda_2|} < 4.89$. If $\dfrac{|\lambda_4|}{|\lambda_2|} = 1.2$ is chosen, equation (4.31) implies an upper bound for the angle θ of approximately 41°. Picking $\theta = 10°$ and the center of the

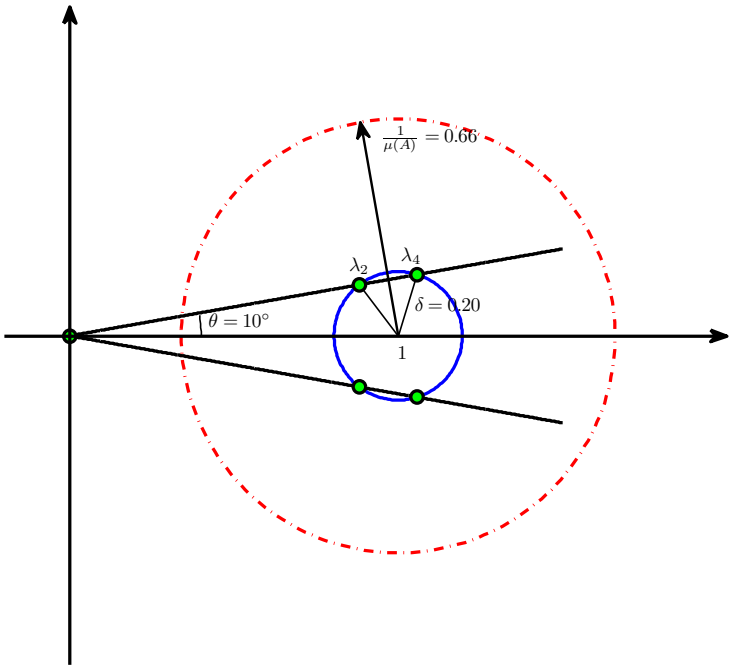


Figure 4.3: Position of the eigenvalues for the digraph Laplacian in Example 4.3.1.

fan region at $r_0 = 1$, it follows that $|\lambda_2| = 0.8953$ and $|\lambda_4| = 1.0743$, leading to the following set of eigenvalues, also depicted in Figure 4.3, for the digraph Laplacian:

$$\lambda_1 = 0,\ \lambda_{2,3} = 0.88168 \pm 0.15546,\ \lambda_{4,5} = 1.058 \pm 0.18656j.$$

Taking the inverse Fast Fourier Transform of a properly arranged vector of these eigenvalues, the Laplacian is obtained as

$$\mathcal{L} = \begin{bmatrix} 0.77588 & -0.26206 & -0.21812 & -0.24868 & -0.047017 \\ -0.047017 & 0.77588 & -0.26206 & -0.21812 & -0.24868 \\ -0.24868 & -0.047017 & 0.77588 & -0.26206 & -0.21812 \\ -0.21812 & -0.24868 & -0.047017 & 0.77588 & -0.26206 \\ -0.26206 & -0.21812 & -0.24868 & -0.047017 & 0.77588 \end{bmatrix}.$$

From the Laplacian definition in equation (4.1), the weighted adjacency matrix is computed

as

$$\mathcal{A} = \begin{bmatrix} 0 & 0.26206 & 0.21812 & 0.24868 & 0.047017 \\ 0.047017 & 0 & 0.26206 & 0.21812 & 0.24868 \\ 0.24868 & 0.047017 & 0 & 0.26206 & 0.21812 \\ 0.21812 & 0.24868 & 0.047017 & 0 & 0.26206 \\ 0.26206 & 0.21812 & 0.24868 & 0.047017 & 0 \end{bmatrix}.$$

In order to find the controller, the solution X to the DARE in equation (4.30) is solved for $\gamma = \dfrac{1}{\delta} = 5.1185 > \mu(A) = 1.5142$. Thus, the controller gain is given by

$$K = \left[1 + (1 - \gamma^{-2})B^*XB\right]^{-1} B^*XA = \begin{bmatrix} 0.83376 & 0.11012 & 0.13091 \end{bmatrix}.$$

In furtherance of this example, each element of the initial state vectors of all agents is randomly generated from the uniform distribution on the interval $[-5, 5]$. As it can be seen in Figures 4.4, 4.5, and 4.6, the deviation of each state of each agent from the average state value goes asymptotically to zero. In conclusion, the 5 agent system reaches a consensus.

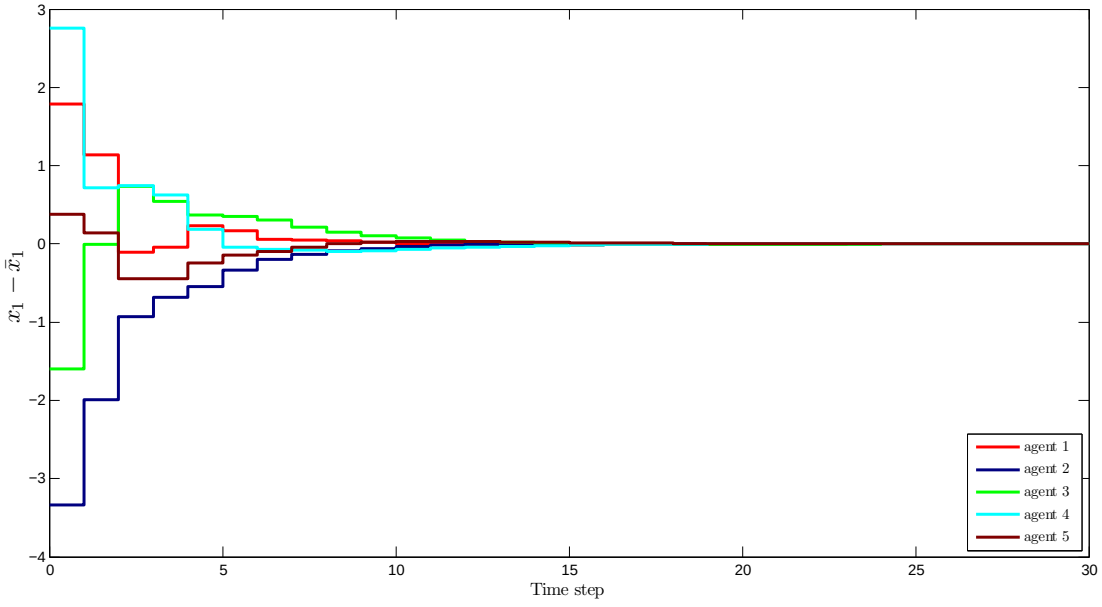


Figure 4.4: Deviations of states $x_1(k)$ from their average value $\bar{x}_1(k)$.

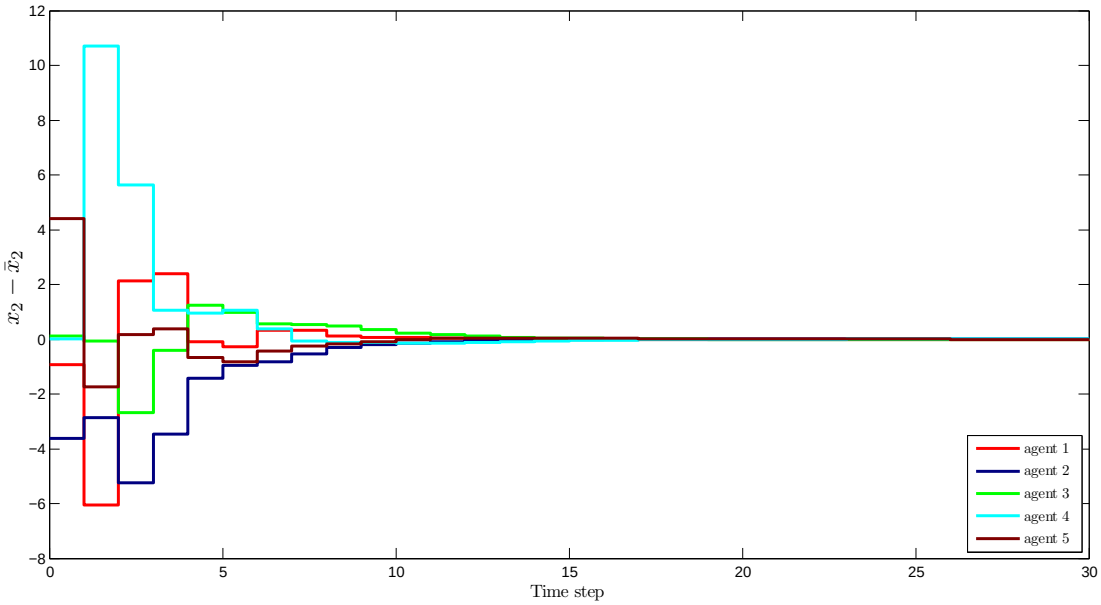


Figure 4.5: Deviations of states $x_2(k)$ from their average value $\bar{x}_2(k)$.

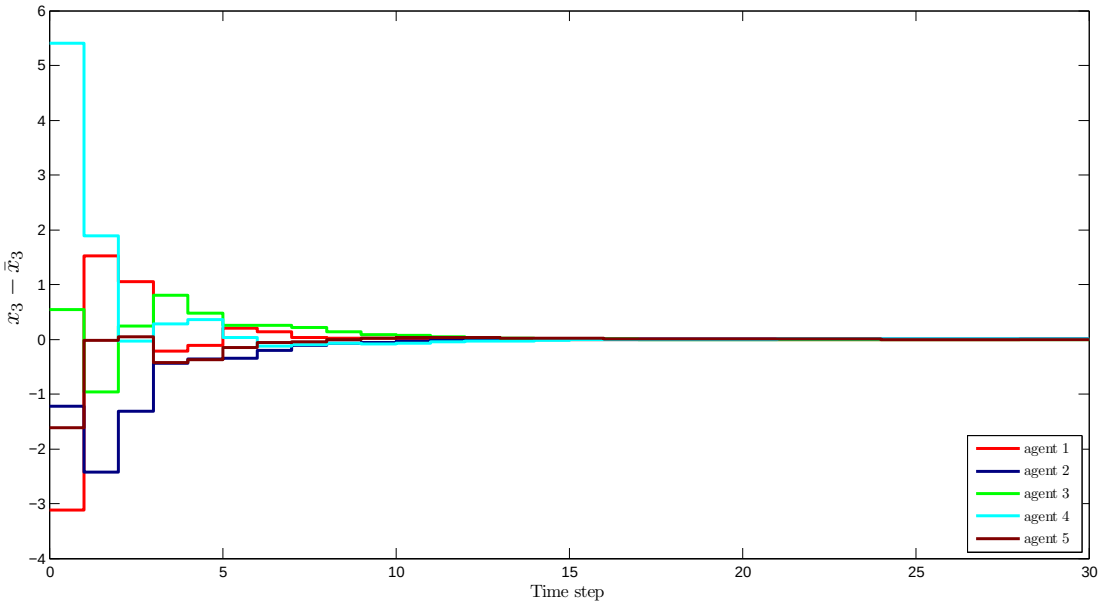


Figure 4.6: Deviations of states $x_3(k)$ from their average value $\bar{x}_3(k)$.

4.3.2 Multiple Input Systems

Consider the fundamental $\mathcal{H}_\infty$ optimization problem for a general system

$$x(k+1) = Ax(k) + Bu(k) + Bd(k),$$

where $x(k) \in \mathbb{R}^{n\times 1}$ is the state vector, $u(k) \in \mathbb{R}^{m\times 1}$ is the input vector, $d(k) \in \mathbb{R}^{m\times 1}$ is an energy bounded or $\mathcal{H}_2$-norm bounded disturbance, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and (A, B) is a stabilizable pair. Without loss of generality, assume that A and B are partitioned as follows

$$A = \begin{bmatrix} A_u & 0 \\ 0 & A_s \end{bmatrix}, \; B = \begin{bmatrix} B_u \\ B_s \end{bmatrix}, \tag{4.44}$$

where $A_u \in \mathbb{R}^{n_u\times n_u}$ is completely unstable, $A_s \in \mathbb{R}^{(n-n_u)\times(n-n_u)}$ is a stability matrix, and $B_u \in \mathbb{R}^{n_u\times m}$. Denote the complementary sensitivity function under state feedback by $T_0(z) = K(zI - A + BK)^{-1}B$, $K \in \mathbb{R}^{m\times n}$. For the case of multi-input systems, the following lemma is beneficial.

Lemma 4.3.4. *With $r = \text{rank}\{B_u\}$, there holds*

$$\sqrt[r]{\mu(A)} \leq \gamma_{opt} := \inf_{K\in\mathbb{R}^{m\times n}} \|T_0(z)\|_\infty \leq \mu(A), \tag{4.45}$$

with $\mu(A)$ being the Mahler measure defined in equation (4.29).

Proof. The objective of the state feedback $\mathcal{H}_\infty$ control problem is to apply the control law $u(k) = Kx(k)$ that would stabilize the feedback system and have an induced $\mathcal{H}_2$ norm not greater than $\gamma > \gamma_{\text{opt}}$ over all possible disturbances $d(k)$ with unity $\mathcal{H}_2$ norm. According to [4] and [18], such a state feedback control law exists and is equivalent with the existence of a stabilizing $X \geq 0$ satisfying

$$X = A^*X\left[I_n + (1-\gamma^{-2})BB^*X\right]^{-1}A, \; \gamma^2 I_m - B^*XB > 0. \tag{4.46}$$

First, assume that A_u does not have any eigenvalues on the unit circle. Then, the DARE in equation (4.46) admits a stabilizing semi-positive solution for $\gamma > \gamma_{\text{opt}}$, which has the

form

$$X = \begin{bmatrix} X_u & 0 \\ 0 & 0 \end{bmatrix} \Rightarrow X_u = A_u^* X_u \left[I_{n_u} + (1-\gamma^{-2}) B_u B_u^* X_u \right]^{-1} A_u, \tag{4.47}$$

with the condition $\gamma^2 I_m - B^* X B > 0$ being reduced to $\gamma^2 I_m - B_u^* X_u B_u > 0$. The pair $(A_u,\, B_u)$ is reachable due to the initial hypothesis of $(A,\, B)$ being stabilizable. Moreover, under the assumption made on A_u, the stabilizing solution X_u is nonsingular. Thus, the DARE is equivalent to the following Lyapunov equation

$$X_u^{-1} = A_u^{-1} X_u^{-1} (A_u^{-1})^* + (1-\gamma^{-2}) A_u^{-1} B_u B_u^* (A_u^{-1})^*. \tag{4.48}$$

The assumption on A_u also implies that A_u^{-1} is a stability matrix, and, therefore, the solution $X_u^{-1} > 0$ is unique and given by

$$\begin{aligned} X_u^{-1} &= (1-\gamma^{-2}) Z_u \\ Z_u &= \sum_{i=1}^{\infty} A_u^{-i} B_u B_u^* (A_u^{-i})^*. \end{aligned} \tag{4.49}$$

Since $\text{rank}\{B_u\} = r$, there exists a unitary matrix $Q \in \mathbb{R}^{r\times m}\, (QQ^* = I_r)$, such that $B_u = \widetilde{B}_u Q$ with $\widetilde{B}_u \in \mathbb{R}^{n_u \times r}$. Hence, the inequality in equation (4.46) becomes

$$\begin{aligned} &\gamma^2 I_r > \widetilde{B}_u^* X_u \widetilde{B}_u = (1-\gamma^{-2})^{-1} \widetilde{B}_u^* Z_u^{-1} \widetilde{B}_u \Leftrightarrow \\ &\gamma^2 I_r > I_r + \widetilde{B}_u^* Z_u^{-1} \widetilde{B}_u. \end{aligned} \tag{4.50}$$

Obviously, the last inequality entails $\gamma_{\text{opt}} = \sqrt{1 + \lambda_{max}(\widetilde{B}_u^* Z_u^{-1} \widetilde{B}_u)}$. Applying matrix determinant properties to inequality in equation (4.50) leads to

$$\begin{aligned} \gamma^{2r} &> \det(I_r + \widetilde{B}_u^* Z_u^{-1} \widetilde{B}_u) = \det(I_{n_u} + \widetilde{B}_u \widetilde{B}_u^* Z_u^{-1}) = \frac{\det(Z_u + \widetilde{B}_u \widetilde{B}_u^*)}{\det(Z_u)} \\ &= \frac{\det(Z_u + \widetilde{B}_u \widetilde{B}_u^*)}{\det[(A_u^{-1})(Z_u + \widetilde{B}_u \widetilde{B}_u^*)(A_u^{-1})^*]} = \det(A_u^* A_u) = \mu^2(A) \Rightarrow \gamma > \sqrt[r]{\mu(A)}. \end{aligned}$$

Therefore, if $\gamma^2 I_r > I_r + \widetilde{B}_u^* Z_u^{-1} \widetilde{B}_u$, then $X_u = (1-\gamma^{-2})^{-1} Z_u^{-1} > 0$ exists, and the inequality $\gamma > \sqrt[r]{\mu(A)}$ holds. Consequently, the inequalities in equation (4.45) are obtained by taking γ sufficiently close to γ_{opt}. This concludes the proof of equation (4.45) under the assumption that A has no eigenvalues on the unit circle. Suppose that A has eigenvalues on the unit circle. The complementary sensitivity function whose $\mathcal{H}_\infty$-norm is to be minimized over all stabilizing state feedback gains K, can be rewritten as

$$T_\varepsilon = \begin{bmatrix} K \\ \varepsilon C \end{bmatrix} (zI - A + BK)^{-1} B,\ \varepsilon > 0,\ C = \begin{bmatrix} 0 & C_u \end{bmatrix},\ C_u \in \mathbb{R}^{1 \times n_u},$$

with $(C_u,\ A_u)$ being an observable pair. Since A_u is nonsingular, such a C_u exists. Then, it can be argued that given any $\gamma > \gamma_{\text{opt}}$, $\|T_\varepsilon\|_\infty < \gamma$ could be achieved for some stabilizing K by taking $\varepsilon > 0$ sufficiently small. □

Remark 4.3.2. With $\lambda_i(\cdot)$ denoting the i-th eigenvalue, if $\text{rank}\{B_u\} = m$, then

$$\lambda_i(I_m + B_u^* Z_u^{-1} B_u) = 1 + \lambda_i(B_u^* Z_u^{-1} B_u),$$

and $\gamma_{\text{opt}}^2 = 1 + \lambda_{max}(B_u^* Z_u^{-1} B_u)$. Hence, the condition number of $B_u^* Z_u^{-1} B_u$ determines how tight the lower and upper bounds in equation (4.45) are. A condition number close to 1 will get γ_{opt} closer to the lower bound. Otherwise, γ_{opt} is close to the upper bound. As a matter of fact, if $\text{rank}\{B_u\} = 1$, the problem is similar to the single input case, and, therefore, γ_{opt} equals the upper bound. Also, if $X \geq 0$ is the stabilizing solution of the DARE in equation (4.46) satisfying the inequality in equation (4.46), then a stabilizing state feedback gain K achieving $\|T_0\|_\infty < \gamma$, for some $\gamma > \gamma_{\text{opt}}$ is implemented according to

$$K = \left[I_m + (1 - \gamma^{-2}) B^* X B\right]^{-1} B^* X A.$$

Rewrite the complementary sensitivity function under state feedback by introducing a

square nonsingular matrix D, as shown below

$$T_D(z) = K_D(zI - A + BD^{-1}K_D)^{-1}BD^{-1}. \tag{4.51}$$

Let $\text{rank}\{B\} = \text{rank}\{B_u\} = r > 0$. Then

$$\sqrt[r]{\mu(A)} \leq \gamma_{\text{opt}} := \inf_{D,\, K_D} \|T_D\|_\infty \leq \mu(A).$$

It can be proven that by properly designing D and K_D, the following result holds

$$\inf_{D,\, K_D} \|T_D\|_\infty \approx \sqrt[r]{\mu(A)}.$$

This conclusion is supported by the following example.

Example 4.3.2. Consider a system described by the state-space model $x(k+1) = Ax(k) + Bu(k)$, with

$$A = \begin{bmatrix} -0.4326 & 1.1909 & -0.1867 & 0.1139 & 0.2944 \\ -1.6656 & 1.1892 & 0.7258 & 1.0668 & -1.3362 \\ 0.1253 & -0.0376 & -0.5883 & 0.0593 & 0.7143 \\ 0.2877 & 0.3273 & 2.1832 & -0.0956 & 1.6236 \\ -1.1465 & 0.1746 & -0.1364 & -0.8323 & -0.6918 \end{bmatrix};$$

$$B = \begin{bmatrix} 1.7160 & -0.7998 & 1.3372 \\ 2.5080 & 1.3800 & 2.3817 \\ -3.1875 & 1.6312 & -2.4049 \\ -2.8819 & 1.4238 & -0.0396 \\ 1.1423 & 2.5805 & -0.3134 \end{bmatrix}.$$

The set of eigenvalues for A is $\{-1.1192,\ -0.5521 \pm 1.7693j,\ 0.8022 \pm 0.877j\}$, with absolute values $\{1.1192, 1.8535, 1.1886\}$, respectively, and $\text{rank}\{B\} = 3$. Thus, in Lemma 4.3.4, $A_u = A$ and $B_u = B$. Applying the definition of Mahler measure, $\mu(A) = 5.432$, and hence, $\sqrt[3]{\mu(A)} = 1.758$. Following the steps detailed in the proof of Lemma 4.3.4, the Lyapunov equation (4.48) is solved for $[(1-\gamma^{-2})X_u]^{-1} = Z_u = Z$. It follows that

$$1.758 = \sqrt[3]{\mu(A)} < \gamma_{\text{opt}} = \sqrt{1 + \lambda_{\max}(B^* Z^{-1} B)} = 2.964 < \mu(A) = 5.432.$$

The next question is whether or not γ_{opt} can be made closer and closer to $\sqrt[3]{\mu(A)}$, the lower bound in equation (4.45), by involving a square nonsingular matrix D. First, the Cholesky factorization is performed on $B^* Z^{-1} B = D_1^* D_1$, and set $D = D_1$. As explained at the end of Remark 4.3.2, a minimization of $\|T_D(z)\|_\infty$, with $T_D(z)$ as in equation (4.51), is performed over all state feedback gains, according to Lemma 4.3.4. This yields

$$Z_1 = \sum_{k=1}^{\infty} A^{-k} B (D_1^* D_1)^{-1} B^* (A^*)^{-k} > 0,$$

and $\gamma_{\text{opt}}^{(1)} = \sqrt{1 + \lambda_{\max}((BD^{-1})^* Z_1^{-1} (BD^{-1}))} = 2.396$. The process can be further repeated by taking a new Cholesky decomposition of $(BD^{-1})^* Z_1^{-1} (BD^{-1}) = D_2^* D_2$, setting $D = D_2 D_1$, and minimizing $\|T_D(z)\|_\infty$, which leads to a new $\gamma_{\text{opt}}^{(2)} = 2.09$. Proceeding with more iterations, a sequence $\{D_i\}_{i=1}^{\mathcal{K}}$ of square nonsingular matrices is obtained. Setting $D = D_{\mathcal{K}} \dots D_2 D_1$, $\|T_D(z)\|_\infty$ is minimized over all sate feedback gains yielding $\gamma_{\text{opt}}^{(\mathcal{K})}$. Figure 4.7 shows the computation results after $\mathcal{K} = 15$ iterations, leading to $\gamma_{\text{opt}}^{(15)} = 1.8537$, which is about 5.5% higher than the lower bound. Regardless of the fact that the method in this example fails to reach the exact lower bound, the improvement of γ_{opt} by refining the condition number of $B^* Z^{-1} B$ with a square nonsingular matrix D is evident. Moreover, it can be noticed that $\gamma_{\text{opt}}^{(\mathcal{K})}$ drops fast in the beginning, which entails that a small value for $\mathcal{K}$ is sufficient to reach the goal.

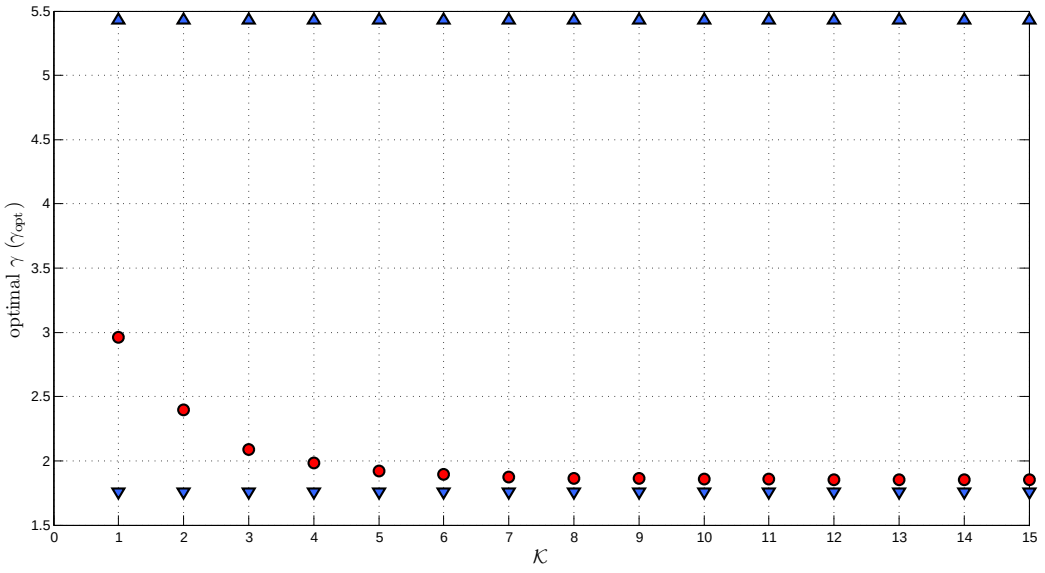


Figure 4.7: $\gamma_{\rm opt}^{(\mathcal{K})}$ ($\circ$- curve), lower bound (∇-line), upper bound (Δ-line).

Corollary 4.3.1. *Given a digraph $\mathcal{G}$ modeling the network topology of the discrete-time multi-agent system (4.7), with $(A,\, B)$ a stabilizable pair and $m > 1$, the system is consensusable under protocol (4.9) if*

$$\theta < \arcsin \sqrt{\frac{\left(\frac{r_N}{r_2}+1\right)^2}{4\frac{r_N}{r_2}\mu_{\rm opt}^2} - \frac{\left(\frac{r_N}{r_2}-1\right)^2}{4\frac{r_N}{r_2}}}, \tag{4.52}$$

where r_2 and r_N are the magnitudes of the second smallest and the largest eigenvalues of the digraph Laplacian, and $\mu_{\rm opt} = \gamma_{\rm opt}$ verifies the lower and upper bounds in equation (4.45).

Chapter 5
Conclusions and Future Work

Inspired by the multitude of applications NCS area has to offer, this dissertation has studied two of the current challenges related to communication-constrained feedback stability and consensusability of components in an MAS.

The first part of the thesis had the primary goal of analyzing the stabilization of LDTI systems using only a finite number of control values and measurement levels. In other words, due to limited bandwidth channels, the communication between actuators and plant, and between sensors and controller is prone to quantization errors, which, in turn, propagate to the systems states. Unlike the previous studies, the new approach has considered the quantized input values as being known, and thus available to the controller. Therefore, a new control law has been proposed and used in this study in order to synthesize the controllers that would ensure stability even in the presence of a more coarsely quantized signal. Following the idea in [16], this work has extended the use of classical sector bound method to LDTI systems under the newly suggested feedback. The quantization errors had been converted into sector bound uncertainties, so that the quantized feedback problems could be treated as robust control problems. First, the case with only quantized input has been considered. It has been shown what the coarsest quantization density is and how to design the controller under state feedback, and the observer-based controller under output feedback case. Then, the method has been extended to systems affected by logarithmic quantization at both input and output. The largest admissible uncertainty, which represents the sector bound condition, has been estimated using a scaled $\mathcal{H}_\infty$ optimization algorithm, that has been detailed, coded and used to back up the analytical results through a series of examples.

The second part of the dissertation has dealt with the problem of consensusability for discrete-time MASs under a time-invariant communication topology described by a

directed graph. Under a proposed distributed feedback control protocol based on agents' local information, the consensusability condition has been derived. It has been shown that this condition depends strictly on the Mahler measure of the agent, and is given in terms of the second smallest and largest eigenvalues of the digraph Laplacian.

Combining the two main studies of this thesis into one could be the topic of a future research work. It would be of great interest to define the consensusability condition and how to design the consensus control protocol in the case of a communication-constrained network between agents.

References

[1] B. D. O. Anderson. The small-gain theorem, the passivity theorem, and their equivalence. *Journal of the Franklin Institute*, 293(2):105–115, February 1972.

[2] Panos Antsaklis. Special issues on networked control systems. *IEEE Transactions on Automatic Control*, 49(9):1421–1423, September 2004.

[3] Panos Antsaklis and John Baillieul. Special issue on technology of networked control systems. In *Proceedings of the IEEE*, volume 95, pages 5–8, January 2007.

[4] T. Başar and P. Bernhard. $\mathcal{H}_\infty$ optimal control and related minimax design problems: A dynamic game approach. In *Systems and Control: Foundations and Applications*. Birkhäuser, Boston, MA, 1991.

[5] John Baillieul. *Feedback Designs in Information-Based Control*, volume 280. Springer Berlin/Heidelberg, 2002.

[6] John Baillieul and Panos Antsaklis. Control and communication challenges in networked real-time systems. In *Proceedings of the IEEE*, volume 95, pages 9–28, January 2007.

[7] Lowell W. Beineke, Robin J. Wilson, and Peter J. Cameron. *Topics in Algebraic Graph Theory*. Cambridge Univeristy Press, 2004.

[8] T. Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[9] Richard A. Brualdi and Herbert J. Ryser. *Combinatorial Matrix Theory*. Cambridge University Press, 1991.

[10] Daniel Coutinho, Minyue Fu, and Carlos E. De Souza. Output feedback control of linear systems with input and output quantization. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4706–4711, Cancun, Mexico, December 2008.

[11] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.

[12] Renwick E. Curry. *Estimation and Control with Quantized Measurements*. MIT Press, Cambridge, MA, 1970.

[13] C. E. De Souza, M. Fu, and L. Xie. H-∞ analysis and synthesis of discrete-time systems with time-varying uncertainty. *IEEE Transaction on Automatic Control*, 38(3):459–462, March 1993.

[14] J. C. Doyle and Stein G. Multivariable feedback design: Concepts for a classical/-modern synthesis. *IEEE Transactions in Automatic Control*, 26(1):4–16, February 1981.

[15] Nicola Elia and Sanjoy K. Mitter. Stabilization of linear systems with limited information. *IEEE Transactions of Automatic Control*, 46(9):1384–1400, September 2001.

[16] Minnyue Fu and Lihua Xie. The sector bound approach to quantized feedback control. *IEEE Transactions on Automatic Control*, 50(11):1698–1711, November 2005.

[17] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer, 2001.

[18] M. Green and D. J. L. Limebeer. *Linear Robust Control*. Prentice-Hall, 1995.

[19] G. Gu, L. Marinovici, and F. Lewis. Consensusability and consensus control of discrete-time multi-agent systems. In *Proceedings of the 30th Chinese Control Conference*, 2011.

[20] Guoxiang Gu and Li Qiu. Connection of multiplicative/relative perturbation in coprime factors and gap metric uncertainty. *Automatica*, 34(5):603 – 607, 1998.

[21] Guoxiang Gu and Li Qiu. Networked stabilization of multi-input sytems with channel resource allocation. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 2008.

[22] Guoxiang Gu and Li Qiu. Stabilization of networked multi-input systems with channel resource allocation. In *10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, December 2008.

[23] Guoxiang Gu and Li Qiu. Stabilization of multiplicative/relative uncertain systems with applications to networked feedback control. In *Proceedings of the 7th International Conference on Control and Automation*, Christchurch, New Zealand, December 2009.

[24] João P. Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. In *Proceedings of the IEEE*, volume 95, pages 138–162, January 2007.

[25] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge Univeristy Press, 1991.

[26] Pablo A. Iglesias and Keith Glover. State-space approach to discrete-time $\mathcal{H}_\infty$ control. *International Journal of Control*, 54(5):1031 – 1073, 1991.

[27] P. Lancaster and M. Tismenetsky. *The theory of Matrices: with applications*. Academic Press, 2 edition, 1985.

[28] Z. Li, Z. Duan, G. Chen, and L. Huang. Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint. *IEEE Transactions on Circuits and Systems - I*, 57(1), 2010.

[29] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., 1996.

[30] Cui-Qin Ma and Ji-Feng Zhang. Necessary and sufficient conditions for consensusability of linear multi-agent systems. *IEEE Transactions on Automatic Control*, 55(5):1263 – 1268, May 2010.

[31] Stephe D. J. McArthur, Euan M. Davidson, Victoria M. Catterson, Aris L Dimeas, Nikos D. Hatziargyriou, Ferdinanda Ponci, and Toshihisa Funabashi. Multi-agent systems for power engineering applications - part i: Concepts, approaches and technical chalenges. *IEEE Transactions on Power Systems*, 22(4):1743 – 1752, November 2007.

[32] Girish N. Nair and Robin J. Evans. Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica*, 39(4):585–593, April 2003.

[33] Girish N. Nair, Fabio Fagnani, and Robin J. Evans. Feedback control under data rate constraints: An overview. In *Proceedings of the IEEE*, volume 95, pages 108–137, January 2007.

[34] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), January 2007.

[35] A. Packard and J. C. Doyle. Quadratic stability with real and complex perturbations. *IEEE Transactions on Automatic Control*, 35(2):198–201, February 1990.

[36] K. B. Petersen and M. S. Pedersen. The matrix cookbook, october 2008. Version 20081110.

[37] W. Ren and R. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, May 2005.

[38] J. S. Shamma. Robust stability with time-varying structured uncertainty. *IEEE Transactions on Automatic Control*, 39(4):714–724, April 1994.

[39] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948. Reprinted in 'Claude Elwood Shannon Collected Papers', IEEE Press, 1993.

[40] Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *Institute of Radio Engineers*, 1959. Reprinted in 'Claude Elwood Shannon Collected Papers', IEEE Press, 1993.

[41] Peter Stoen and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robotics*, 8(3), July 2000.

[42] Sekhar Tatikonda and Sanjoy Mitter. Control under communication constraints. *IEEE Transactions in Automatic Control*, 49(7):1056–1068, July 2004.

[43] Y. Tipsuwan and M.-Y. Chow. Fuzzy logic microcontroller implementation for dc motor speed control. In *Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics SOCIETY (IECON 99)*, volume 3, pages 1271–1276, 1999.

[44] W. S. Wong and R. W. Brockett. Systems with finite communication bandwidth constraints ii: Stabilization with limited information feedback. *IEEE Transactions on Automatic Control*, 44(5):1049–1053, May 1999.

[45] Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley and Sons, Ltd., 2002.

[46] Chai Wah Wu. Synchronization in networks of nonlinear dynamical systems coupled via a directed graph. *Nonlinearity*, 18:1057 – 1064, February 2005.

[47] L. Xie and Y. C. Soh. Guaranteed cost control of uncertain discrete-time systems. *Control Theory and Technology*, 10(4):1235–1251, 1995.

[48] Keyou You and Lihua Xie. Consensusability of discrete-time multi-agent systems via relative output feedback. In *Proceedings of 2010 11th International Conference on Control, Automation, Robotics and Vision*, pages 1239 – 1244, 2010.

[49] Keyou You and Lihua Xie. Coordination of discrete-time multi-agent systems via relative output feedback. *International Journal of Robust and Nonlinear Control*, May 2010.

[50] Keyou You and Lihua Xie. Network topology and communication data rate for consensusability of discrete-time multi-agent systems. *IEEE Transactions on Automatic Control*, 2011. Accepted for publication in the Special Issue on Wireless Sensor and Actuator Networks.

[51] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

Appendix A

Controller Synthesis - Simple Output Feedback Case

For the case in Figure 3.7, using the scaling matrices $\widetilde{W_l} = \begin{bmatrix} \tau I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_p \end{bmatrix}$, and $\widetilde{W_r} = \begin{bmatrix} \tau I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_q \end{bmatrix}$, $\tau > 0$ the system becomes:

$$\begin{aligned}
\widetilde{W_l} G \widetilde{W_r}^{-1} &= \left[\begin{array}{ccc} \tau I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_p \end{array}\right] \left[\begin{array}{cc:c} 0 & 0 & I_q \\ P & 0 & P \\ \hdashline P & I_p & P \end{array}\right] \left[\begin{array}{ccc} \tau^{-1} I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_q \end{array}\right] \\
&= \left[\begin{array}{cc:c} 0 & 0 & \tau I_q \\ P & 0 & P \\ \hdashline P & I_p & P \end{array}\right] \left[\begin{array}{ccc} \tau^{-1} I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_q \end{array}\right] \\
&= \left[\begin{array}{cc:c} 0 & 0 & \tau I_q \\ \tau^{-1} P & 0 & P \\ \hdashline \tau^{-1} P & I_p & P \end{array}\right] = \left[\begin{array}{c|cc:c} A & B & 0 & \tau B \\ \hline 0 & 0 & 0 & \tau I_q \\ \tau^{-1} C & 0 & 0 & 0 \\ \hdashline \tau^{-1} C & 0 & I_p & 0 \end{array}\right].
\end{aligned} \tag{A.1}$$

Denoting $\widetilde{B}_1 = \begin{bmatrix} B & 0 \end{bmatrix}$, $\widetilde{B}_2 = \tau B$, $\widetilde{C}_1 = \begin{bmatrix} 0 \\ \tau^{-1}C \end{bmatrix}$, $\widetilde{C}_2 = \tau^{-1}C$, $\widetilde{D}_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $\widetilde{D}_{12} = \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix}$, $\widetilde{D}_{21} = \begin{bmatrix} 0 & I_p \end{bmatrix}$, and $\widetilde{D}_{22} = 0$, it can be seen that one of the simplified assumptions in the $\mathcal{H}_\infty$ theory is not satisfied, that is

$$\widetilde{D}_{12}^* \begin{bmatrix} \widetilde{C}_1 & \widetilde{D}_{12} \end{bmatrix} = \begin{bmatrix} \tau I_q & 0 \end{bmatrix} \begin{bmatrix} 0 & \tau I_q \\ \tau^{-1}C & 0 \end{bmatrix} = \begin{bmatrix} 0 & \tau^2 I_q \end{bmatrix} \neq \begin{bmatrix} 0 & I_q \end{bmatrix}.$$

The other condition

$$\begin{bmatrix} \widetilde{B}_1 \\ \widetilde{D}_{21} \end{bmatrix} \widetilde{D}_{21}^* = \begin{bmatrix} B & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} 0 \\ I_p \end{bmatrix} = \begin{bmatrix} 0 \\ I_p \end{bmatrix}$$

is satisfied. Normalizing $\widetilde{D}_{12} = \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix}$ by writing it as

$$\widetilde{D}_{12} = U_p \begin{bmatrix} 0 \\ I_q \end{bmatrix} R_p = \begin{bmatrix} 0 & I_q \\ I_p & 0 \end{bmatrix} \begin{bmatrix} 0 \\ I_q \end{bmatrix} \tau I_q = \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix},$$

and writing $\widetilde{D}_{21}$ as

$$\widetilde{D}_{21} = \widetilde{R}_p \begin{bmatrix} 0 & I_p \end{bmatrix} \widetilde{U}_p = I_p \begin{bmatrix} 0 & I_p \end{bmatrix} \begin{bmatrix} I_q & 0 \\ 0 & I_p \end{bmatrix} = \begin{bmatrix} 0 & I_p \end{bmatrix},$$

the new model for $\mathcal{H}_\infty$ synthesis becomes

$$
\begin{aligned}
\widehat{G} &= \left[\begin{array}{cc:c} 0 & I_p & 0 \\ I_q & 0 & 0 \\ \hdashline 0 & 0 & I_p \end{array}\right] \widetilde{W}_l G \widetilde{W}_r^{-1} \left[\begin{array}{cc:c} I_q & 0 & 0 \\ 0 & I_p & 0 \\ \hdashline 0 & 0 & \tau^{-1} I_q \end{array}\right] \\
&= \left[\begin{array}{c|c:c} A & \begin{bmatrix} B & 0 \end{bmatrix} \begin{bmatrix} I_q & 0 \\ 0 & I_p \end{bmatrix} & \tau B \tau^{-1} I_q \\ \hline \begin{bmatrix} 0 & I_p \\ I_q & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \tau^{-1} C \end{bmatrix} & \begin{bmatrix} 0 & I_p \\ I_q & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_q & 0 \\ 0 & I_p \end{bmatrix} & \begin{bmatrix} 0 & I_p \\ I_q & 0 \end{bmatrix} \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix} \tau^{-1} I_q \\ \hdashline I_p \tau^{-1} C & I_p \begin{bmatrix} 0 & I_p \end{bmatrix} \begin{bmatrix} I_q & 0 \\ 0 & I_p \end{bmatrix} & I_p 0 \tau^{-1} I_q \end{array}\right] \\
&= \left[\begin{array}{c|cc:c} A & B & 0 & B \\ \hline \tau^{-1} C & 0 & 0 & 0 \\ 0 & 0 & 0 & I_q \\ \hdashline \tau^{-1} C & 0 & I_p & 0 \end{array}\right]
\end{aligned}
\tag{A.2}
$$

The new partitioned matrices for $\widehat{G}$ are $\widehat{B}_1 = \begin{bmatrix} B & 0 \end{bmatrix}$, $\widehat{B}_2 = B$, $\widehat{C}_1 = \begin{bmatrix} \tau^{-1} C \\ 0 \end{bmatrix}$, $\widehat{C}_2 = \tau^{-1} C$, $\widehat{D}_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $\widehat{D}_{12} = \begin{bmatrix} 0 \\ I_q \end{bmatrix}$, $\widehat{D}_{21} = \begin{bmatrix} 0 & I_p \end{bmatrix}$, and $\widehat{D}_{22} = 0$, and thus the orthogonality conditions are satisfied.

Denote by K_G the stabilizing controller for G and by $K_{\widehat{G}}$ the stabilizing controller for $\widehat{G}$. Then, it follows that

$$K_{\widehat{G}} = \tau I_q K_G I_p = \tau K_G. \tag{A.3}$$

According to the solution to the discrete-time $\mathcal{H}_\infty$ control problem in [26], some preliminary matrices must be defined before proceeding with synthesizing the actual controller.

$$D_{1\bullet} = \begin{bmatrix} \widehat{D}_{11} & \widehat{D}_{12} \end{bmatrix} = \left[\begin{array}{cc:c} 0 & 0 & 0 \\ 0 & 0 & I_q \end{array}\right] = \begin{bmatrix} 0_{p\times(q+p)} & 0_{p\times q} \\ 0_{q\times(q+p)} & I_q \end{bmatrix}. \tag{A.4}$$

$$D_{\bullet 1} = \begin{bmatrix} \widehat{D}_{11} \\ \widehat{D}_{21} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \hdashline 0 & I_p \end{bmatrix} = \begin{bmatrix} 0_{(p+q)\times q} & 0_{(p+q)\times p} \\ 0_{p\times q} & I_p \end{bmatrix}. \tag{A.5}$$

From equations (A.4) and (A.5), it follows that

$$R = D_{1\bullet}^* D_{1\bullet} - \begin{bmatrix} \gamma^2 I & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -\gamma^2 I_{q+p} & 0_{(q+p)\times q} \\ 0_{q\times(q+p)} & I_q \end{bmatrix}, \tag{A.6}$$

$$\widetilde{R} = D_{\bullet 1} D_{\bullet 1}^* - \begin{bmatrix} \gamma^2 I & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -\gamma^2 I_{p+q} & 0_{(p+q)\times p} \\ 0_{p\times(p+q)} & I_p \end{bmatrix}. \tag{A.7}$$

The DAREs needed to solve the $\mathcal{H}_\infty$ control problem involve two pairs of symplectic matrices defines as follows:

$$\begin{aligned} S_\infty &:= \left(\begin{bmatrix} A - \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix} R^{-1} D_{1\bullet}^* \widehat{C}_1 & 0 \\ -\widehat{C}_1^* \left(I - D_{1\bullet} R^{-1} D_{1\bullet}^*\right) \widehat{C}_1 & I \end{bmatrix}, \begin{bmatrix} I & \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix} R^{-1} \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix}^* \\ 0 & \left(A - \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix} R^{-1} D_{1\bullet}^* \widehat{C}_1 \right)^* \end{bmatrix} \right) \\ &= \left(\begin{bmatrix} A & 0 \\ -\tau^{-2} C^* C & I \end{bmatrix}, \begin{bmatrix} I & (1-\gamma^{-2}) B B^* \\ 0 & A^* \end{bmatrix} \right) = (S_{\infty 1}, S_{\infty 2}) \end{aligned} \tag{A.8}$$

and

$$T_\infty := \left(\begin{bmatrix} \left(A - \widehat{B}_1 D_{\bullet 1}^* \widetilde{R}^{-1} \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix} \right)^* & 0 \\ -\widehat{B}_1 \left(I - D_{\bullet 1}^* \widetilde{R}^{-1} D_{\bullet 1} \right) \widehat{B}_1^* & I \end{bmatrix}, \begin{bmatrix} I & \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix}^* \widetilde{R}^{-1} \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix} \\ 0 & A - \widehat{B}_1 D_{\bullet 1}^* \widetilde{R}^{-1} \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix} \end{bmatrix} \right)$$

$$= \left(\begin{bmatrix} A^* & 0 \\ -BB^* & I \end{bmatrix}, \begin{bmatrix} I & (1-\gamma^{-2})\tau^{-2} C^* C \\ 0 & A \end{bmatrix} \right) = (T_{\infty 1}, T_{\infty 2}). \tag{A.9}$$

The DAREs corresponding to the symplectic pairs in equations (A.8) and (A.9) are therefore given by

$$X_\infty = A^* X_\infty \left[I + (1-\gamma^{-2}) B B^* X_\infty \right]^{-1} A + \tau^{-2} C^* C, \tag{A.10}$$

$$Y_\infty = A Y_\infty \left[I + (1-\gamma^{-2}) \tau^{-2} C^* C Y_\infty \right]^{-1} A^* + B B^*. \tag{A.11}$$

Once an admissible controller $K_{\widehat{G}}$ for $\widehat{G}$ has been designed, according to equation A.3, it results that

$$K_G = \tau^{-1} K_{\widehat{G}}. \tag{A.12}$$

Appendix B

Controller Synthesis - Input-Output Feedback Case

For the case in Figure 3.9, using the scaling matrices $\widetilde{W}_l = \begin{bmatrix} \tau I_q & 0 & 0 & 0 \\ 0 & I_p & 0 & 0 \\ 0 & 0 & I_q & 0 \\ 0 & 0 & 0 & I_p \end{bmatrix}$, and $\widetilde{W}_r = \begin{bmatrix} \tau I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_q \end{bmatrix}$, $\tau > 0$ the system becomes:

$$\widetilde{W}_l G \widetilde{W}_r^{-1} = \left[\begin{array}{cccc} \tau I_q & 0 & 0 & 0 \\ 0 & I_p & 0 & 0 \\ 0 & 0 & I_q & 0 \\ 0 & 0 & 0 & I_p \end{array}\right] \left[\begin{array}{cc:c} 0 & 0 & I_q \\ P & 0 & P \\ \hdashline I_q & 0 & I_q \\ P & I_p & P \end{array}\right] \left[\begin{array}{ccc} \tau^{-1} I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_q \end{array}\right]$$

$$= \left[\begin{array}{cc:c} 0 & 0 & \tau I_q \\ P & 0 & P \\ \hdashline I_q & 0 & I_q \\ P & I_p & P \end{array}\right] \left[\begin{array}{ccc} \tau^{-1} I_q & 0 & 0 \\ 0 & I_p & 0 \\ 0 & 0 & I_q \end{array}\right] = \left[\begin{array}{cc:c} 0 & 0 & \tau I_q \\ \tau^{-1} P & 0 & P \\ \hdashline \tau^{-1} I_q & 0 & I_q \\ \tau^{-1} P & I_p & P \end{array}\right] = \left[\begin{array}{c|cc:c} A & B & 0 & \tau B \\ \hline 0 & 0 & 0 & \tau I_q \\ \tau^{-1} C & 0 & 0 & 0 \\ \hdashline 0 & \tau^{-1} I_q & 0 & I_q \\ \tau^{-1} C & 0 & I_p & 0 \end{array}\right]. \tag{B.1}$$

Denoting $\widetilde{B}_1 = \begin{bmatrix} B & 0 \end{bmatrix}$, $\widetilde{B}_2 = \tau B$, $\widetilde{C}_1 = \begin{bmatrix} 0 \\ \tau^{-1}C \end{bmatrix}$, $\widetilde{C}_2 = \begin{bmatrix} 0 \\ \tau^{-1}C \end{bmatrix}$, $\widetilde{D}_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $\widetilde{D}_{12} = \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix}$, $\widetilde{D}_{21} = \begin{bmatrix} \tau^{-1}I_q & 0 \\ 0 & I_p \end{bmatrix}$, and $\widetilde{D}_{22} = \begin{bmatrix} I_q \\ 0 \end{bmatrix}$, it can be seen that the simplified orthogonality assumptions in the $\mathcal{H}_\infty$ theory are not satisfied, that is

$$\widetilde{D}_{12}^* \begin{bmatrix} \widetilde{C}_1 & \widetilde{D}_{12} \end{bmatrix} = \begin{bmatrix} \tau I_q & 0 \end{bmatrix} \begin{bmatrix} 0 & \tau I_q \\ \tau^{-1}C & 0 \end{bmatrix} = \begin{bmatrix} 0 & \tau^2 I_q \end{bmatrix} \neq \begin{bmatrix} 0 & I_q \end{bmatrix},$$

and

$$\begin{bmatrix} \widetilde{B}_1 \\ \widetilde{D}_{21} \end{bmatrix} \widetilde{D}_{21}^* = \begin{bmatrix} B & 0 \\ \tau^{-1}I_q & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} \tau^{-1}I_q & 0 \\ 0 & I_p \end{bmatrix} = \begin{bmatrix} \tau^{-1}B & 0 \\ \tau^{-2}I_q & 0 \\ 0 & I_p \end{bmatrix} \neq \begin{bmatrix} 0 & 0 \\ I_q & 0 \\ 0 & I_p \end{bmatrix}$$

Normalizing $\widetilde{D}_{12} = \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix}$ and $\widetilde{D}_{21} = \begin{bmatrix} \tau^{-1}I_q & 0 \\ 0 & I_p \end{bmatrix}$ by writing them as

$$\widetilde{D}_{12} = U_p \begin{bmatrix} 0 \\ I_q \end{bmatrix} R_p = \begin{bmatrix} 0 & I_q \\ I_p & 0 \end{bmatrix} \begin{bmatrix} 0 \\ I_q \end{bmatrix} \tau I_q = \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix},$$

and

$$\widetilde{D}_{21} = \widetilde{R}_p \begin{bmatrix} I_p & 0 \\ 0 & I_q \end{bmatrix} \widetilde{U}_p = \begin{bmatrix} 0 & \tau^{-1}I_q \\ I_p & 0 \end{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & I_q \end{bmatrix} \begin{bmatrix} 0 & I_p \\ I_q & 0 \end{bmatrix} = \begin{bmatrix} \tau^{-1}I_q & 0 \\ 0 & I_p \end{bmatrix}$$

respectively, the new model for $\mathcal{H}_\infty$ synthesis becomes

$$
\begin{aligned}
\widehat{G} &= \left[\begin{array}{cc:cc} 0 & I_p & 0 & 0 \\ I_q & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & I_p \\ 0 & 0 & \tau I_q & 0 \end{array}\right] \widetilde{W}_l G \widetilde{W}_r^{-1} \left[\begin{array}{cc:c} 0 & I_q & 0 \\ I_p & 0 & 0 \\ \hdashline 0 & 0 & \tau^{-1} I_q \end{array}\right] \\
&= \left[\begin{array}{c|c:c} A & \begin{bmatrix} B & 0 \end{bmatrix} \begin{bmatrix} 0 & I_q \\ I_p & 0 \end{bmatrix} & \tau B \tau^{-1} I_q \\ \hline \begin{bmatrix} 0 & I_p \\ I_q & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \tau^{-1} C \end{bmatrix} & \begin{bmatrix} 0 & I_p \\ I_q & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & I_q \\ I_p & 0 \end{bmatrix} & \begin{bmatrix} 0 & I_p \\ I_q & 0 \end{bmatrix} \begin{bmatrix} \tau I_q \\ 0 \end{bmatrix} \tau^{-1} I_q \\ \hdashline \begin{bmatrix} 0 & I_p \\ \tau I_q & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \tau^{-1} C \end{bmatrix} & \begin{bmatrix} 0 & I_p \\ \tau I_q & 0 \end{bmatrix} \begin{bmatrix} \tau^{-1} I_q & 0 \\ 0 & I_p \end{bmatrix} \begin{bmatrix} 0 & I_q \\ I_p & 0 \end{bmatrix} & \begin{bmatrix} 0 & I_p \\ \tau I_q & 0 \end{bmatrix} \begin{bmatrix} I_q \\ 0 \end{bmatrix} \tau^{-1} I_q \end{array}\right] \\
&= \left[\begin{array}{c|cc:c} A & 0 & B & B \\ \hline \tau^{-1} C & 0 & 0 & 0 \\ 0 & 0 & 0 & I_q \\ \hdashline \tau^{-1} C & I_p & 0 & 0 \\ 0 & 0 & I_q & I_q \end{array}\right]
\end{aligned}
\tag{B.2}
$$

The controller for $\widehat{G}$, denoted $K_{\widehat{G}}$, is going to be given by

$$K_{\widehat{G}} = R_p K_G \widetilde{R}_p, \tag{B.3}$$

where K_G is the initial controller corresponding to G, $R_p = \tau I_q$, and $\widetilde{R}_p = \begin{bmatrix} 0 & \tau^{-1} I_q \\ I_p & 0 \end{bmatrix}$. This implies

$$K_G = R_p^{-1} K_{\widehat{G}} \widetilde{R}_p^{-1}, \tag{B.4}$$

where $R_p^{-1} = \tau^{-1} I_q$, and $\widetilde{R}_p^{-1} = \begin{bmatrix} 0 & I_p \\ \tau I_q & 0 \end{bmatrix}$. To simplify the controller design, let us first assume $\widetilde{D}_{22} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and, therefore,

$$\widehat{G}_0 = \left[\begin{array}{c|cc:c} A & 0 & B & B \\ \hline \tau^{-1}C & 0 & 0 & 0 \\ 0 & 0 & 0 & I_q \\ \hdashline \tau^{-1}C & I_p & 0 & 0 \\ 0 & 0 & I_q & 0 \end{array}\right], \tag{B.5}$$

for which the controller is denoted by $K_{\widehat{G}_0}$. Then, the controller for $\widehat{G}$ is going to be given by

$$K_{\widehat{G}} = K_{\widehat{G}_0} \left(I + \begin{bmatrix} I_q \\ 0 \end{bmatrix} K_{\widehat{G}_0} \right)^{-1}. \tag{B.6}$$

The new partitioned matrices for $\widehat{G}_0$ are $\widehat{B}_1 = \begin{bmatrix} 0 & B \end{bmatrix}$, $\widehat{B}_2 = B$, $\widehat{C}_1 = \begin{bmatrix} \tau^{-1}C \\ 0 \end{bmatrix}$, $\widehat{C}_2 = \begin{bmatrix} \tau^{-1}C \\ 0 \end{bmatrix}$, $\widehat{D}_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, $\widehat{D}_{12} = \begin{bmatrix} 0 \\ I_q \end{bmatrix}$, $\widehat{D}_{21} = \begin{bmatrix} I_p & 0 \\ 0 & I_q \end{bmatrix}$, and $\widehat{D}_{22} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. In this case, only one orthogonality condition is satisfied, that is

$$\widehat{D}_{12}^* \begin{bmatrix} \widehat{C}_1 & \widehat{D}_{12} \end{bmatrix} = \begin{bmatrix} 0 & I_q \end{bmatrix} \begin{bmatrix} \tau^{-1}C & 0 \\ 0 & I_q \end{bmatrix} = \begin{bmatrix} 0 & I_q \end{bmatrix},$$

while the other is still not met

$$\begin{bmatrix} \widehat{B}_1 \\ \widehat{D}_{21} \end{bmatrix} \widehat{D}_{21}^* = \begin{bmatrix} 0 & B \\ I_p & 0 \\ 0 & I_q \end{bmatrix} \begin{bmatrix} I_p & 0 \\ 0 & I_q \end{bmatrix} = \begin{bmatrix} 0 & B \\ I_p & 0 \\ 0 & I_q \end{bmatrix} \neq \begin{bmatrix} 0 & 0 \\ I_p & 0 \\ 0 & I_q \end{bmatrix}.$$

Once again, according to the solution to the discrete-time $\mathcal{H}_\infty$ control problem in [26], some preliminary matrices must be defined before proceeding with synthesizing the actual controller. First, write

$$D_{1\bullet} = \begin{bmatrix} \widehat{D}_{11} & \widehat{D}_{12} \end{bmatrix} = \left[\begin{array}{cc:c} 0 & 0 & 0 \\ 0 & 0 & I_q \end{array}\right] = \begin{bmatrix} 0_{p\times(p+q)} & 0 \\ 0 & I_q \end{bmatrix}, \tag{B.7}$$

$$D_{\bullet 1} = \begin{bmatrix} \widehat{D}_{11} \\ \widehat{D}_{21} \end{bmatrix} = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ \hdashline I_p & 0 \\ 0 & I_q \end{array}\right] = \begin{bmatrix} 0_{p+q} \\ I_{p+q} \end{bmatrix}. \tag{B.8}$$

and then, from equations (B.7) and (B.8) compute

$$R = D_{1\bullet}^* D_{1\bullet} - \begin{bmatrix} \gamma^2 I_{p+q} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -\gamma^2 I_{p+q} & 0 \\ 0 & I_q \end{bmatrix} \tag{B.9}$$

$$\widetilde{R} = D_{\bullet 1} D_{\bullet 1}^* - \begin{bmatrix} \gamma^2 I_{p+q} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -\gamma^2 I_{p+q} & 0 \\ 0 & I_{p+q} \end{bmatrix}. \tag{B.10}$$

The corresponding symplectic pairs needed to solve the $\mathcal{H}_\infty$ discrete-time control problem are shown below.

$$S_\infty := \left(\begin{bmatrix} A - \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix} R^{-1} D_{1\bullet}^* \widehat{C}_1 & 0 \\ -\widehat{C}_1^* \left(I - D_{1\bullet} R^{-1} D_{1\bullet}^* \right) \widehat{C}_1 & I \end{bmatrix}, \begin{bmatrix} I & \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix} R^{-1} \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix}^* \\ 0 & \left(A - \begin{bmatrix} \widehat{B}_1 & \widehat{B}_2 \end{bmatrix} R^{-1} D_{1\bullet}^* \widehat{C}_1 \right)^* \end{bmatrix} \right)$$

$$= \left(\begin{bmatrix} A & 0 \\ -\tau^{-2} C^* C & I \end{bmatrix}, \begin{bmatrix} I & (1-\gamma^{-2}) B B^* \\ 0 & A^* \end{bmatrix} \right) = (S_{\infty 1}, S_{\infty 2}) \tag{B.11}$$

and

$$T_\infty := \left(\begin{bmatrix} \left(A - \widehat{B}_1 D_{\bullet 1}^* \widetilde{R}^{-1} \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix} \right)^* & 0 \\ -\widehat{B}_1 \left(I - D_{\bullet 1}^* \widetilde{R}^{-1} D_{\bullet 1} \right) \widehat{B}_1^* & I \end{bmatrix}, \begin{bmatrix} I & \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix}^* \widetilde{R}^{-1} \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix} \\ 0 & A - \widehat{B}_1 D_{\bullet 1}^* \widetilde{R}^{-1} \begin{bmatrix} \widehat{C}_1 \\ \widehat{C}_2 \end{bmatrix} \end{bmatrix} \right)$$

$$= \left(\begin{bmatrix} A^* & 0 \\ 0 & I \end{bmatrix}, \begin{bmatrix} I & (1-\gamma^{-2}) \tau^{-2} C^* C \\ 0 & A \end{bmatrix} \right) = (T_{\infty 1}, T_{\infty 2}). \tag{B.12}$$

The DAREs corresponding to the symplectic pairs in equations (B.11) and (B.12) are therefore given by

$$X_\infty = A^* X_\infty \left[I + (1-\gamma^{-2}) B B^* X_\infty \right]^{-1} A + \tau^{-2} C^* C, \tag{B.13}$$

$$Y_\infty = A Y_\infty \left[I + (1-\gamma^{-2}) \tau^{-2} C^* C Y_\infty \right]^{-1} A^*. \tag{B.14}$$

After $K_{\widehat{G}_0}$ is synthesized, $K_{\widehat{G}}$ can be obtained from equation (B.6). The stabilizing controller K for the initial system given by G can then be obtained by using equation (B.4).

Appendix C

MATLAB Code

C.1 MATLAB Search Algorithm for Simulations in Section 3.3

```
function [K, CL, gamma, tau] = ControllerNormalizedSearch(G,
   BlkStruc, nmeas, ninp, w, epsilon)
% Block Structure BlkStruc is needed for the structured
   singular value
% function
screenSize = get(0, 'screensize');
fig = figure;
set(fig, 'Position', [screenSize(1)+20, screenSize(2)+60,...
    3.75*screenSize(3)/4, 3*screenSize(4)/4],...
    'PaperOrientation', 'landscape', 'PaperPositionMode', '
       auto');

% transform G into state-space, such that we can write the
   scaled function
% as Wl*G*Wr^{-1} = Wl*(C*(sI-A)^{-1}*B + D)*Wr^{-1} =
% Wl*C*(sI-A)^{-1}*B*Wr^{-1} + Wl* D * Wr^{-1}
% getting the total number of inputs and outputs of the system
G_nout = size(G, 1);
G_ninp = size(G, 2);

eps_min = epsilon(1);
eps_max = epsilon(end);
n_eps = length(epsilon);
eps_step = (eps_max - eps_min)/(n_eps - 1);

Kiter = 1;
redo = 1;
lorr = logical(1); % to the LEFT or RIGHT of 1; 1 for LEFT, 0
   for RIGHT

while redo == 1
    for idx = 1:length(epsilon)
        % scaling matrices
        % this will only work for the case of 2 scalar
           uncertainties blocks
```

```
30          % Delta = diagonal{Delta1, Delta_2}
31
32          % the search for tau will be done by searching an
               epsilon in interval (0,1), and
33          % therefore we switch between 2 cases of scaling
               factors to include
34          % the cases when tau should actually be greater than 1
35
36          % we start searching in (0, 1)
37          if lorr
38              % case tau < 1
39              tau_e(idx) = epsilon(idx);
40              Wl = diag([epsilon(idx), ones(1, G_nout - 1 -
                   nmeas)]);
41              Wr = diag([epsilon(idx), ones(1, G_ninp - 1 - ninp
                   )]);
42              % augmented scaling matrices
43              Wl_a = diag([epsilon(idx), ones(1, G_nout -1)]);
44              Wr_a = diag([epsilon(idx), ones(1, G_ninp -1)]);
45              lefttitle = ['$K$ iteration \# ', num2str(Iter, '
                   %d'),...
46                '', $D$ iteration \# '',...
47                num2str(idx, '%d'), '', with $\varepsilon = '',
                     num2str(epsilon(idx), '%.4f'),...
48                '$, ($\tau = '', num2str(epsilon(idx), '%.4f'),
                       '$)''];
49          else
50              % case tau > 1
51              tau_e(idx) = 1/epsilon(idx);
52              Wl = diag([1/epsilon(idx), ones(1, G_nout - 1 -
                   nmeas)]);
53              Wr = diag([1/epsilon(idx), ones(1, G_ninp - 1 -
                   ninp)]);
54              % augmented scaling matrices
55              Wl_a = diag([1/epsilon(idx), ones(1, G_nout -1)]);
56              Wr_a = diag([1/epsilon(idx), ones(1, G_ninp -1)]);
57              lefttitle = ['$K$ iteration \# ', num2str(Iter, '
                   %d'),...
58                '', $D$ iteration \# '',...
59                num2str(idx, '%d'), '', with $\varepsilon = '',
                     num2str(epsilon(idx), '%.4f'),...
60                '$, ($\tau = '', num2str(1/epsilon(idx), '%.4f'
                     ), '$)''];
61          end
62
```

```
63          G_a.a = G.a;
64          G_a.b = G.b/Wr_a;
65          G_a.c = Wl_a*G.c;
66          G_a.d = Wl_a*G.d/Wr_a;
67          G_a_ss = ss(G_a.a, G_a.b, G_a.c, G_a.d);
68          [K{idx}, CL{idx}, gamma(idx), info] = hinfsyn(G_a_ss,
               nmeas, ninp, 'Method', 'ric', 'Display', 'off');
69          % returned K and CL are in state-space
70
71          % scaled closed-loop model, norm, FRD model, and
               singular values
72          CLSS_scaled = CL{idx}; %minreal(ss(Wl*tf(lft(G, tf(K{
               idx}))))/Wr));
73          CL_scaled_norms(idx) = norm(CLSS_scaled, inf);
74
75          % frequency response model of the scaled closed-loop
76          CLf_scaled = frd(CLSS_scaled, w);
77          SV_scaled = sigma(CLf_scaled, w);
78          % unscaled closed-loop model, norm, FRD model, and
               singular values
79          CLSS = lft(G, K{idx});
80          CL_norms(idx) = norm(CLSS, inf);
81          CLf = frd(CLSS, w);
82          SV = sigma(CLf, w);
83          % structured singular values bounds; these are
               computed for
84          % unscaled closed loop
85          mubnds = mussv(CLf, BlkStruc);
86          % maximum lower bound
87          [maxmubndl, mubndlidx] = max(mubnds(1,2).ResponseData
               (:));
88          % maximum upper bound
89          [maxmubndu, mubnduidx] = max(mubnds(1,1).ResponseData
               (:));
90
91          % for stability, we should just look at the proper
               block in the
92          % closed-loop model, that is CL_11
93          % the following code only applies in this case, since
               I have
94          % Delta_1 and Delta_2 as 1 by 1 blocks
95          CL_scaled_11 = CLSS_scaled(1,1);
96          CL_scaled_11_norm(idx) = norm(CL_scaled_11, inf);
97
98          figure(fig);
```

```
99          subplot(1, 2, 1);
100         g1 = gca;
101         set(g1, 'Position', [0.05, 0.05, 0.4, 0.8]);
102         semilogx(w, squeeze(mubnds.ResponseData), 'Color', '
               green', 'LineWidth', 3)
103         hold on
104         % semilogx(w, CL_norm(idx)*ones(size(w)), 'Color', '
               yellow', 'LineWidth', 2.5, 'LineStyle', '-')
105         semilogx(w, CL_scaled_norms(idx)*ones(size(w)), 'Color
               ', 'red', 'LineWidth', 1.5, 'LineStyle', '-')
106         semilogx(w, gamma(idx)*ones(size(w)), 'Color', 'black'
               , 'LineWidth', 1, 'LineStyle', '-.')
107         semilogx(w, SV_scaled, 'Color', 'magenta', 'LineWidth'
               , 1.5, 'LineStyle', '-.')
108
109         legend('upper mussv', 'lower mussv',...
110             'inf norm (scaled CL)', 'gamma (hinfsyn)', ...
111             'sv (Scaled FRD)', 'sv (Scaled FRD)')
112         title(lefttitle, 'FontWeight', 'bold', 'FontSize', 12)
113         hold off
114     end
115     % find the tau that gives the minimum norm
116     [min_scaled_norm, minidx] = min(CL_scaled_norms);
117
118     figure(fig);
119     subplot(1, 2, 2);
120     g2 = gca;
121     set(g2, 'Position', [0.55, 0.05, 0.4, 0.8]);
122     stem(epsilon, gamma, 'o', 'MarkerSize', 8,...
123         'MarkerEdgecolor', 'black', 'MarkerfaceColor', 'white'
               )
124     hold on
125     stem(epsilon, CL_scaled_norms, 'o', 'MarkerSize', 6,...
126         'MarkerEdgecolor', 'black', 'MarkerfaceColor', 'green'
               )
127     plot(epsilon(minidx), min_scaled_norm, 's', 'MarkerSize',
           4,...
128         'MarkerEdgeColor', 'black', 'MarkerFaceColor', 'red')
129     if lorr
130         righttitle = {['$K$ it. \# ', num2str(Kiter, '%d'),...
131                 ', $\|W_{l}\mathcal{F}(G,K)W_{r}^{-1}\|_{\
                       infty} = ',...
132                 num2str(min_scaled_norm, '%.4f'),...
133                 '$ ($\| \Delta \|_{\infty} < ', num2str(1/
                       min_scaled_norm, '%.4f'), '$)'];
```

```
134                 ['$\| M_{11} \|_{\infty} = ',...
135                 num2str(CL_scaled_11_norm(minidx), '%.4f'),...
136                 '$ ($\| \Delta_{1} \|_{\infty} < ',...
137                 num2str(1/CL_scaled_11_norm(minidx), '%.4f')
                        ,...
138                 '$) for $\varepsilon = ', num2str(epsilon(
                        minidx), '%.4f'),...
139                 '$, ($\tau = ', num2str(epsilon(minidx), '%.4f
                        '), '$)']};
140     else
141         righttitle = {['$K$ it. \# ', num2str(Kiter, '%d'),...
142                 ', $\|W_{l}\mathcal{F}(G,K)W_{r}^{-1}\|_{\
                        infty} = ',...
143                 num2str(min_scaled_norm, '%.4f'),...
144                 '$ ($\| \Delta \|_{\infty} < ', num2str(1/
                        min_scaled_norm, '%.4f'), '$)'];
145                 ['$\| M_{11} \|_{\infty} = ',...
146                 num2str(CL_scaled_11_norm(minidx), '%.4f'),...
147                 '$ ($\| \Delta_{1} \|_{\infty} < ',...
148                 num2str(1/CL_scaled_11_norm(minidx), '%.4f')
                        ,...
149                 '$) for $\varepsilon = ', num2str(epsilon(
                        minidx), '%.4f'),...
150                 '$, ($\tau = ', num2str(1/epsilon(minidx), '
                        %.4f'), '$)']};
151     end
152     title(righttitle, 'FontWeight', 'bold', 'FontSize', 12)
153     hold off
154
155     if epsilon(minidx) == epsilon(1)
156         redo = 0;
157         K = K{minidx};
158         CL = CL{minidx};
159         gamma = gamma(minidx);
160         tau = epsilon(minidx);
161     else if epsilon(minidx) == epsilon(end)
162             lorr = ~lorr;
163             redo = 1;
164         else if Kiter == 1
165                 eps_norm_min_old = epsilon(minidx);
166                 eps_min = epsilon(minidx) - eps_step;
167                 eps_max = epsilon(minidx) + eps_step;
168                 eps_step = (eps_max - eps_min)/(n_eps - 1);
169                 epsilon = eps_min:eps_step:eps_max;
170                 redo = 1;
```

```
                Kiter = Kiter + 1;
            else if floor(100*abs(epsilon(minidx) -
               eps_norm_min_old))/100 <= 0.01
                    redo = 0;
                    K = K{minidx};
                    CL = CL{minidx};
                    gamma = gamma(minidx);
                    tau = epsilon(minidx);
                else
                    eps_norm_min_old = epsilon(minidx);
                    eps_min = epsilon(minidx) - eps_step;
                    eps_max = epsilon(minidx) + eps_step;
                    eps_step = (eps_max - eps_min)/(n_eps - 1)
                       ;
                    epsilon = eps_min:eps_step:eps_max;
                    redo = 1;
                    Kiter = Kiter + 1;
                end
            end
        end
    end
end
```

C.2 MATLAB Main File for Simulations in Section 3.3

```
% in the file, since the uncertainties coming from
   quantization are
% time-variant, dksyn function won't do it; so I have to solve
    the Hinfity
% problem all by myself; let's see if I am gonna get any good
   results
clear all
close all
clc
format short g
format compact

screenSize = get(0, 'screensize');
set(0,'DefaultTextInterpreter','LaTex');

gain = 1;
Ts = 0.01;
fignum = 1;
```

```
% Discrete-time plant TF model
% P = [zpk([], [2], gain, Ts)]
% P = [zpk([], [0, 2], gain, Ts)]
% P = [zpk([], [0, 1.4], gain, Ts)]
% P = [zpk([3], [0, 2], gain, Ts)]
% P = [zpk([2], [0, 1.6], gain, Ts)]
% P = [zpk([1.5], [0, 1.7], gain, Ts)]
% P = [zpk([], [1.1], gain, Ts)]
% P = [zpk([], [0.8], gain, Ts)]
% P = [zpk([0.8], [0, 0.3], gain, Ts)]
P = [zpk([0.3], [0, 0.1], gain, Ts)]
% discrete-time state-space model needed for the Simulink run
Pss = ss(P);
% continuous-time plant TF model
Pc = d2c(P, 'tustin')
% continuous-time plant SS model
PcSS = minreal(ss(Pc))

% titlePP = 'For plant $\displaystyle P(z) = \frac{1}{z-2}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{1}{z(z-2)}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{1}{z(z-1.4)}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{z-3}{z(z-2)}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{z-2}{z(z-1.6)}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{z-1.5}{z(z-1.7)}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{1}{z-1.1}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{1}{z-0.8}$';
% titlePP = 'For plant $\displaystyle P(z) = \frac{z-0.8}{z(z-0.3)}$';
titlePP = 'For plant $\displaystyle P(z) = \frac{z-0.3}{z(z-0.1)}$';

textP = evalc('P');
ix1 = strfind(textP,':');
ix2 = strfind(textP,')');
titleP = textP(ix1(1)+2:ix2(end));

w = logspace(-2,10,1000);
tau_start = 0.01;
```

```
52 tau_end = 1;
53 % number of tau's for which we do the search; basically, it is
       the length
54 % of tau vector
55 n_tau = 10;
56 tau_step = (tau_end - tau_start)/(n_tau - 1);
57 init_tau = tau_start:tau_step:tau_end;
58
59 % signals used in the structures, that are going to be used by
       iconnect and
60 % isignal to create the model
61 d1 = icsignal(1);
62 d2 = icsignal(1);
63 v = icsignal(1);
64 u = icsignal(1);
65 y = icsignal(1);
66 yq = icsignal(1);
67
68 % the LFT TF
69 % for the case with input and output quantization, both
       multiplicative,
70 % but just simple output feedback
71
72 G1 = iconnect;
73 G1.Input = [d1; d2; v];
74 G1.Output = [v; y; yq];
75 G1.Equation{1} = equate(u, d1+v);
76 G1.Equation{2} = equate(y, PcSS*u);
77 % PcSS needs to be in state-space such the state-space for G1c
       is going to
78 % have the matrices accordingly... G1c.a = PcSS.a, and so on
       ...
79 G1.Equation{3} = equate(yq, d2+y);
80 G1c = ss(G1.System);
81
82 nmeas1 = 1;
83 ninp1 = 1;
84
85 BlckStruct1 = [1, 1; 1, 1];
86
87 [K1, CL1, gamma1, tau1] = ControllerNormalizedSearch(G1c,
       BlckStruct1, nmeas1, ninp1, w, init_tau);
88 fignum = fignum + 1;
89 gamma1
90 tau1
```

```
eig(CL1.a)
pole(CL1)
K1d = c2d(K1, Ts, 'tustin');

set(gcf,'PaperOrientation', 'landscape', 'PaperPositionMode', 'auto',...
    'PaperUnits', 'normalized', 'PaperType', 'uslegal',...
    'PaperPosition', [0.05, 0.05, 0.9, 0.9]);
ax = axes('Position',[0,0,1,1],'Visible','off');
MasterTitle = text(0.4,0.95, titlePP);
set(MasterTitle,'FontWeight','bold', 'FontSize', 12)

ftime = clock;
filename = char(['./Figs/', datestr(clock,'yyyy-mm-dd_HH-MM-SS'), '_SearchController_1']);
print('-dpdf', '-r150', filename);

% the LFT TF
% for the case with input and output quantization, both multiplicative,
% but output feedback augmented with feedback from quantized input as
% well

G2 = iconnect;
G2.Input = [d1; d2; v];
G2.Output = [v; y; u; yq];
G2.Equation{1} = equate(u, d1+v);
G2.Equation{2} = equate(y, PcSS*u);
G2.Equation{3} = equate(yq, d2+y);
G2c = ss(G2.System)

nmeas2 = 2;
ninp2 = 1;

BlckStruct2 = [1, 1; 1, 1];

[K2, CL2, gamma2, tau2] = ControllerNormalizedSearch(G2c, BlckStruct2, nmeas2, ninp2, w, init_tau);
fignum = fignum + 1;
gamma2
tau2
eig(CL2.a)
pole(CL2)
K2d = c2d(K2, Ts, 'tustin');
```

```
131
132 set(gcf,'PaperOrientation', 'landscape', 'PaperPositionMode',
      'auto',...
133     'PaperUnits', 'normalized', 'PaperType', 'uslegal',...
134     'PaperPosition', [0.05, 0.05, 0.9, 0.9]);
135 ax = axes('Position',[0,0,1,1],'Visible','off');
136 MasterTitle = text(0.4,0.95, titlePP);
137 set(MasterTitle,'FontWeight','bold', 'FontSize', 12)
138
139 ftime = clock;
140 filename = char(['./Figs/', datestr(clock,'yyyy-mm-dd_HH-MM-SS
      '), '_SearchController_2']);
141 print('-dpdf', '-r150', filename);
142
143
144 %%
145 CLL1 = lft(G1c, K1);
146 delta1_11 = 1/norm(CLL1(1,1), inf)
147 CLL2 = lft(G2c, K2);
148 delta2_11 = 1/norm(CLL2(1,1), inf)
149
150 %%
151 % Run Simulink model
152 % for positive uncertainty interval only
153 Delta_min = 1/gamma1;
154 Delta_max = 1/gamma2;
155
156 curr_time = clock;
157 seed_1 = abs(curr_time(end) + 1);
158 seed_2 = abs(curr_time(end) - 1);
159
160 set_param('my_usim_model_both','AlgebraicLoopSolver','
      LineSearch')
161 set_param('my_usim_model_both/D1', 'Minimum', num2str(
      Delta_min,'%.5f'))
162 set_param('my_usim_model_both/D1', 'Maximum', num2str(
      Delta_max,'%.5f'))
163 set_param('my_usim_model_both/D1', 'Seed', num2str(seed_1,'%.2
      f'))
164 set_param('my_usim_model_both/D2', 'Minimum', num2str(
      Delta_min,'%.5f'))
165 set_param('my_usim_model_both/D2', 'Maximum', num2str(
      Delta_max,'%.5f'))
166 set_param('my_usim_model_both/D2', 'Seed', num2str(seed_2,'%.2
      f'))
```

```
167
168 sim('my_usim_model_both')
169 %sim('my_usim_model_both_1')
170
171 deltaSignals = [delta1.signals.values, delta2.signals.values];
172 Delta_norm_max = max(max(deltaSignals(:,1)), max(deltaSignals
    (:,2)));
173 Delta_norm_min = min(min(deltaSignals(:,1)), min(deltaSignals
    (:,2)));
174
175 fig4 = figure(fignum);
176 fignum = fignum + 1;
177 set(fig4, 'Position', [screenSize(1)+20, screenSize(2)+60,...
178     3.5*screenSize(3)/4, 3*screenSize(4)/4],...
179     'PaperOrientation', 'landscape', 'PaperPositionMode', '
        auto');
180 subplot(2,2,1:2)
181 stairs(delta1.time, delta1.signals.values, 'LineWidth', 1.5, '
    Color', [0, 0.8, 0.2]);
182 hold on;
183 stairs(delta2.time, delta2.signals.values, 'LineWidth', 1.5, '
    Color', [0.5, 0, 0.8]);
184 %line([delta1.time(1), delta1.time(end)],[delta1_11, delta1_11
    ], 'LineWidth', 1, ...
185 %    'LineStyle', '-.', 'Color', 'black')
186 %line([delta1.time(1), delta1.time(end)],[delta2_11, delta2_11
    ], 'LineWidth', 1, ...
187 %    'LineStyle', '-.', 'Color', 'black')
188 hold off;
189 title(['Input and output uncertainty values ($', num2str(
    Delta_norm_min,'%.3f'),...
190     ' \leq \Delta_{i} \leq ', num2str(Delta_norm_max, '%.3f')
        ,...
191     ', \; i = 1,2$)'], 'FontSize', 14, 'FontWeight', 'bold'
        ,...
192     'Interpreter', 'Latex')
193 legend('Input uncertainty', 'Output uncertainty')
194 subplot(2,2,3)
195 stairs(y1.time, y1.signals.values, 'LineWidth', 1.5, 'Color',
    'red')
196 title(['Output (simple output feedback)'], 'FontSize', 14, '
    FontWeight', 'bold',...
197     'Interpreter', 'Latex')
198 subplot(2,2,4)
```

```
199 stairs(y2.time, y2.signals.values, 'LineWidth', 1.5, 'Color',
    'blue')
200 title(['Output (input-output feedback)'], 'FontSize', 14, '
    FontWeight', 'bold',...
201     'Interpreter', 'Latex')
202
203 set(gcf,'PaperOrientation', 'landscape', 'PaperPositionMode',
    'auto',...
204     'PaperUnits', 'normalized', 'PaperType', 'uslegal',...
205     'PaperPosition', [0.05, 0.05, 0.9, 0.9]);
206 ax = axes('Position',[0,0,1,1],'Visible','off');
207 MasterTitle = text(0.4,0.05, titlePP);
208 set(MasterTitle,'FontWeight','bold', 'FontSize', 14)
209
210 ftime = clock;
211 filename = char(['./Figs/', datestr(clock,'yyyy-mm-dd_HH-MM-SS
    '), '_Pos_Simulink_Run']);
212 print('-dpdf', '-r150', filename);
213
214 set_param('my_usim_model_both/D1', 'Maximum', '1')
215 set_param('my_usim_model_both/D1', 'Minimum', '0')
216 set_param('my_usim_model_both/D2', 'Maximum', '1')
217 set_param('my_usim_model_both/D2', 'Minimum', '0')
218
219 %%
220 % Run Simulink model
221 % for negative uncertainty interval only
222 Delta_min = -1/gamma2;
223 Delta_max = -1/gamma1;
224
225 curr_time = clock;
226 seed_1 = abs(curr_time(end) + 1);
227 seed_2 = abs(curr_time(end) - 1);
228
229 set_param('my_usim_model_both','AlgebraicLoopSolver','
    LineSearch')
230 set_param('my_usim_model_both/D1', 'Minimum', num2str(
    Delta_min,'%.5f'))
231 set_param('my_usim_model_both/D1', 'Maximum', num2str(
    Delta_max,'%.5f'))
232 set_param('my_usim_model_both/D1', 'Seed', num2str(seed_1,'%.2
    f'))
233 set_param('my_usim_model_both/D2', 'Minimum', num2str(
    Delta_min,'%.5f'))
```

```
234 set_param('my_usim_model_both/D2', 'Maximum', num2str(
      Delta_max,'%.5f'))
235 set_param('my_usim_model_both/D2', 'Seed', num2str(seed_2,'%.2
      f'))
236
237 sim('my_usim_model_both')
238 %sim('my_usim_model_both_1')
239
240 deltaSignals = [delta1.signals.values, delta2.signals.values];
241 Delta_norm_max = max(max(deltaSignals(:,1)), max(deltaSignals
      (:,2)));
242 Delta_norm_min = min(min(deltaSignals(:,1)), min(deltaSignals
      (:,2)));
243
244 fig4 = figure(fignum);
245 fignum = fignum + 1;
246 set(fig4, 'Position', [screenSize(1)+20, screenSize(2)+60,...
247     3.5*screenSize(3)/4, 3*screenSize(4)/4],...
248     'PaperOrientation', 'landscape', 'PaperPositionMode', '
        auto');
249 subplot(2,2,1:2)
250 stairs(delta1.time, delta1.signals.values, 'LineWidth', 1.5, '
      Color', [0, 0.8, 0.2]);
251 hold on;
252 stairs(delta2.time, delta2.signals.values, 'LineWidth', 1.5, '
      Color', [0.5, 0, 0.8]);
253 %line([delta1.time(1), delta1.time(end)],[delta1_11, delta1_11
      ], 'LineWidth', 1, ...
254 %    'LineStyle', '-.', 'Color', 'black')
255 %line([delta1.time(1), delta1.time(end)],[delta2_11, delta2_11
      ], 'LineWidth', 1, ...
256 %    'LineStyle', '-.', 'Color', 'black')
257 hold off;
258 title(['Input and output uncertainty values ($', num2str(
      Delta_norm_min,'%.3f'),...
259     ' \leq \Delta_{i} \leq ', num2str(Delta_norm_max, '%.3f')
        ,...
260     ', \; i = 1,2$)'], 'FontSize', 14, 'FontWeight', 'bold'
        ,...
261     'Interpreter', 'Latex')
262 legend('Input uncertainty', 'Output uncertainty')
263 subplot(2,2,3)
264 stairs(y1.time, y1.signals.values, 'LineWidth', 1.5, 'Color',
      'red')
```

```
265 title(['Output (simple output feedback)'], 'FontSize', 14, '
    FontWeight', 'bold',...
266     'Interpreter', 'Latex')
267 subplot(2,2,4)
268 stairs(y2.time, y2.signals.values, 'LineWidth', 1.5, 'Color',
    'blue')
269 title(['Output (input-output feedback)'], 'FontSize', 14, '
    FontWeight', 'bold',...
270     'Interpreter', 'Latex')
271
272 set(gcf,'PaperOrientation', 'landscape', 'PaperPositionMode',
    'auto',...
273     'PaperUnits', 'normalized', 'PaperType', 'uslegal',...
274     'PaperPosition', [0.05, 0.05, 0.9, 0.9]);
275 ax = axes('Position',[0,0,1,1],'Visible','off');
276 MasterTitle = text(0.4,0.05, titlePP);
277 set(MasterTitle,'FontWeight','bold', 'FontSize', 14)
278
279 ftime = clock;
280 filename = char(['./Figs/', datestr(clock,'yyyy-mm-dd_HH-MM-SS
    '), '_Neg_Simulink_Run']);
281 print('-dpdf', '-r150', filename);
282
283 set_param('my_usim_model_both/D1', 'Maximum', '1')
284 set_param('my_usim_model_both/D1', 'Minimum', '0')
285 set_param('my_usim_model_both/D2', 'Maximum', '1')
286 set_param('my_usim_model_both/D2', 'Minimum', '0')
287
288 %%
289 % Run Simulink model
290 % for whole uncertainty interval
291 Delta_min = -1/gamma2;
292 Delta_max = 1/gamma2;
293
294 curr_time = clock;
295 seed_1 = abs(curr_time(end) + 1);
296 seed_2 = abs(curr_time(end) - 1);
297
298 set_param('my_usim_model_both','AlgebraicLoopSolver','
    LineSearch')
299 set_param('my_usim_model_both/D1', 'Minimum', num2str(
    Delta_min,'%.5f'))
300 set_param('my_usim_model_both/D1', 'Maximum', num2str(
    Delta_max,'%.5f'))
```

```
301 set_param('my_usim_model_both/D1', 'Seed', num2str(seed_1,'%.2
    f'))
302 set_param('my_usim_model_both/D2', 'Minimum', num2str(
    Delta_min,'%.5f'))
303 set_param('my_usim_model_both/D2', 'Maximum', num2str(
    Delta_max,'%.5f'))
304 set_param('my_usim_model_both/D2', 'Seed', num2str(seed_2,'%.2
    f'))
305
306 sim('my_usim_model_both')
307 %sim('my_usim_model_both_1')
308
309 deltaSignals = [delta1.signals.values, delta2.signals.values];
310 Delta_norm_max = max(max(deltaSignals(:,1)), max(deltaSignals
    (:,2)));
311 Delta_norm_min = min(min(deltaSignals(:,1)), min(deltaSignals
    (:,2)));
312
313 fig4 = figure(fignum);
314 fignum = fignum + 1;
315 set(fig4, 'Position', [screenSize(1)+20, screenSize(2)+60,...
316     3.5*screenSize(3)/4, 3*screenSize(4)/4],...
317     'PaperOrientation', 'landscape', 'PaperPositionMode', '
        auto');
318 subplot(2,2,1:2)
319 stairs(delta1.time, delta1.signals.values, 'LineWidth', 1.5, '
    Color', [0, 0.8, 0.2]);
320 hold on;
321 stairs(delta2.time, delta2.signals.values, 'LineWidth', 1.5, '
    Color', [0.5, 0, 0.8]);
322 line([delta1.time(1), delta1.time(end)],[1/gamma1, 1/gamma1],
    'LineWidth', 1, ...
323     'LineStyle', '-.', 'Color', 'black')
324 line([delta1.time(1), delta1.time(end)],[-1/gamma1, -1/gamma1
    ], 'LineWidth', 1, ...
325     'LineStyle', '-.', 'Color', 'black')
326 hold off;
327 title(['Input and output uncertainty values ($', num2str(
    Delta_norm_min,'%.3f'),...
328     ' \leq \Delta_{i} \leq ', num2str(Delta_norm_max, '%.3f')
        ,...
329     ', \; i = 1,2$)'], 'FontSize', 14, 'FontWeight', 'bold'
        ,...
330     'Interpreter', 'Latex')
331 legend('Input uncertainty', 'Output uncertainty')
```

```
subplot(2,2,3)
stairs(y1.time, y1.signals.values, 'LineWidth', 1.5, 'Color', 'red')
title(['Output (simple output feedback)'], 'FontSize', 14, 'FontWeight', 'bold',...
    'Interpreter', 'Latex')
subplot(2,2,4)
stairs(y2.time, y2.signals.values, 'LineWidth', 1.5, 'Color', 'blue')
title(['Output (input-output feedback)'], 'FontSize', 14, 'FontWeight', 'bold',...
    'Interpreter', 'Latex')

set(gcf,'PaperOrientation', 'landscape', 'PaperPositionMode', 'auto',...
    'PaperUnits', 'normalized', 'PaperType', 'uslegal',...
    'PaperPosition', [0.05, 0.05, 0.9, 0.9]);
ax = axes('Position',[0,0,1,1],'Visible','off');
MasterTitle = text(0.4,0.05, titlePP);
set(MasterTitle,'FontWeight','bold', 'FontSize', 14)

ftime = clock;
filename = char(['./Figs/', datestr(clock,'yyyy-mm-dd_HH-MM-SS'), '_Both_Simulink_Run']);
print('-dpdf', '-r150', filename);

set_param('my_usim_model_both/D1', 'Maximum', '1')
set_param('my_usim_model_both/D1', 'Minimum', '0')
set_param('my_usim_model_both/D2', 'Maximum', '1')
set_param('my_usim_model_both/D2', 'Minimum', '0')
```

C.3 MATLAB Main File for Example 4.3.1

```
clc
clear all
close all
format short g
format compact

screenSize = get(0, 'screensize');
set(0,'DefaultTextInterpreter','Latex')

A = [1.5, 0.2, 0;
     -1.2, 0, 1.8;
     1.1, 0, 0];
```

```
B = [1.2;
     -1.5;
     0.4];

eig_A = eig(A); % eigenvalues of A
abs_eig_A = abs(eig_A);
fprintf('The eigen values of A are: ')
for eig_ind = 1:length(eig_A)
    fprintf([num2str(eig_A(eig_ind)), ',\t']);
end
fprintf('\n with absolute values: ')
for eig_ind = 1:length(eig_A)
    fprintf([num2str(abs_eig_A(eig_ind)), ',\t']);
end
fprintf('\n')
% find the number of unstable eigenvalues of A, meaning the
   ones that are
% outside the unit circle, since we are dealing with discrete-
   time systems
ue_num = length(find(abs(eig(A)) >= 1));
% find the number of stable eigenvalues of A, meaning the ones
    that are
% inside the unit circle, since we are dealing with discrete-
   time systems
se_num = size(A, 1) - ue_num;
% in this case, though, since A has all eigenvalues outside
   the unit
% circle, I am not going to bother about it and just think Au
   = A and Bu =
% B

% calculating the mahler measure according to definition
mu_A = 1; % initialize Mahler measure
for i = 1:length(eig_A)
    mu_A = mu_A * max(abs(eig_A(i)), 1);
end
fprintf('Mahler measure mu_A = %.4f \n', mu_A)
fprintf('1/mu_A = %.4f \n', 1/mu_A)
fprintf('We need delta < 1/mu_A \n')
%delta = 0.09;
%fprintf('delta = %.4f \n', delta)
% eig_ratio = (1 + delta)/(1 - delta);
eig_ratio_ub = (mu_A + 1)/(mu_A - 1);
fprintf('Upper bound for lambda_N/lambda_2 is (mu_A + 1)/(mu_A
    - 1) = %.4f \n', eig_ratio_ub)
```

```
eig_ratio = 1.2;
fprintf('eigen ratio lambda_N/lambda_2 = %.4f < %.4f\n',
   eig_ratio, eig_ratio_ub)
% upper bound for theta, with the chosen eigen ratio
theta_ub = asind(sqrt((eig_ratio+1)^2/(4*eig_ratio*mu_A^2)
   -...
   (eig_ratio-1)^2/(4*eig_ratio)));
fprintf('upper bound for theta = %.4f degrees \n', theta_ub)
% pick theta
theta = 10; % degrees
fprintf('pick theta = %.4f < %.4f \n', theta, theta_ub)
thetar = theta*pi/180; % radians
% pick center at r0 = 1
r0 = 1;
fprintf('Center of the fan region is chosen at r0 = %.2f.\n',
   r0)
% solve to get rN and r2
% rNr2 = [1, 1; 1, -eig_ratio]\[2*r0*cosd(theta); 0];
rNr2 = inv([1, 1; 1, -eig_ratio])*[2*r0*cosd(theta); 0];
rN = rNr2(1);
r2 = rNr2(2);
fprintf('It results, rN = %.4f and r2 = %.4f,\n that have
   verified ratio %.4f.\n',...
    rN, r2, rN/r2)

% eigen values for the Laplacian
lambda_1 = 0;
lambda_2 = r2*exp(j*thetar);
lambda_3 = conj(lambda_2);
lambda_4 = rN*exp(j*thetar);
%lambda_4 = abs(rN*exp(j*thetar));
lambda_5 = conj(lambda_4);
lambdas = [lambda_1, lambda_4, lambda_2, lambda_3, lambda_5];
%lambdas = [lambda_1, lambda_2, lambda_4, lambda_3]
fprintf('Eigenvalues for the Laplacian given by the above
   conditions:\n')
for eig_ind = 1:length(lambdas)
    fprintf([num2str(lambdas(eig_ind)), ',\t']);
end
fprintf('with absolute values:\n')
for eig_ind = 1:length(lambdas)
    fprintf([num2str(abs(lambdas(eig_ind))), ',\t']);
end
fprintf('\n')

```

```
91 fprintf('Checking to see that the radius (delta) condition is
     satisfied.\n')
92 fprintf('Distance from lambda_2 to center r0 -> %.4f\n', abs(
     lambda_2 - r0)/abs(r0));
93 fprintf('Distance from lambda_4 to center r0 -> %.4f\n', abs(
     lambda_4 - r0)/abs(r0));
94 delta = abs(lambda_4 - r0)/abs(r0);
95 fprintf('delta = %.4f < 1/mu_A = %.4f \n', delta, 1/mu_A)
96
97 % first row of L = invers DFT of eigenvalue vector
98 row1_L = ifft(lambdas); % this should not have complex
     elements
99 fprintf('Laplacian is built using IDFT of the eigen values:\n'
     )
100 Lcal = toeplitz([row1_L(1), row1_L(end:-1:2)], row1_L)
101 Dcal = diag(diag(Lcal));
102 fprintf('Weighted adjacency matrix:\n')
103 Acal = Dcal - Lcal
104 fprintf('Pick gamma > gamma_opt = mu_A, to design the
     controller.\n')
105 gamma = 1/delta;
106 fprintf('So, pick gamma = 1/delta = %.4f. \n', gamma)
107 % design controller by solving the DARE
108 Q = zeros(size(A, 1));
109 R = eye(size(B, 2));
110 fprintf('Solution X of DARE:\n')
111 [X, L, G] = dare(A, sqrt(1 - gamma^(-2))*B, Q, R);
112 X
113 fprintf('Controller:\n')
114 K = (eye(size(B, 2)) + (1 - gamma^(-2))*B'*X*B)\B'*X*A
115 break
116 % number of agents
117 N = 5;
118
119 % abs(eig(kron(eye(N), A) - kron(Lcal, B*K)))
120
121 % starting simulations
122 % time vector, which in this case is the same with the sample
     numbers
123 t = 0:1:30;
124 % initial condition for states
125 x0_min = -5;
126 x0_max = 5;
127 for ag_ind = 1:N
```

```
128     x(:, 1, ag_ind) = x0_min + (x0_max - x0_min).*rand(size(A
            ,1), 1);
129 end
130
131 % create input vector
132 u = zeros(size(B, 2), length(t), N);
133 for k = 1:length(t)-1
134     for i = 1:N
135         for j = 1:N
136             u(:, k, i) = u(:, k, i) + K * (-Lcal(i,j))* x(:, k
                    , j);
137         end
138         x(:, k+1, i) = A * x(:, k, i) + B * u(:, k, i);
139     end
140 end
141
142 % getting the average values for states
143 x_sum = zeros(size(A, 1), length(t));
144 for j = 1:length(t)
145     for i = 1:N
146         x_sum (:, j) = x_sum(:, j) + x(:, j, i);
147     end
148     x_avg(:, j) = x_sum(:, j)/N;
149 end
150
151 % getting the deviation of each agent from the average states
152 for j = 1:N
153     x_err(:, :, j) = x(:, :, j) - x_avg;
154 end
155
156 % setting up a vector of colors, to have different colors for
       each agent,
157 % but the same color for all states of one agent
158 % colors are kept per line
159 color_vector = [1, 0, 0; % red
160                 0, 0, 0.5; % navy
161                 0, 1, 0; % green
162                 0, 1, 1; %another shade of blue -> aqua
163                 0.5, 0, 0]; % maroon
164
165 fig_num = 1;
166 f2 = figure(fig_num);
167 fig_num = fig_num + 1;
168 set(f2, 'Position', [screenSize(1)+20, screenSize(2)+60,...
169     3.75*screenSize(3)/4, 3*screenSize(4)/4],...
```

```
    'PaperOrientation', 'landscape', 'PaperPositionMode', '
      auto',...
    'PaperUnits', 'normalized', 'PaperType', 'usletter',...
    'NumberTitle', 'off',...
    'Name', 'Graph',...
    'MenuBar', 'figure',...
    'ToolBar', 'auto')
% this figure is showing the states values, but they will go
   to infinity
% since the system is unstable
for i = 1:size(A, 1)
    legend_text = [];
    figure(f2);
    subplot(ceil(size(A, 1)/ceil(sqrt(size(A, 1)))), ceil(sqrt
      (size(A, 1))), i)
    for ag_ind = 1:N
        stairs(t, x(i, :, ag_ind), 'Color', color_vector(
           ag_ind, :),...
            'LineWidth', 1);
        hold on
        legend_text = [legend_text; ['agent ', num2str(ag_ind,
            '%d')]];
    end
    % legend(legend_text, 4)
    hold off
end
% print -dpdf -r150 'Chap4_Ex2_MAS.pdf';

f3 = figure(fig_num);
fig_num = fig_num + 1;
set(f3, 'Position', [screenSize(1)+20, screenSize(2)+60,...
    3.75*screenSize(3)/4, 3*screenSize(4)/4],...
    'PaperOrientation', 'landscape', 'PaperPositionMode', '
      auto',...
    'PaperUnits', 'normalized', 'PaperType', 'uslegal',...
    'NumberTitle', 'off',...
    'Name', 'Graph',...
    'MenuBar', 'figure',...
    'ToolBar', 'auto')
% this figure shows the fan region with the Laplacian
   eigenvalues placed in
% space
g1 = gca;
set(g1, 'Units', 'normalized')
```

```
207 set(g1, 'Position', [0.05, 0.05, 0.9, 0.9]) % (0.1, 0.1) lower
      left corner, then the length on x-axis and the length on y
    -axis
208
209 ang = 0:pi/100:2*pi;
210 x_mu_A = (1/mu_A).*cos(ang) + 1;
211 y_mu_A = (1/mu_A).*sin(ang);
212 x_delta = delta.*cos(ang) + 1;
213 y_delta = delta.*sin(ang);
214 line3x = [0:0.001:1.5];
215 line3y = line3x*tan(thetar);
216
217 plot(x_mu_A, y_mu_A, 'LineWidth', 2, 'LineStyle', '-.', 'Color
    ', 'red',...
218     'LineWidth', 2)
219 hold on
220 plot(x_delta, y_delta, 'LineWidth', 2, 'LineStyle', '-', '
    Color', 'blue')
221 plot(line3x, line3y, 'LineWidth', 2, 'LineStyle', '-', 'Color'
    , 'black')
222 plot(line3x, -line3y, 'LineWidth', 2, 'LineStyle', '-', 'Color
    ', 'black')
223
224 % arrow for the 1/muA radius circle
225 annotation('arrow', [0.05+3*0.9/4, 0.05+(2+(1/mu_A)*cosd(100)
    +1)*0.9/4],...
226     [0.05+1*0.9/2, 0.05+(1+(1/mu_A)*sind(100)+0)*0.9/2], '
        LineWidth', 1.5)
227
228 % lines from center r0 = 1 to the extreme eigenvalues lambda_2
     and lambda_N
229 line([1, real(lambda_2)], [0, imag(lambda_2)], 'Color', 'black
    ', 'LineWidth', 1);
230 line([1, real(lambda_4)], [0, imag(lambda_4)], 'Color', 'black
    ', 'LineWidth', 1);
231
232 % small arc to show the angle
233 arc3x = 0.4*cos([0:pi/100:thetar]);
234 arc3y = 0.4*sin([0:pi/100:thetar]);
235 plot(arc3x, arc3y, '-', 'LineWidth', 1, 'Color', 'black');
236
237 % text annotations
238 text(0.42, 0.05, ['$\theta = ', num2str(theta, '%d'), '^{\circ
    }$'],...
239     'FontSize', 12, 'FontWeight', 'bold')
```

```
240 text(0.95, 0.6, ['$\frac{1}{\mu(A)} = ', num2str(1/mu_A, '%.2f
    '), '$'],...
241     'FontSize', 12, 'FontWeight', 'bold')
242 text(1.05, 0.1, ['$\delta = ', num2str(delta, '%.2f'), '$'
    ],...
243     'FontSize', 12, 'FontWeight', 'bold')
244 text(1, -.05, '$1$', 'FontSize', 12, 'FontWeight', 'bold')
245 text(real(lambda_2)-0.03, imag(lambda_2)+0.06, '$\lambda_{2}$'
    ,...
246     'FontSize', 12, 'FontWeight', 'bold')
247 text(real(lambda_4)-0.05, imag(lambda_4)+0.06, '$\lambda_{4}$'
    ,...
248     'FontSize', 12, 'FontWeight', 'bold')
249
250 % these are the axis
251 annotation('arrow', [0.05+1.8*0.9/4, 0.95], [0.5, 0.5], '
    LineWidth', 2)
252 annotation('arrow', [0.05+2*0.9/4, 0.05+2*0.9/4], [0.05,
    0.95], 'LineWidth', 2)
253
254 plot(lambdas, 'o', 'MarkerSize', 8, 'MarkerEdgeColor', 'black'
    ,...
255     'MarkerFaceColor', 'green', 'LineWidth', 2)
256 %grid
257 axis([-2, 2, -1, 1])
258 hold off
259 axis off
260 ftime = clock;
261 filename1 = char(['./Figs/Chap4_SI_MAS_3_states_mySol_EV_',
    num2str(ftime(1)),...
262     '_', num2str(ftime(2)), '_', num2str(ftime(3)), '_',
        num2str(ftime(4)),...
263     'h_', num2str(ftime(5)), 'min']);
264 print('-dpdf', '-r150', filename1);
265
266 % next figures show the deviation of each agent from the
    average state
267 for i = 1:size(A, 1)
268     legend_text = [];
269     figure(fig_num);
270     fig_num = fig_num + 1;
271     set(gcf, 'Position', [screenSize(1)+20, screenSize(2)
        +60,...
272         3.75*screenSize(3)/4, 3*screenSize(4)/4],...
```

```
273         'PaperOrientation', 'landscape', 'PaperPositionMode',
              'auto',...
274         'PaperUnits', 'normalized', 'PaperType', 'usletter'
              ,...
275         'NumberTitle', 'off',...
276         'Name', 'Graph',...
277         'MenuBar', 'figure',...
278         'ToolBar', 'auto')
279     for ag_ind = 1:N
280         figure(gcf)
281         stairs(t, x_err(i, :, ag_ind), 'Color', color_vector(
              ag_ind, :),...
282           'LineWidth', 2);
283         hold on
284         legend_text = [legend_text; ['agent ', num2str(ag_ind,
              '%d')]];
285     end
286     legend(legend_text, 4)
287     xlabel('Time step', 'FontWeight', 'bold', 'FontSize', 12)
288     ylabel(['$ x_{', num2str(i,'%d'), '} - \bar{x}_{', num2str
          (i,'%d'),...
289       '}$'], 'FontWeight', 'bold', 'FontSize', 16)
290     hold off
291     filename2 = char(['./Figs/
          Chap4_SI_MAS_3_states_mySol_state_', num2str(i, '%d')
          ,...
292       '_', num2str(ftime(1)),...
293       '_', num2str(ftime(2)), '_', num2str(ftime(3)), '_',
            num2str(ftime(4)),...
294       'h_', num2str(ftime(5)), 'min']);
295     print('-dpdf', '-r150', filename2);
296 end
```

C.4 MATLAB Main File for Example 4.3.2

```
1  clc
2  clear all
3  close all
4  format short g
5  format compact
6
7  screenSize = get(0, 'screensize');
8  set(0,'DefaultTextInterpreter','Latex')
9
10 f1 = figure(1);
```

```
set(f1, 'Position', [screenSize(1)+20, screenSize(2)+60,...
    3.75*screenSize(3)/4, 3*screenSize(4)/4],...
    'PaperOrientation', 'landscape', 'PaperPositionMode', '
       auto',...
    'PaperUnits', 'normalized', 'PaperType', 'usletter',...
    'NumberTitle', 'off',...
    'Name', 'Graph',...
    'MenuBar', 'figure',...
    'ToolBar', 'auto')
% 'PaperPosition', [0.05, 0.05, 0.9, 0.9],...
%g1 = gca;
%set(g1, 'Units', 'normalized')
%set(g1, 'Position', [0.05, 0.05, 0.9, 0.9]) % (0.1, 0.1)
   lower left corner, then the length on x-axis and the length
    on y-axis

A = [-0.4326    1.1909   -0.1867    0.1139    0.2944
   -1.6656    1.1892    0.7258    1.0668   -1.3362
    0.1253   -0.0376   -0.5883    0.0593    0.7143
    0.2877    0.3273    2.1832   -0.0956    1.6236
   -1.1465    0.1746   -0.1364   -0.8323   -0.6918];
B = [1.7160   -0.7998    1.3372
    2.5080    1.3800    2.3817
   -3.1875    1.6312   -2.4049
   -2.8819    1.4238   -0.0396
    1.1423    2.5805   -0.3134];

eig_A = eig(A); % eigenvalues of A
% calculating the mahler measure according to definition
mu_A = 1; % initialize Mahler measure
for i = 1:length(eig_A)
    mu_A = mu_A * max(abs(eig_A(i)), 1);
end
upper_bound = mu_A;
lower_bound = nthroot(mu_A, rank(B));
% Since I know that this example has all eigenvalues of A
   outside the unit
% circle, I know that Au = A, Bu = B.
% The Lyapunov equation in the lemma is solved considering its
    solution
% being given by Zu = ((1 - gamma^(-2)) Xu)^(-1).
gamma_opt = []; % initialize the vector that would store the
   optimal values for gamma
Zu = dlyap(inv(A), inv(A)*B*B'*inv(A)');
% initial value for the optimal gamma
```

```
gamma_opt(1) = sqrt(max(eig(eye(rank(B)) + B'*inv(Zu)*B)));
gamma_opt

% Starting to introduce the D matrix
D = eye(size(B, 2)); % size(B, 2) gives the number of columns
   of B
B_temp = B;
Zu_temp = Zu;
tol = 0.095;
i = 1;

while floor(abs(lower_bound - gamma_opt(end))*1000)/1000 > tol
    Zu_temp = dlyap(inv(A), inv(A)*B_temp*B_temp'*inv(A)');
    gamma_opt(i) = sqrt(max(eig(eye(rank(B_temp)) + B_temp'*
       inv(Zu_temp)*B_temp)));
    D_temp = chol(B_temp'*inv(Zu_temp)*B_temp);
    D = D_temp*D;
    B_temp = B*inv(D);
    rank(B_temp);
    i = i + 1;
end

gamma_opt
figure(f1)
plot(1:1:length(gamma_opt), gamma_opt, 'ob', 'LineWidth',
   1.5,...
    'MarkerSize', 8, 'MarkerEdgeColor', 'black',...
    'MarkerFaceColor', 'red')
hold on
plot(1:1:length(gamma_opt), lower_bound, 'vr', 'MarkerSize',
   8,...
    'MarkerEdgeColor', 'black', 'MarkerFaceColor', [0.2, 0.4,
       1],...
    'LineWidth', 1.5)
plot(1:1:length(gamma_opt), upper_bound, '^r', 'LineWidth',
   1.5,...
    'MarkerSize', 8, 'MarkerEdgeColor', 'black',...
    'MarkerFaceColor', [0.2, 0.4, 1])
grid
hold off
ca = gca;
caXTick = get(ca, 'XTick');
set(ca, 'XTick', [caXTick(1):1:caXTick(end)]);
ylabel('optimal $\gamma$ ($\gamma_{\mathrm{opt}}$)', '
   FontWeight', 'bold', 'FontSize', 14)
```

```
88 xlabel('$\mathcal{K}$', 'FontWeight', 'bold', 'FontSize', 14)
89 legend('$\gamma_{\mathrm{opt}}$', 'lower bound', 'upper bound'
     )
90 print -dpdf -r150 'Chap4_Ex1.pdf';
```

Appendix D

Extra Equations

Useful Equations for Proof of Lemma 4.3.1

$$\begin{aligned}\left((\mathbf{1}r^{\mathrm{T}}) \otimes I_n\right)(I_N \otimes \mathcal{A}) &= \left((\mathbf{1}r^{\mathrm{T}})I_N\right) \otimes (I_n\mathcal{A}) = \left(I_N(\mathbf{1}r^{\mathrm{T}})\right) \otimes (\mathcal{A}I_n) \\ &= (I_N \otimes \mathcal{A})\left((\mathbf{1}r^{\mathrm{T}}) \otimes I_n\right) \end{aligned} \tag{D.1}$$

$$\left((\mathbf{1}r^{\mathrm{T}}) \otimes I_n\right)(\mathcal{L} \otimes BK) = \left[\left(\mathbf{1}r^{\mathrm{T}}\right)\mathcal{L}\right] \otimes (I_nBK) = \left[\left(\mathbf{1}r^{\mathrm{T}}\right)\mathcal{L}\right] \otimes (BKI_n)$$

Since r is a left eigenvector for the Laplacian matrix $\mathcal{L}$ corresponding to the 0 eigenvalue, i.e. $r^{\mathrm{T}}\mathcal{L} = 0 \cdot r^{\mathrm{T}}$, and

$$\begin{aligned}\mathcal{L} \cdot \mathbf{1} &= \begin{bmatrix} \sum_{j=1}^{N} a_{1j} & -a_{12} & -a_{13} & \dots & -a_{1N} \\ -a_{21} & \sum_{j=1}^{N} a_{2j} & -a_{23} & \dots & -a_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{N1} & -a_{N2} & -a_{N3} & \dots & \sum_{j=1}^{N} a_{Nj} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} a_{11} \\ a_{22} \\ \vdots \\ a_{NN} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \cdot 0 = \mathbf{1} \cdot 0 \end{aligned}$$

it follows that $\left(\mathbf{1}r^{\mathrm{T}}\right)\mathcal{L} = \mathbf{1}\cdot 0\cdot r^{\mathrm{T}} = \mathcal{L}\cdot\mathbf{1}\cdot r^{\mathrm{T}}$. Therefore,

$$\begin{aligned}\left((\mathbf{1}r^{\mathrm{T}})\otimes I_n\right)(\mathcal{L}\otimes BK) &= \left[\mathcal{L}\left(\mathbf{1}r^{\mathrm{T}}\right)\right]\otimes(BKI_n)\\ &= \left[\mathcal{L}\otimes BK\right]\left[\left(\mathbf{1}r^{\mathrm{T}}\right)\otimes I_n\right]\end{aligned} \tag{D.2}$$

The following equations are used to detail the process of getting the result in equation (4.18).

$$\begin{aligned}(\mathcal{T}^{-1}\otimes I_n)(I_N\otimes A) &= (\mathcal{T}^{-1}I_N)\times(I_nA) = (I_N\mathcal{T}^{-1})\times(AI_n)\\ &= (I_N\otimes A)(\mathcal{T}^{-1}\otimes I_n)\end{aligned} \tag{D.3}$$

$$\begin{aligned}(\mathcal{T}^{-1}\otimes I_n)(\mathcal{L}\otimes BK) &= (\mathcal{T}^{-1}\mathcal{L})\otimes(I_nBK) = (\mathcal{J}\mathcal{T}^{-1})\otimes(BKI_n)\\ &= (\mathcal{J}\otimes BK)(\mathcal{T}^{-1}\otimes I_n)\end{aligned} \tag{D.4}$$

$$\begin{aligned}\varepsilon(k+1) &= \left[(I_N\otimes A)(\mathcal{T}^{-1}\otimes I_n) - (\mathcal{J}\otimes BK)(\mathcal{T}^{-1}\otimes I_n)\right]\\ &= \left[I_N\otimes A - \mathcal{J}\otimes BK\right](\mathcal{T}^{-1}\otimes I_n)\varpi k\\ &= \left[I_n\otimes A - \mathcal{J}\otimes BK\right]\varepsilon(k)\end{aligned} \tag{D.5}$$

To obtain equation (4.19), the following set of equations are useful.

$$(r^{\mathrm{T}}\otimes I_n)(\widehat{M}\otimes I_n) = (r^{\mathrm{T}}\widehat{M})\otimes(I_nI_n) = \mathbf{0}_{1\times N}\otimes I_n = \mathbf{0}\in\mathbb{R}^{n\times nN}. \tag{D.6}$$

$$
\begin{aligned}
r^{\mathrm{T}}\widehat{M} &= \begin{bmatrix} r_1 & r_2 & \dots & r_N \end{bmatrix} \begin{bmatrix} 1-r_1 & -r_2 & \dots & -r_N \\ -r_1 & 1-r_2 & \dots & -r_N \\ -r_1 & -r_2 & \dots & 1r_N \end{bmatrix} \\
&= \begin{bmatrix} r_1(1-\sum_{j=1}^{N} r_j) & r_2(1-\sum_{j=1}^{N} r_j) & \dots & r_N(1-\sum_{j=1}^{N} r_j) \end{bmatrix} = \mathbf{0} \in \mathbb{R}^{1\times N}. \quad \text{(D.7)}
\end{aligned}
$$

Vita

Laurenţiu Dan Marinovici was born in October, 1976, in Bicaz, Romania. He completed his undergraduate studies in computer engineering at "Gh. Asachi" Technical University of Iaşi, Romania, in August 2000, after spending a summer semester as an exchange student at The University of Sheffield, England, writing his thesis. He earned a Master of Science degree in June 2001 in automatic control at "Gh. Asachi" Technical University of Iaşi, Romania. From September 2000 until August 2004, he worked as an instructor for the department of Automatic Control and Applied Informatics, within the Faculty of Automatic Control and Computer Engineering at the above mentioned university. In January 2005 he came to Louisiana State University to pursue graduate studies in the Electrical and Computer Engineering Department, focusing on systems control. He was awarded the Master of Science in Electrical Engineering degree by Louisiana State University in December 2007. He is currently a candidate for the Doctor of Philosophy degree in electrical engineering to be awarded in August 2011.