2007

# Distributed controller synthesis and decision making

Vikas Yadav
*Iowa State University*

# Distributed controller synthesis and decision making

by

Vikas Yadav

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering

Program of Study Committee:
Murti V. Salapaka, Major Professor
Nicola Elia
Wolfgang Kliemann
Degang J. Chen
Zhengdao Wang
Atul Kelkar

Iowa State University

Ames, Iowa

2007

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

## DEDICATION

*To my mommy and papa, Raj Kumari and Ram Asrey Yadav, loving sister Shraddha; and younger brothers Puneet and Praveen, for their thoughts, love, hardship, care and support.*

# TABLE OF CONTENTS

## LIST OF FIGURES

# ACKNOWLEDGEMENTS

I have come a long way from graduating from high school to this day. Regardless of the initial condition, the present is as good as it gets. This would not have been possible without timely support and encouragement from a long list of people. This page is to talk about all of them, without whom this work would not have materialized in its final form. I am deeply grateful to all of them for their association in my life.

First and foremost I would like to thank my advisor Dr. Murti V. Salapaka. I could not imagine a better advisor for a research student. He groomed me into a researcher who can think independently about problems and find solutions. It was his faith in me that took me to the culmination of my research work. I would cherish my years spent with him for the rest of my life. I would like to thank Dr. Nicola Elia who had been extremely accessible for various valuable discussions related to research which had helped me in developing a critical perspective towards varying problems and their solutions. I would also like to thank Dr. Wolfgang Klienmann who always took time out of his busy schedule to discuss with me about some possibility of solving quadratic equations of rational stable transfer functions. I would further like to thank Dr. Zhangdao Wang, Dr. Degang Chen and Dr. Atul Kelkar for being on my thesis committee; they had not only been critical of my work but also very helpful with their suggestions. I would also like to acknowledge Dr. Umesh Vaidya, Dr. Govindrasu Manimaram and Dr. Alexandra Dogandzic for being very supportive and encouraging teachers. Special thanks are due to our department graduate office secretary Pam Myers for being magical in managing seemingly impossible situations.

I would like to express my sincere thanks to all of my friends in different parts of the world, busy with their own work, who had been with me when I needed them, and also when I did not need them. Specially the residents of the apartment number eighteen on one-two-three Sheldon Ave viz. Vipul Katyal, Nikhil Ranade, Shourya Otta, Rahul Biyani, Hullas Sehgal and Pranav Agarwal for all those fun years together and the great food that we cooked for each other. Still the best food that I had during my graduate school days at Iowa State was always made by Durga (Dr. Murti's wife). She always made sure that our group met at her house for dinner each semester (followed by those dumb-charades and discussions). She had been superb. Throughout my research stint at Iowa State I had been fortunate to meet guys like Abu Sebastian, Anil Gannepalli, Deepak Sahoo, Tanuj Aggarwal and Tathagate De, the part of AFM team of Nanodyanmics Lab. Tatha came to Iowa State at the same time as me, and since then we had been with each other during good, bad and ugly times. It was a beginning of a wonderful friendship.

I would also like to attempt to list all possible names of my friends that I could recollect at this moment of time. My big thanks to:

Friends from school days at Kanpur: *Avinash Sharma, Jeetendra Pal, Kapil Goel, Arshad Khan, Shalini Bajpai, Ritesh Tewari, Manish Pant, Mayank Goel, ...*

Friends from days at Indian Institute of Technology, Kanpur: *Anwitaman Datta, Tamal Das, Manpreet Singh, Kousik Das, Shishir Awasthi, Sunil Kumar, Anilabh Pandey, N. Sudarshan, Shourya P Otta, Anup Malhotra, Vishal Dhall, Vishal Gupta, Amit Chhetri, Damanpal S Dogra, Ravi Ranjan, Amit Gupta, Nishant Agarwal, Manish Shrotriya, Abhishek Singh, Vipul Katyal, Naveen Tewari, Srijoy Das, Nishit Verma, Chandra S Nath, Anupam Verma, ...*

Friends from days at work in Chennai: *Bharanidharan, Anil Chaitanya, Roma Shukla, Zahara Aliya, Sangeetha K, Jayabharti, Nitin Gupta, Amresh, Akash, Girish, Ravi, Shailaja, Nithya, Divya, Viju, Dhanya, Manish, Chandan, Deepak, Andy, Roshit, Janani, Amit Banwari, Ojha, Chitralekha, Anumodh, Aniruddha, Amit Sekhar, Tarun, Kannan, Kanagvel, Srikanth N, Kesav, Mahalakshmi, Ramanna, Senthil, Saumzy, Swati, Prachi, Hari, Vivek, Subrat Mohanty, Rajesh MV, Sindu Chawla, Girija R, Amudha, PF, ...*

From Graduate school days at Iowa State: *Rajesh Ramanujam, MuraliRavirala, Nikhil Ranade, Pritesh, Jacob Purakal, Sridhar Sutty, Rakesh Raghvan, Pallab Datta, Deepak Sa-*

*hoo, Tathagata, Ravi Cherukuri, Srikanth Koma, Deepak Makarala, Pavan Vennupusa, Abu Sebastian, Anil Gannepalli, Amruta Karindkar, Janhavi Agashe, Anupreet Kaur, Preeti Kohli, Vishwas Jaju, Shailesh Kumar, Abhinav Anand, Durga Panda, Durga K, Donatello Materassi, Ashutosh Tewari, Debju Gosh, Bipin Singh, Tanuj Aggarwal, Meghna Kapur, Supriyo Das, Kamal K, Rahul Marathe, Murali Subramanium, Rashmi Bahuguna, Vinay Deodeshmukh, Satyadev, Ganesh TS, Souvik, Hemant Bhanot, Avanika Mahajan, Aakriti Chaudhari, Shamik, Sayaan, Naveen, Venkat Krish, Sunil Singh, Manoj, Ruchita Fuloria, Aarohi Zokarkar, Kavitha B, Dinesh Cheryagi, Vidya Iyer, Shantha Daniel, Rajeshwari Mani, Muthu KS, KS Karup, Muthu, Zara, Saikat, Anup Gokran, Maduresh, Sourabh Dawar, Pavan Karra, Hemant Porumamilla, Virendra Allada, Soma Dhavale, Bhooshan Karnik, Aparna Karnik, Amit Sachan, Amita Sachan, Arpita Ray, Liu Jialing, Jinhua Jin, Xin Qi, Jeff Eisenbeis, Xiao Fan, ...*

I must had missed several names in the above lists, I would also like to thank them for being nice, encouraging and helpful friends (and my apologies for being forgetful enough to miss their names). In general, I would like to thank people of India for it was their policies and monies which provided me an opportunity to get one of the finest educations in the world, the stepping stone to this thesis. Oh yeah, how can I forget to say thanks to the wonderful city of Ames and its great people.

Finally, I would like to thank my parents Raj Kumari Yadav and Ram Asrey Yadav for their dedication and hardship; and to my sister Shraddha who had been my guide, my contact person to the outside world for whole of my childhood; and to my younger brothers Puneet and Praveen who had been very caring and supportive to me in my endeavor to achieve my goals.

Thanks to all of you, I am a better person now.

# ABSTRACT

Distributed controller design and distributed decision making have been hot topics of investigation in the last few years. New technologies have led to systems where it is critical to identify architectures that distribute the controller effort over sub-controllers to respect the information flow and/or resource constraints. The communication uncertainty between sub-controllers partly governs the optimality of the architecture of the controller. The related synthesis methodology for optimal distributed controller has to address internal stability concerns and has to incorporate the effect of communication uncertainty into the performance metric. In the first part of this thesis, a methodology is developed to address the concerns of sub-controller communication uncertainty. It is demonstrated that different canonical architectures of a centralized design result in appreciably different performance. Methods to identify architectures of information flow where the optimal performance problem is convex are developed. In addition, synthesis methods to incorporate robustness measures with respect to model uncertainty of the communication channel are obtained for the associated distributed architectures. These methods are further refined for specific structures of information flow in the system. In the second part of this thesis, issues in distributed decision making in a large network of nodes are discussed, in particular a distributed averaging consensus protocol is considered which converges asymptotically. However, each node individually never comes to know of the occurrence of convergence, and thus it keeps running required computation and communication throughout its life. This is not desired, as in most of the networks the power of each node is a very limited resource. This thesis provides a distributed algorithm through which each node can distributively detect when the convergence has occurred within a given error margin. This distributed detection takes finite time and happens simultaneously.

# CHAPTER 1. INTRODUCTION

Recent technological demands of high performance in the presence of information flow constraints and large computational loads have posed new challenges for controller synthesis. In many of these scenarios, the controllers have to be synthesized in a manner that is amenable to distributive implementation.

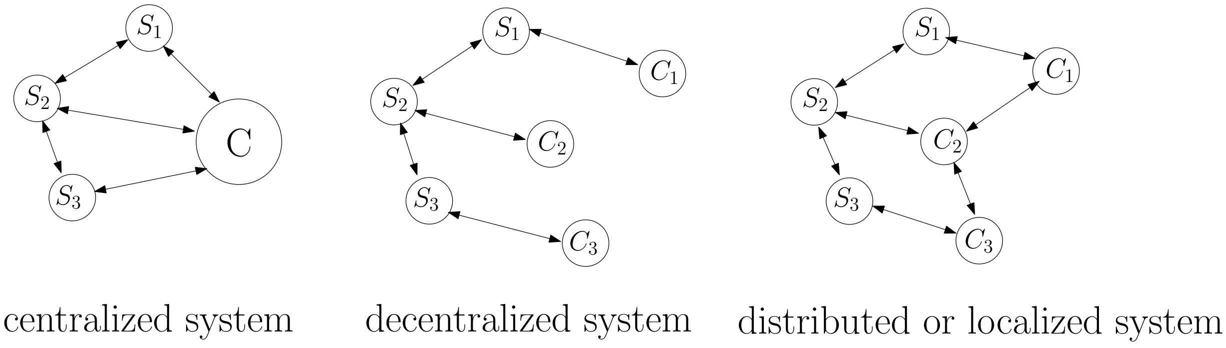
Large computational load might arise in systems consisting of several subsystems connected with each other via large numbers of sensors and actuators. An example where such a large load is apparent is the recently proposed, massively parallel cantilever based data storage device, where thousands of cantilevers operating in parallel have to be controlled (1). Large computational load may also be imposed by high performance specifications on a small system where the resulting controller is sufficiently complex. In these situations, the controller cannot be realized at one location and the large load has to be shared by multiple stations.

Distributed controller design is also motivated by new hardware that is increasingly employing various components that meet different specifications. For example, recent hardware for real-time applications is being realized with a mix and match of various computational components like FPGA's and DSP's. In such cases it is often possible to abstract the hardware into distinct regions with communication between the regions that is possibly uncertain.

The need to address sub-controller communication uncertainty arises naturally in distributed systems where a sub-controller associated with a sub-system is to be locally realized at a station with structured information exchange with other substations. Typically this communication between stations is corrupted or uncertain. In applications like sensor networks (2) and power distribution, a need for collocating a sub-controller with a corresponding subsystem imposes a need for distributing the overall controller effort with a pre-specified information flow architecture.

In all examples outlined above, the task of determining how the controller task can be divided into various sub-controllers such that plant-controller interconnection is stable and the effect of sub-controller communication uncertainty on the performance is minimized is to be determined.

In *centralized controller* framework, a controller is realized at a single station and there is no need to consider effects of uncertainty affecting various components of the controller. Such a centralized framework is not possible due to reasons provided previously. In *decentralized controller* design method, a sub-controller interacts only with its subsystem and is independent from other subsystems. While, in framework of distributed controller design, local sub-controllers are collocated with their subsystems. However in this case, communication between various sub-stations might be possible.



centralized system decentralized system distributed or localized system

Figure 1.1 Centralized, decentralized and distributed frameworks

The task of designing distributed controllers lead to structural constraints due to the absence of specific communication links between various sub-controllers. The design of controllers that achieve optimal performance when structural constraints are present, even without considering uncertain communication, is a difficult problem (see for example (3; 4; 5) and references therein). Recently, identification of specific classes of problems where the structural constraints can be addressed via a convex optimization methods is reported in (6; 7; 8). In (7) a necessary and sufficient condition on the controller structure was derived that ensures that the optimization problem remains convex in the Youla parameter $Q$ (9). In these works even though the optimal controller transfer matrices satisfying structural constraint are obtained, the realization that implements the transfer matrix into various sub-controllers remains unaddressed. As will be shown in this thesis, this aspect is not obvious and seemingly natural ways of distributed realization may even destabilize the interconnection. The internal stability of distributed realization is discussed in Chapter 1.

In other related work, (10; 11) distributed spatially invariant systems are studied using a state space approach and a convex method based on solving constraints in the form of LMIs is presented to obtain structured controllers. In (12), a heuristic method for structured $\mathcal{H}_2$ controllers is presented based on low dimensional LMIs. These efforts do not consider the effect of uncertainty affecting the communication between different sub-controllers.

Motivated by the concerns outlined above, in Chapter 3 a framework for designing architectures for distributed implementation of controllers, that incorporates sub-controller uncertainty is obtained. Unlike the case when there is no sub-controller noise, a central issue in this thesis is the identification of the signals that need to be transmitted between sub-controllers. In the presence of sub-controller noise it is also important to consider a bound on the strength of sub-controller communication signal. Such a concern adds another variable to the performance metric. In related prior work, in (13), distributive implementation of a state-feedback control law is considered and an iterative algorithm is presented to minimize the effect of sub-controller noise on performance. Preliminary work by authors related to issues raised are reported in (14; 15; 16). Two special structures on controllers due to information constraints viz. nested and banded structures are considered in Chapter 4, and architectures for distributed implementation of controllers is identified by using results from the previous chapter. In Chapter 5 analysis and synthesis of distributed robust controllers in presence of uncertainty in the channel model is presented. The significance of these architectures is illustrated with help of two examples in Chapter 6.

Distributed decision making problems like consensus and self-organization have been a very well studied topic among scientists from various backgrounds like computer science, control engineering, physics and biology (for example (2; 37)). Consensus or agreement in a large network of agents refers to the event in which each agent has same information. It is assumed that each node is sharing information with its neighbors. Averaging consensus is a special case where the each node starts with some initial node-value and as a result of agreement it obtains a value which is an average of initial node-values of all the nodes in the network. Averaging consensus protocol refers to the action to be performed on the received information. In this thesis, the focus is on the linear averaging protocol presented in (33) where each node takes an average of the information received from neighboring nodes. (33) provides a necessary and sufficient condition that underlying network is strongly connected and balanced which leads to an asymptotic convergence in absence of any malicious user. It has been shown in (38) that the requirement for balanced graph can be dropped by using weighted integrators in the protocol. A faster linear averaging protocol similar to (33) is proposed in (39). A condition on functions that can be computed distributively is provided in (23) is to be time invariant. A good survey of consensus problems is provided in (34; 32). In (30) authors have provided convergence analysis for the angular interaction among agents using a switched linear model. This model also assumes that over every finite period of time the particles are jointly connected for the length of the entire interval. See also (35; 36) for extension of (30). Similar agrement problem over random graphs is addressed in (29) for graphs having binomial distribution. Most of these works assume some kind of connectivity in the network. In (24) work has been done towards maintaining the connectivity of network by controlling the algebraic connectivity (also known as the second smallest eigenvalue of Laplacian of graph) of the network. In Chapter 7, the distributed averaging protocol along with maximum and minimum consensus protocol are reviewed. In (28) authors have addressed agreement problem over geometric random graphs with noisy communication. They showed the convergence in presence of a modified update rule where the nearest neighbor value is scaled by a special time varying step size. A malicious or faulty node is one which is not following the consensus protocol. In presence of such nodes, the averaging protocol becomes unstable, i.e. it fails to converge. In (22) authors have provided results on stabilizing consensus protocol in presence of faults by assuming that node-values are strictly non-negative. Their method is restrictive and suffers from extensive use of message passing and buffering. An algorithm based on observing spatial correlation of parameters of each node is proposed in (25), where a node is declared faulty with a high accuracy if its behavior is not correlated with its neighbors.

In large sensor networks, each node is having limited power for its computational and communication need. In all consensus protocols, convergence takes place in asymptotic sense, and there is no distributed way for each individual node to know if the convergence has reached within desired error margin. Because, if each node can detect the consensus occurrence, then they can stop doing computation and communication required by the consensus protocol, and thus saving on the limited power supply. In Chapter 8, a distributed algorithm is presented which facilitates each node to detect the occurrence of consensus within desired bounds in finite time. This algorithm requires implementation of maximum and minimum consensus protocols, which have finite convergence time bounded by the diameter of the network. Some simulations results are presented in Chapter 9 to demonstrate working of the algorithm.

# PART I

# DISTRIBUTED CONTROLLER SYNTHESIS

# CHAPTER 2. INTERNAL STABILITY OF DISTRIBUTED ARCHITECTURES

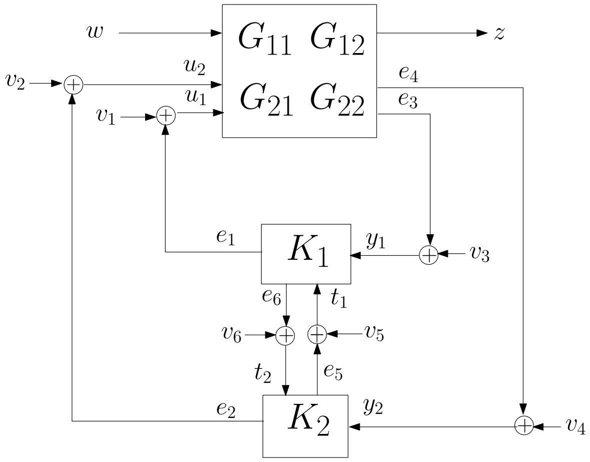


Figure 2.1 The $G - K_1 - K_2$ interconnection

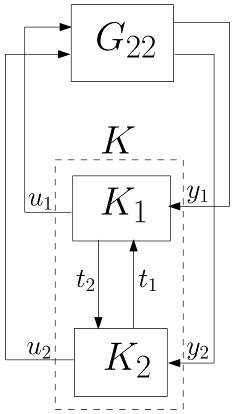


Figure 2.2 $G_{22} - K_1 - K_2$ interconnection

The general framework for a distributed interconnection, considered in this thesis, is illustrated in Figure 2.1 where $G$ represents the generalized plant, sub-controllers $K_1$ and $K_2$ represent distributed implementation of the stabilizing controller $K$, with $w$, $z$, $u$ and $y$ representing, exogenous input, regulated output, control effort and the measured output respectively. $G$, $K_1$ and $K_2$ are assumed to be discrete time, linear and time invariant systems. The plant and controller interconnection with distributed implementation of controller $K$ using sub-controllers $K_1$ and $K_2$ is shown in Figure 2.2 where $G_{22}$ is part of $G$ that maps $u$ to $y$.

The standard interconnection of the generalized plant $G$ with centralized stabilizing controller $K$ is shown in Figure 2.3. The interconnection of the plant $G_{22}$ and $K$ is shown in Figure 2.4.

The main issue addressed in this thesis is the effect of communication uncertainty between sub-controllers $K_1$ and $K_2$ and means of incorporating these effects into controller synthesis

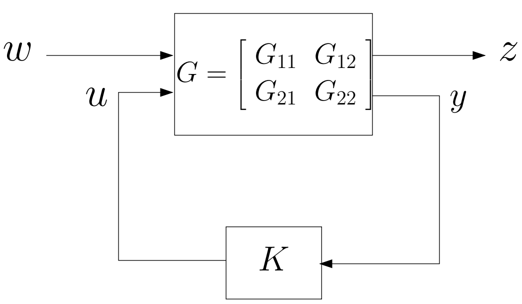

w
u
$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$
z
y
K

Figure 2.3 The $G - K$ interconnection

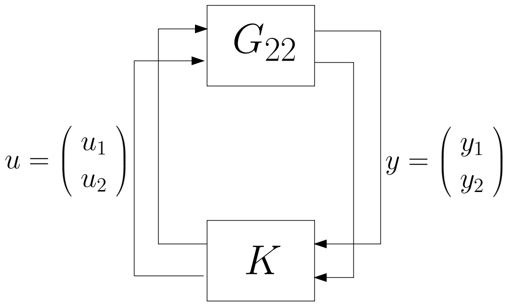

$G_{22}$
$u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$
$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$
K

Figure 2.4 The $G_{22} - K$ interconnection

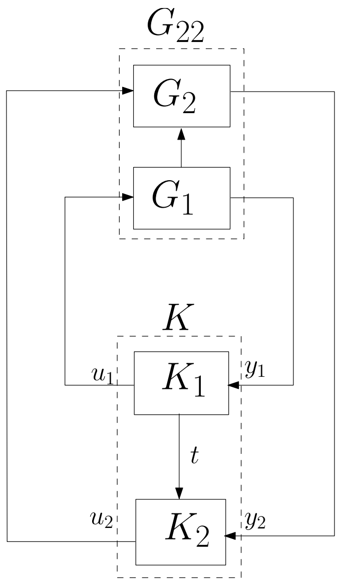


Figure 2.5 2-nest system with signal $t$ transmitted from inner nest sub–controller $K_1$ to outer nest sub-controller $K_2$

methods.

The controller $K$ might also have to satisfy specific information flow constraints. Two specific structures considered in this thesis are the nested structure that is characterized by a block triangular structure of the transfer matrix $K$ and banded structure characterized by delays between sub-systems. Nested and banded structure appear in many applications (8). Figure 2.5 shows a nested structure controller $K$ where $t$ is a transmitted signal from $K_1$ to $K_2$. No signal is transmitted from $K_2$ to $K_1$. The controller $K$ has lower triangular structure

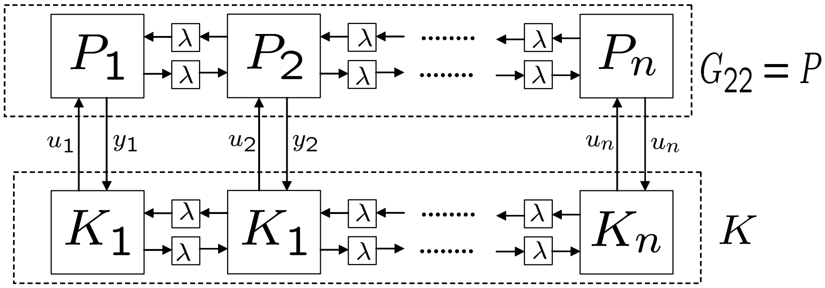


Figure 2.6 Banded structure system.

given by $K = \begin{bmatrix} K_{11} & 0 \\ K_{21} & K_{22} \end{bmatrix}$.

Another structure considered in this thesis is the banded structure which is characterized by one step delay in interactions between nearest subsystems as depicted in Figure 2.6. Controllers with this information constraint are called banded controllers.

The transfer matrix will have banded structure as shown below, where $\lambda$ denotes delay:

$$K(\lambda) = \begin{bmatrix} k_{11} & \lambda k_{12} & \cdots & \lambda^{n-1}k_{1n} \\ \lambda k_{21} & k_{22} & \cdots & \lambda^{n-2}k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda^{n-1}k_{n1} & \lambda^{n-2}k_{n2} & \cdots & k_{nn} \end{bmatrix} \tag{2.1}$$

## 2.1 Internal stability of distributed architectures

Note that in the standard setup the internal stability of Figure 2.3 is equivalent to the internal stability of Figure 2.4 provided the inherited realization of $G_{22}$ from a stabilizable and detectable realization of $G$ is itself stabilizable and detectable (17). The following result from robust control generalizes this result for the distributive setting of Figure 2.1 and Figure 2.2.

**Theorem 2.1.1** *Consider the $G - K_1 - K_2$ interconnection shown in Figure 2.1 where $G = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right]$ is the generalized plant. Let, $K_1$ and $K_2$ have stabilizable and detectable state space realizations such that the induced realization of the controller $K$ is stabilizable and detectable. If the interconnection with distributed implementation using $K_1$ and $K_2$ is well-posed* *and the inherited realization of $G_{22}$ from $G$ is stabilizable and detectable then the $G - K_1 - K_2$ interconnection shown in Figure 2.1 is internally stable if and only if $G_{22} - K_1 - K_2$ interconnection shown in Figure 2.2 is internally stable.*

**Proof** See the Appendix A.2 for proof. ∎

Motivated by the above result the following is assumed throughout the thesis

**Assumption 2.1.1** *All interconnections are well posed and the inherited realization of $G_{22}$ from a generalized plant $G$'s stabilizable and detectable realization is stabilizable and detectable.*

Using above result, by ensuring the internal stability of $G_{22} - K_1 - K_2$ interconnection, the internal stability of the overall distributed interconnection can be guaranteed. Thus, for rest of the discussion on internal stability the $G_{22} - K_1 - K_2$ interconnection is considered equivalent to the $G - K_1 - K_2$ interconnection.

When a controller is implemented distributively, the sub-controller communication architecture needs to be chosen such that the resulting interconnection is internally stable. Most input output approaches (see (8; 7)) obtain the optimal controller that meets the information flow structure but do not address the issue of how to distributively implement the controller. For example, consider the case shown in Figure 2.7 where a triangular structure is imposed on controller, such that location $K_2$ can obtain information from $K_1$ but there is no flow of

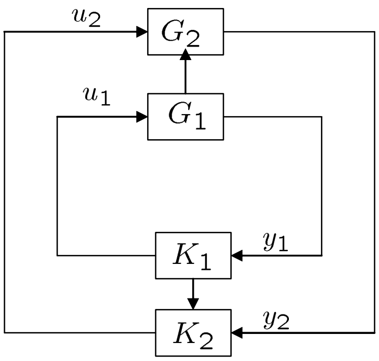


Figure 2.7 Nested structured system

information from $K_2$ to $K_1$. $K_1$ and $K_2$ can be obtained as parts of the overall controller $K$ which needs to be nested, i.e., of the form $K = \begin{bmatrix} K_{11} & 0 \\ K_{21} & K_{22} \end{bmatrix}$. Given a stabilizing $K$, there is a non-unique way of determining $K_1$ and $K_2$ as this depends on what signals are transmitted. For example, one choice is , $K_1 = \begin{bmatrix} K_{11} \\ K_{21} \end{bmatrix}$ and $K_2 = \begin{bmatrix} I & K_{22} \end{bmatrix}$ while a second choice is $K_1 = \begin{bmatrix} K_{11} \\ I \end{bmatrix}$ and $K_2 = \begin{bmatrix} K_{21} & K_{22} \end{bmatrix}$. Both realizations are identical in the absence of sub-controller to sub-controller communication noise.

Existing work in the input-output setting provide means of obtaining the optimal $K$ even when structural constraints are involved (see (7) and (8)). However, they do not address, for example, the task of determining whether to implement the first or the second choice of $K_1$ and $K_2$ in the case delineated above. Indeed, it can be shown that a $K$ that is stabilizing for the $G - K$ interconnection (see Figure 2.3) when implemented distributively in a $G - K_1 - K_2$ (see Figure 2.1) interconnection can be unstable with $K_1$ and $K_2$ chosen according to the second choice (see the Appendix A.1 for a complete description).

The new closed-loop maps in the distributed implementation that need to be considered for internal stability are the maps $\Phi_{zn}$ from the noise $n$ affecting the communication channel to the regulated variable $z$, the map $\Phi_{tw}$ from the exogenous input $w$ to the signal transmitted on the communication channel $t$, and the map $\Phi_{tn}$ from the noise affecting the communication channel to the signal transmitted on the channel. In addition to all the standard closed-loop maps that have to be stable for the $G - K$ interconnection in Figure 2.3 to be stable, these additional maps have to be stable to guarantee internal stability of the $G - K_1 - K_2$ interconnection of Figure 2.1.

The following theorem addresses the internal stability of a particular distributive implementation of a controller $K$ that stabilizes the interconnection in Figure 2.3.

**Theorem 2.1.2** *Consider the $G - K$ interconnection given in Figure 2.3 where $K$ is a centralized stabilizing transfer matrix. The distributive implementation of $K$ into $K_1$ and $K_2$ as shown in Figure 2.1 is internally stabilizing if any stabilizable and detectable realization of $K_1$*

*and $K_2$ is such that the induced realization of $K$ is stabilizable and detectable.*

**Proof** See the Appendix A.3 for proof. ■

The following corollary holds in the case of nested $G_{22}-K_1-K_2$ system shown in Figure 2.5.

**Corollary 2.1.1** *Consider a 2-nest $G_{22} - K$ interconnection where $K$ is a centralized stabilizing controller implemented in distributive manner using sub-controllers $K_1$ and $K_2$ as shown in Figure 2.5 with $t_2 = t$ and no transmission from $K_2$ to $K_1$. Let, $K_1$ and $K_2$ have state space realizations given by Let, $K_1$ and $K_2$ have state space realizations given by* $K_1 = \left[\begin{array}{c|c} A_{C1} & B_{C1} \\ \hline C_{C11} & \\ & 0 \\ C_{C12} & \end{array}\right]$ *and* $K_2 = \left[\begin{array}{c|cc} A_{C2} & B_{C21} & B_{C22} \\ \hline C_{C2} & \multicolumn{2}{c}{0} \end{array}\right]$ *such that $(A_{C1}, B_{C1}, C_{C11})$ and $(A_{C2}, B_{C21}, C_{C2})$ are stabilizable and detectable. Then, the induced realization of controller $K$ obtained from $K_1$ and $K_2$ is stabilizable and detectable and $G_{22} - K$ interconnection with distributed implementation is internally stable.*

**Proof** See the Appendix A.4 for proof. ■

# CHAPTER 3. PERFORMANCE CONSIDERING COMMUNICATION UNCERTAINTY IN DISTRIBUTIVE ARCHITECTURES

Apart from the issue of the possible instability of a particular distributive implementation, the sub-controller uncertainty will affect different implementations differently. Thus, it is important to incorporate the effect of sub-controller noise on the performance and to consider a power constraint on sub-controller communication. These performance objectives supplement the other performance objectives that are typically imposed on the $G - K$ interconnection of Figure 2.3.

There is thus a need to search over all possible closed-loop maps that are achievable via internally stabilizing distributive implementations. The exogenous signals and the regulated variables have to include the sub-controller noise and the transmitted signal respectively to address the internal stability and performance issues.

Let $T(K_1, K_2)$ be the closed-loop map for the generalized system shown in Figure 2.1 with $z$ and the transmitted signal $t$ as the output and $w$ and sub-controller noise $n$ as the input. Thus

$$T(K_1, K_2) : \begin{pmatrix} w \\ n \end{pmatrix} \mapsto \begin{pmatrix} z \\ t \end{pmatrix} = \begin{bmatrix} \Phi_{zw} & \Phi_{zn} \\ \Phi_{tw} & \Phi_{tn} \end{bmatrix}, \tag{3.1}$$

where $\Phi_{zw}$ captures the standard performance requirement of system, $\Phi_{zn}$ captures the effect of sub-controller communication noise on the performance and $\Phi_{tw}$ captures the power of transmitted signal with respect to power of external input signals.

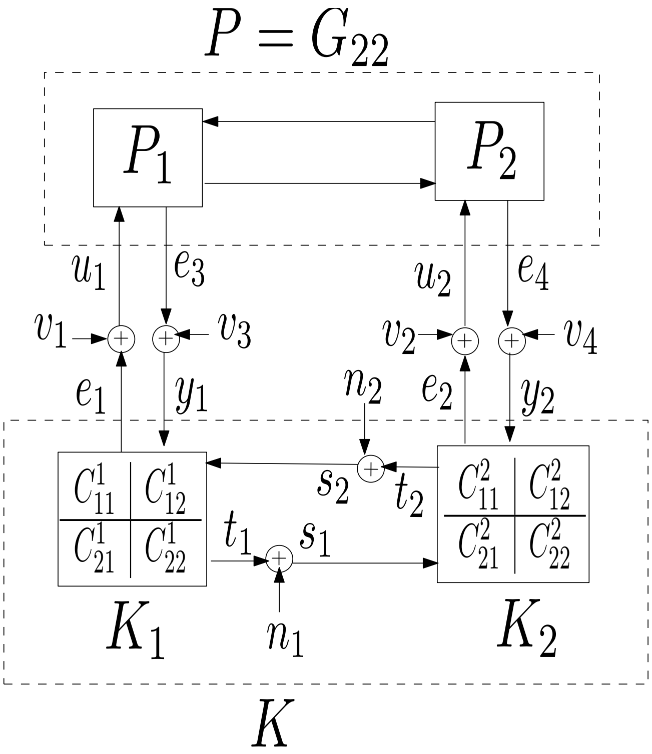


Figure 3.1 The controller with communication noise $n_1$ and $n_2$

### 3.1 Problem statement for obtaining optimal distributed implementation

The objective is to obtain a stabilizing distributed controller implementation $K_1,\ K_2$ such that it minimizes a measure of the closed-loop map $T(K_1, K_2)$. The performance optimization problem of interest can be written as follows:

$$\mu \quad := \quad \underbrace{\qquad\qquad \inf \qquad\qquad}_{K_1 - K_2\text{- is internally stabilizing}} \|T(K_1, K_2)\| \tag{3.2}$$

where $\|.\|$ is a suitable norm.

Consider the $G_{22} - K_1 - K_2$ interconnection shown in Figure 3.1 with communication noise corrupting the transmitted signals where the overall controller $K$ is implemented distributively using two sub-controllers $K_1$ and $K_2$. Let, $K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right]$ and $K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right]$ such that:

$$\begin{pmatrix} e_1 \\ t_1 \end{pmatrix} = \begin{bmatrix} C^1_{11} & C^1_{12} \\ C^1_{21} & C^1_{22} \end{bmatrix} \begin{pmatrix} y_1 \\ s_2 \end{pmatrix}, \tag{3.3}$$

$$\begin{pmatrix} e_2 \\ t_2 \end{pmatrix} = \begin{bmatrix} C^2_{11} & C^2_{12} \\ C^2_{21} & C^2_{22} \end{bmatrix} \begin{pmatrix} y_2 \\ s_1 \end{pmatrix}, \tag{3.4}$$

where $s_1$ and $s_2$ are received signals at $K_1$ and $K_2$, $t_1$ and $t_2$ are transmitted signals which are getting corrupted by additive communication noise $n_1$ and $n_2$, respectively i.e. $s_1 = t_1 + n_1$ and $s_2 = t_2 + n_2$. Note that the dimensions of these sub-controller signals will determine the actual number of signals to be communicated between sub-controllers. In other words, sizes of sub-controller matrices $C^1_{12}$, $C^1_{21}$, $C^1_{22}$, $C^2_{12}$, $C^2_{21}$, and $C^2_{22}$ in conformation with their definition given by (3.3) and (3.4) will determine the dimensions of these sub-controller signals. This is not known apriori and is a part of the design requirement.

The overall controller $K$ can be obtained in terms of sub-controllers $C^1_{11}$, $C^1_{12}$, $C^1_{21}$, $C^1_{22}$, $C^2_{11}$, $C^2_{12}$, $C^2_{21}$, and $C^2_{22}$. Let,

$$K = \left[\begin{array}{c|c} K_{11} & K_{12} \\ \hline K_{21} & K_{22} \end{array}\right], \text{ with } \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

where:

$$\begin{aligned} K_{11} &= C^1_{11} + C^1_{12}(1 - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{21} && (3.5) \\ K_{12} &= C^1_{12}(1 - C^2_{22}C^1_{22})^{-1}C^2_{21} && (3.6) \\ K_{21} &= C^2_{12}(1 - C^1_{22}C^2_{22})^{-1}C^1_{21} && (3.7) \\ K_{22} &= C^2_{11} + C^2_{12}(1 - C^1_{22}C^2_{22})^{-1}C^1_{22}C^2_{21}. && (3.8) \end{aligned}$$

In Figure 3.1 the closed-loop maps from noise $n = (n_1', n_2')^T$ to internal variables $u = (u_1', u_2')^T, y = (y_1', y_2')^T$ and $t = (t_1', t_2')^T$ are given by:

$$\begin{aligned} \Phi_{un} &= (I - KG_{22})^{-1}K_n && (3.9) \\ \Phi_{yn} &= G_{22}(I - KG_{22})^{-1}K_n && (3.10) \\ \Phi_{tn} &= K_t G_{22}(I - KG_{22})^{-1}K_n + K_{tn}, && (3.11) \end{aligned}$$

respectively, where $K_n$ is a map from $n$ to $(e_1', e_2')^T$, $K_t$ is a map from $y$ to $t$ and $K_{tn}$ is a map from $n$ to $t$. These maps are given by:

$$K_n = \begin{bmatrix} C_{12}^1(1 - C_{22}^2 C_{22}^1)^{-1} C_{22}^2 & C_{12}^1(1 - C_{22}^2 C_{22}^1)^{-1} \\ C_{12}^2(1 - C_{22}^1 C_{22}^2)^{-1} & C_{12}^2(1 - C_{22}^1 C_{22}^2)^{-1} C_{22}^1 \end{bmatrix}, \tag{3.12}$$

$$K_t = \begin{bmatrix} (1 - C_{22}^1 C_{22}^2)^{-1} C_{21}^1 & (1 - C_{22}^1 C_{22}^2)^{-1} C_{22}^1 C_{21}^2 \\ (1 - C_{22}^2 C_{22}^1)^{-1} C_{22}^2 C_{21}^1 & (1 - C_{22}^2 C_{22}^1)^{-1} C_{21}^2 \end{bmatrix}, \tag{3.13}$$

$$K_{tn} = \begin{bmatrix} (1 - C_{22}^1 C_{22}^2)^{-1} C_{22}^1 C_{22}^2 & (1 - C_{22}^1 C_{22}^2)^{-1} C_{22}^1 \\ (1 - C_{22}^2 C_{22}^1)^{-1} C_{22}^2 & (1 - C_{22}^2 C_{22}^1)^{-1} C_{22}^2 C_{22}^1 \end{bmatrix}. \tag{3.14}$$

The closed-loop map form external signals $v = (v_1', v_2', v_3', v_4')^T$ to the internal variable $t$ at the site of noise injection in Figure 3.1 is given by:

$$\Phi_{tv} = K_t \begin{bmatrix} (I - G_{22}K)^{-1} G_{22} & (I - G_{22}K)^{-1} \end{bmatrix}. \tag{3.15}$$

See the Appendix A.5, A.6 and A.7 for derivation of (3.5)-(3.15). The closed-loop map $T$ given by (3.1) consists of two maps due to communication uncertainty other than $\Phi_{tn}$ viz. $\Phi_{zn}$ and $\Phi_{tw}$. These additional maps are given by:

$$\Phi_{zn} = G_{12} \Phi_{un} \tag{3.16}$$

$$\Phi_{tw} = K_t (I - G_{22}K)^{-1} G_{21}. \tag{3.17}$$

In the standard robust control setup of Figure 2.3, all closed-loop maps achievable via stabilizing controllers can be parameterized affinely in terms of the Youla parameter $Q$, according to the following lemma.

**Lemma 3.1.1** *(9) Suppose the plant $G_{22}$ shown in Figure 2.4 that maps the control input $u = (u_1', u_2')^T$ to the measured output $y = (y_1', y_2')^T$ has a double-coprime factorization given by eight stable parameters $Y_r$, $M_r$, $X_r$, $N_r$, $X_\ell$, $N_\ell$, $Y_\ell$, and $M_\ell$ satisfying the following identity*

$$\begin{pmatrix} X_l & -Y_l \\ -N_l & M_l \end{pmatrix} \begin{pmatrix} M_r & Y_r \\ N_r & X_r \end{pmatrix} = I \tag{3.18}$$

*such that following statements are equivalent:*

- *$K$ is internally stabilizing for the interconnection shown in Figure 2.4*

- *There exists a stable $Q$ such that*

$$K = (Y_r - M_r Q)(X_r - N_r Q)^{-1} = (X_\ell - QN_\ell)^{-1}(Y_\ell - QM_\ell)$$

*Under the above parametrization of stabilizing controllers, it can be shown that a closed-loop maps $\Phi_{zw}$ is achievable via stabilizing controllers if and only if*

$$\Phi_{zw} \in \{H - UQV|\ Q \ \mathit{stable}\}$$

*where $H,\ U,$ and $V$ are stable transfer matrices determinable from $G$.*

∎

Thus, from above lemma and by using (3.5)-(3.8), sub-controllers $C^1_{11}$, $C^1_{12}$, $C^1_{21}$, $C^1_{22}$, $C^2_{11}$, $C^2_{12}$, $C^2_{21}$, and $C^2_{22}$ can be parameterized in terms of $Q$ by solving following equation:

$$\begin{aligned}\left[\begin{array}{c|c} C^1_{11} + C^1_{12}(1 - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{21} & C^1_{12}(1 - C^2_{22}C^1_{22})^{-1}C^2_{21} \\ \hline C^2_{12}(1 - C^1_{22}C^2_{22})^{-1}C^1_{21} & C^2_{11} + C^2_{12}(1 - C^1_{22}C^2_{22})^{-1}C^1_{22}C^2_{21} \end{array}\right] \\ = (Y_r - M_r Q)(X_r - N_r Q)^{-1} \\ = (X_\ell - QN_\ell)^{-1}(Y_\ell - QM_\ell) \qquad (3.19)\end{aligned}$$

With the help of discussion presented above, the main performance optimization problem for distributed implementation can be summarized in following theorem:

**Theorem 3.1.1** *The optimal distributed controller implementation in terms of sub-controllers $C^1_{11}$, $C^1_{12}$, $C^1_{21}$, $C^1_{22}$, $C^2_{11}$, $C^2_{12}$, $C^2_{21}$, and $C^2_{22}$ which minimizes a given measure of the perfor-*

*mance map $T$ is obtained by solving following optimization problem:*

$$\mu \quad := \overbrace{\begin{array}{l} T(Q) = \begin{bmatrix} \Phi_{zw}(Q) & \Phi_{zn}(Q) \\ \Phi_{tw}(Q) & \Phi_{tn}(Q) \end{bmatrix}, \\ \Phi_{zw}(Q) = H - UQV, \\ \Phi_{zn}(Q) = G_{12}(I - K(Q)G_{22})^{-1}K_n(Q), \\ \Phi_{tw}(Q) = K_t(Q)(I - G_{22}K(Q))^{-1}G_{21}, \\ \Phi_{tn}(Q) = K_t(Q)G_{22}(I - K(Q)G_{22})^{-1}K_n(Q) + K_{tn}(Q), \\ C_{ij}^k(Q) \text{ satisfy } (3.19) \text{ for } k = 1,2; i = 1,2; j = 1,2 \\ Q \text{ - stable} \\ C_{ij}^k \text{ - internally stabilizes distributed interconnection} \end{array}}^{\inf} \|T(Q)\| \qquad (3.20)$$

∎

The optimization variables in Theorem 3.1.1 are sub-controllers $C_{ij}^k$ satisfying (3.19) and Youla parameter $Q$. Since, there is no unique solution to (3.19), sub-controllers $C_{ij}^k$ can be parameterized in terms of $Q$ in more than one way. As discussed earlier in this section, the dimensions of all these sub-controller transfer matrices other than $C_{11}^1$ and $C_{11}^2$ are not fixed, which increase the complexity of solving (3.19) to obtain a parametrization for sub-controllers in terms of $Q$. Even after solving (3.19) to obtain $C_{ij}^k(Q)$, there is no guarantee that the closed-loop map $T(Q)$ will be stable and affine in $Q$ because the additional maps due to communication uncertainty viz. $\Phi_{tw}$, $\Phi_{zn}$ and $\Phi_{tn}$ are in general not stable and affine in $Q$. Thus, a pivotal issue is the identification of signals to be transmitted between sub-controllers such that distributed implementation is internally stable and all closed-loop maps including the above mentioned maps are affine in Youla parameter $Q$. Within such distributed architectures the performance problem becomes a convex problem at the cost of being suboptimal to the main optimization problem given by Theorem 3.1.1.

## 3.2 Performance problem - suboptimal but convex

In this section, design of distributive implementation of controllers which have no structural constraints is presented such that the performance problem in presence of sub-controller noise is convex at the expense of being suboptimal.

As mentioned in the following lemma, in order to have $T(Q)$ being affine in $Q$, it suffices to show that the maps $\Phi_{un}$, $\Phi_{tv}$ and $\Phi_{tn}$ are affine in the Youla parameter $Q$.

**Lemma 3.2.1** *The closed-loop map $T(K_1, K_2)$ corresponding to Figure 2.1 interconnection is affine in the Youla parameter $Q$ if the maps $\Phi_{un}$, $\Phi_{tv}$ and $\Phi_{tn}$ are affine in $Q$.*

Let, $\overline{Y}_r = Y_r - M_r Q$, $\overline{Y}_l = Y_l - QM_l$, $\overline{X}_r = X_r - N_r Q$ and $\overline{X}_l = X_l - QN_l$. Note that $\overline{Y}_r$, $\overline{Y}_l$, $\overline{X}_r$ and $\overline{X}_l$ are stable and affine in $Q$. Using above notation and parameterization, the following sufficient condition is formulated to design sub-controllers for the system that can be implemented distributively as shown in Figure 3.1 such that all closed-loop maps are stable and affine in $Q$.

**Theorem 3.2.1** *Consider plant $G_{22}$ shown in Figure 3.1 that has double-coprime factorization given by Lemma 3.1.1 with $K$ being a stabilizing controller parameterized in terms of $Q$. $K_n, K_t$ and $K_{tn}$ are derived from $K$ as given by (3.12)-(3.14). Let $T_a = \overline{X}_l K_n, T_b = K_t \overline{X}_r$, and $\Phi_{tn} = K_t G_{22}(I - KG_{22})^{-1} K_n + K_{tn}$. Then the closed-loop map $T(K)$ given by (3.1) for the distributive implementation as shown in Figure 3.1 is stable and affine in $Q$ if $T_a, T_b$ and $\Phi_{tn}$ are stable and affine in $Q$.*

**Proof** From (18), $\Phi_{zw}$ is stable and affine in $Q$. By using the fact that $(I - G_{22}K)^{-1} = \overline{X}_r M_l, (I - G_{22}K)^{-1} G_{22} = \overline{X}_r N_l, (I - KG_{22})^{-1} = M_r \overline{X}_l$ and $G_{22}(I - KG_{22})^{-1} = N_r \overline{X}_l$, the closed-loop maps given by (3.9)-(3.11) and (3.15) are written as follows:

$$\Phi_{un} = M_r \overline{X}_l K_n \tag{3.21}$$

$$\Phi_{yn} = N_r \overline{X}_l K_n \tag{3.22}$$

$$\Phi_{tn} = K_t N_r \overline{X}_l K_n + K_{tn} \tag{3.23}$$

$$\Phi_{tv} = K_t \begin{bmatrix} \overline{X}_r N_l & \overline{X}_r M_l \end{bmatrix}. \tag{3.24}$$

Since, $T_a = \overline{X}_l K_n$ and $T_b = K_t \overline{X}_r$. Above four closed-loop maps can be rewritten in terms of $T_a, T_b$ and $K_{tn}$ as follows:

$$\Phi_{un} = M_r T_a \tag{3.25}$$

$$\Phi_{yn} = N_r T_a \tag{3.26}$$

$$\Phi_{tn} = T_b \overline{X}_r^{-1} N_r T_a + K_{tn} \tag{3.27}$$

$$\Phi_{tv} = \begin{bmatrix} T_b N_l & T_b M_l \end{bmatrix}. \tag{3.28}$$

Since, $M_r, N_r, M_l$ and $N_l$ are stable and constant matrices, in order to find distributive implementation for controllers such that all closed-loop maps are stable and affine in $Q$, one must be able to find sub-controllers $K_1$ and $K_2$ such that $T_a$, $T_b$ and $\Phi_{tn}$ are stable and affine in $Q$. ■

This provides a sufficient condition for guaranteing that sub-controllers $C_{ij}^k$ internally stabilizes the distributed interconnection. Using this a suboptimal problem to the main optimization problem can be formulated such that it is convex.

**Theorem 3.2.2** *The distributed controller implementation in terms of sub-controllers $C_{11}^1$, $C_{12}^1$, $C_{21}^1$, $C_{22}^1$, $C_{11}^2$, $C_{12}^2$, $C_{21}^2$, and $C_{22}^2$ which minimizes some measure of the performance map $T$*

*is obtained by solving following optimization problem:*

$$\mu_1 \;:=\; \underbrace{\inf}_{\begin{array}{l} T(Q) = \begin{bmatrix} \Phi_{zw}(Q) & \Phi_{zn}(Q) \\ \Phi_{tw}(Q) & \Phi_{tn}(Q) \end{bmatrix}, \\ \Phi_{zw}(Q) = H - UQV, \\ \Phi_{zn}(Q) = G_{12}M_rT_a(Q), \\ \Phi_{tw}(Q) = T_b(Q)M_lG_{21}, \\ \Phi_{tn}(Q) = T_b(Q)\overline{X(Q)}_r^{-1}N_rT_a(Q) + K_{tn}(Q), \\ T_a(Q) = \overline{X}_l(Q)K_n(Q), \\ T_b(Q) = K_t(Q)\overline{X}_r(Q), \\ C_{ij}^k(Q) \textit{ satisfy } (3.19) \textit{ for } k=1,2; i=1,2; j=1,2 \\ T_a(Q), T_b(Q), \Phi_{tn}(Q) \;\; \textit{- stable and affine in } Q \\ Q \textit{ - stable} \end{array}} \|T(Q)\| \tag{3.29}$$

*The above optimization problem is convex with $\mu_1 \geq \mu$.*

$\blacksquare$

With this result, the impetus will be on obtaining $C_{ij}^k(Q)$ satisfying (3.19) for $k = 1, 2; i = 1, 2; j = 1, 2$ such that $T_a(Q), T_b(Q)$ and $\Phi_{tn}(Q)$ are stable and affine in $Q$. Towards this, let $K_0 = \begin{bmatrix} C_{11}^1 & 0 \\ 0 & C_{11}^2 \end{bmatrix}$ and by using definitions for $K_n$ and $K_t$ given by (3.12)-(3.13), the controller $K$ can be written in following two ways as (see Appendix A.6 for details):

$$K = K_0 + K_n \begin{bmatrix} C_{21}^1 & 0 \\ 0 & C_{21}^2 \end{bmatrix}, \qquad \text{and} \tag{3.30}$$

$$K = K_0 + \begin{bmatrix} 0 & C_{12}^1 \\ C_{12}^2 & 0 \end{bmatrix} K_t. \tag{3.31}$$

Since, $n = (n_1', n_2')'$, $K_n$ can be written as $K_n = \begin{bmatrix} K_{n1} & K_{n2} \end{bmatrix}$, i.e.

$$T_a = \overline{X}_l K_n = \begin{bmatrix} \overline{X}_l K_{n1} & \overline{X}_l K_{n2} \end{bmatrix} =: \begin{bmatrix} T_{a1} & T_{a2} \end{bmatrix}. \tag{3.32}$$

Similarly with $t = (t_1', t_2')'$, $K_t$ can be written as $K_t = \begin{bmatrix} K_{t1} \\ K_{t2} \end{bmatrix}$, i.e.

$$T_b = K_t\overline{X}_r = \begin{bmatrix} K_{t1}\overline{X}_r \\ K_{t2}\overline{X}_r \end{bmatrix} =: \begin{bmatrix} T_{b1} \\ T_{b2} \end{bmatrix}. \tag{3.33}$$

Using (3.32), (3.30) can be further simplified as follows:

$$K - K_0 \quad = \quad K_n \begin{bmatrix} C_{21}^1 & 0 \\ 0 & C_{21}^2 \end{bmatrix}.$$

Therefore,

$$\overline{X}_l(K - K_0) \quad = \quad \overline{X}_l K_n \begin{bmatrix} C_{21}^1 & 0 \\ 0 & C_{21}^2 \end{bmatrix}.$$

This implies that

$$\overline{X}_l(\overline{X}_l^{-1}\overline{Y}_l - K_0) \quad = \quad T_a \begin{bmatrix} C_{21}^1 & 0 \\ 0 & C_{21}^2 \end{bmatrix},$$

therefore,

$$(\overline{Y}_l - \overline{X}_l K_0) \quad = \quad \begin{bmatrix} T_{a1}C_{21}^1 & T_{a2}C_{21}^2 \end{bmatrix}. \tag{3.34}$$

Similarly, using (3.33), (3.31) can be further simplified to:

$$(Y_r - K_0 X_r) \quad = \quad \begin{bmatrix} C_{12}^1 T_{b2} \\ C_{12}^2 T_{b1} \end{bmatrix}. \tag{3.35}$$

It is not clear how to solve (3.34) and (3.35) for all possible solutions. However, by fixing $K_0$, one solution for each (3.34) and (3.35) can be obtained, leading to two stabilizing distributed implementations with convex performance problem as discussed in the next section.

## 3.3 Two stabilizing distributed implementation with convex performance problem

Coprime factors of $G_{22}$ are partitioned in conformation with partitioning of $K$ in Figure 3.1 $\overline{X}_l = \begin{pmatrix} \widetilde{X}_{11} & \widetilde{X}_{12} \\ \widetilde{X}_{21} & \widetilde{X}_{22} \end{pmatrix}$ ; $\overline{Y}_l = \begin{pmatrix} \widetilde{Y}_{11} & \widetilde{Y}_{12} \\ \widetilde{Y}_{21} & \widetilde{Y}_{22} \end{pmatrix}$ ; $N_l = \begin{pmatrix} \widetilde{N}_{11} & \widetilde{N}_{12} \\ \widetilde{N}_{21} & \widetilde{N}_{22} \end{pmatrix}$ ; $M_l = \begin{pmatrix} \widetilde{M}_{11} & \widetilde{M}_{12} \\ \widetilde{M}_{21} & \widetilde{M}_{22} \end{pmatrix}$ ; $\overline{X}_r = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$ ; $\overline{Y}_r = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix}$ ; $N_r = \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix}$ and $M_r = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$. $T_a$ and $T_b$ can be further partitioned in confirmation with $\overline{X}_l$ and $X_r$, respectively as $T_a = \begin{pmatrix} T^u_{a1} & T^u_{a2} \\ T^d_{a1} & T^d_{a2} \end{pmatrix}$ and $T_b = \begin{pmatrix} T^l_{b1} & T^r_{b1} \\ T^l_{b2} & T^r_{b2} \end{pmatrix}$.

Consider the case when two communication channels for transmission in each direction are used between sub-controllers i.e. $t_1 = \begin{pmatrix} t_{11} \\ t_{12} \end{pmatrix}$, $t_2 = \begin{pmatrix} t_{21} \\ t_{22} \end{pmatrix}$, $n_1 = \begin{pmatrix} n_{11} \\ n_{12} \end{pmatrix}$ and $n_2 = \begin{pmatrix} n_{21} \\ n_{22} \end{pmatrix}$. In this special case, following two corollaries present two stabilizing distributed implementations with convex performance problem.

**Corollary 3.3.1** *Left-coprime architecture:*

*The two sub-controllers $K_1$ and $K_2$ using two communication channel architecture as given by following equations:*

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & \widetilde{X}_{11}^{-1}\widetilde{Y}_{12} & -\widetilde{X}_{11}^{-1}\widetilde{X}_{12} \\ \hline I & 0 & 0 \\ \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & \widetilde{X}_{11}^{-1}\widetilde{Y}_{12} & -\widetilde{X}_{11}^{-1}\widetilde{X}_{12} \end{array}\right] \tag{3.36}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{22}^{-1}\widetilde{Y}_{22} & \widetilde{X}_{22}^{-1}\widetilde{Y}_{21} & -\widetilde{X}_{22}^{-1}\widetilde{X}_{21} \\ \hline I & 0 & 0 \\ \widetilde{X}_{22}^{-1}\widetilde{Y}_{22} & \widetilde{X}_{22}^{-1}\widetilde{Y}_{21} & -\widetilde{X}_{22}^{-1}\widetilde{X}_{21} \end{array}\right] \tag{3.37}$$

*satisfies parametrization given by (3.19) and is such that $T_a, T_b$ and $\Phi_{tn}$ are stable and affine in $Q$. This architecture given by (3.36)-(3.37) is called left-coprime architecture. and is shown in*

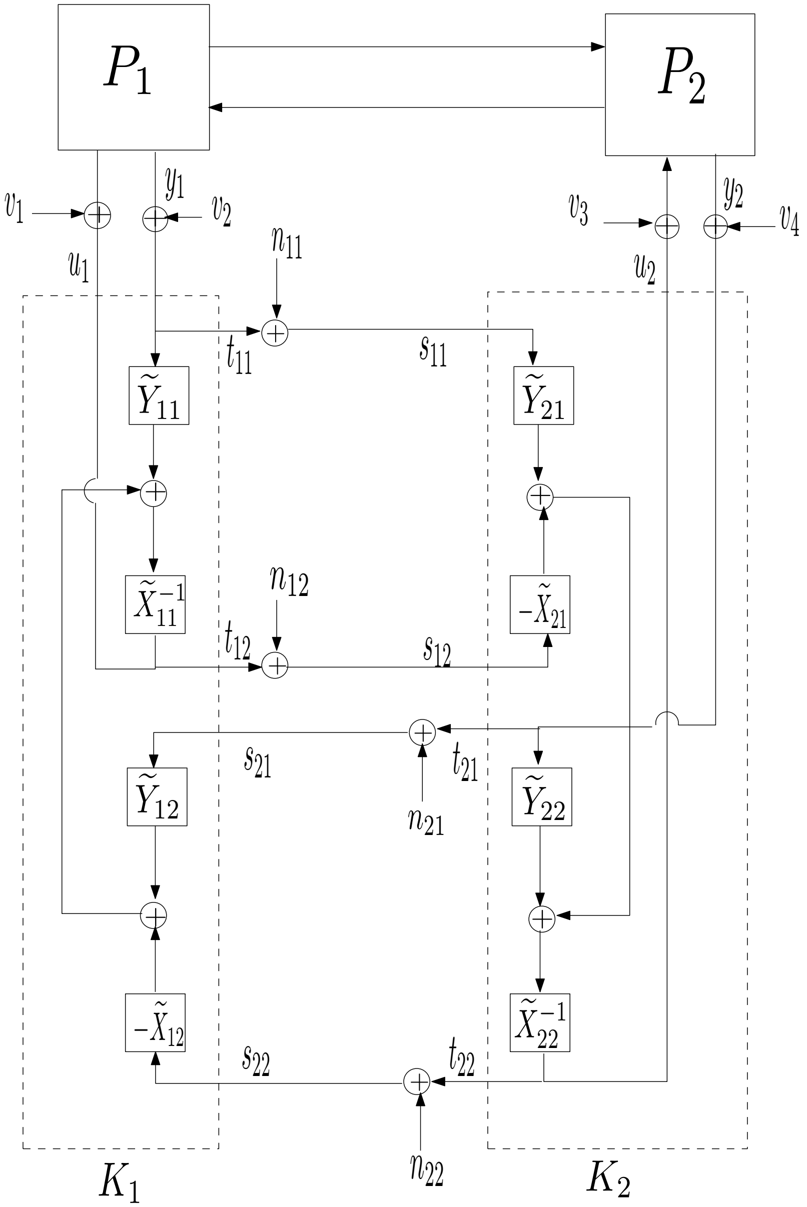


*Figure 3.2 Left-coprime architecture*

*Figure 3.2. The transmitted signals in this architecture are* $t = (t_1', t_2')' = (t_{11}', t_{12}', t_{21}', t_{22}')' = (y_1', u_1', y_2', u_2')'$.

**Proof** It can be verified that sub-controllers $C_{11}^1$, $C_{12}^1$, $C_{21}^1$, $C_{22}^1$, $C_{11}^2$, $C_{12}^2$, $C_{21}^2$, and $C_{22}^2$ satisfies (3.19). $T_a, T_b$ and $\Phi_{tn}$ for this architecture are given by:

$$T_a = \begin{bmatrix} 0 & -\widetilde{X}_{12} & 0 & \widetilde{Y}_{12} \\ -\widetilde{X}_{21} & 0 & \widetilde{Y}_{21} & 0 \end{bmatrix} \Pi^T \tag{3.38}$$

$$T_b = \Pi \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \\ X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \tag{3.39}$$

$$\Phi_{tn} = \Pi \begin{bmatrix} M_r \\ N_r \end{bmatrix} T_a \tag{3.40}$$

which are stable and affine in $Q$. $\Pi$ is the permutation matrix given by $\Pi = \begin{bmatrix} 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & I & 0 & 0 \end{bmatrix}$. ■

The left-coprime architecture obtained in above corollary can be used to reduce the convex performance optimization problem given in Theorem 3.2.2 to the following suboptimal problem:

$$\gamma_L := \underbrace{\inf}_{\substack{\text{Left-coprime architecture} \\ Q \text{ - stable}}} \|T(Q)\| \tag{3.41}$$

$$= \underbrace{\inf}_{Q \text{ - stable}} \left\| \begin{bmatrix} H - U * Q * V & H_1^L - U_1^L * Q * V_1^L \\ H_2^L - U_2^L * Q * V_2^L & H_3^L - U_3^L * Q * V_3^L \end{bmatrix} \right\| \tag{3.42}$$

where $H_1^L, U_1^L, V_1^L, H_2^L, U_2^L, V_2^L, H_3^L, U_3^L$ and $V_3^L$ are determined based on left-coprime architecture.

Another set of sub-controllers can be obtained using right-coprime factors as discussed in following corollary.

**Corollary 3.3.2** *Right-coprime architecture:*

*The two sub-controllers $K_1$ and $K_2$ using two communication channel architecture as given by following equations:*

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{11}X_{11}^{-1} & I & Y_{11}X_{11}^{-1} \\ \hline Y_{21}X_{11}^{-1} & 0 & Y_{21}X_{11}^{-1} \\ -X_{21}X_{11}^{-1} & 0 & -X_{21}X_{11}^{-1} \end{array}\right] \tag{3.43}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{22}X_{22}^{-1} & I & Y_{22}X_{22}^{-1} \\ \hline Y_{12}X_{22}^{-1} & 0 & Y_{12}X_{22}^{-1} \\ -X_{12}X_{22}^{-1} & 0 & -X_{12}X_{22}^{-1} \end{array}\right] \tag{3.44}$$

*satisfies parametrization given by (3.19) and is such that $T_a, T_b$ and $\Phi_{tn}$ are stable and affine in $Q$. This architecture given by (3.43)-(3.44) is called right-coprime architecture and is shown in Figure 3.3. The transmitted signal in this architecture is $t = (t_1', t_2')' = (t_{11}', t_{12}', t_{21}', t_{22}')' = (Y_{21}X_{11}^{-1}y_1', -X_{21}X_{11}^{-1}y_1', Y_{12}X_{22}^{-1}y_2', -X_{21}X_{22}^{-1}y_2')'$.*

**Proof** It can be verified that sub-controllers $C^1_{11}$, $C^1_{12}$, $C^1_{21}$, $C^1_{22}$, $C^2_{11}$, $C^2_{12}$, $C^2_{21}$, and $C^2_{22}$ satisfies (3.19). $T_a, T_b$ and $\Phi_{tn}$ for this architecture are given by:

$$T_a = \begin{bmatrix} \widetilde{X}_{11} & \widetilde{X}_{12} & \widetilde{Y}_{11} & \widetilde{Y}_{12} \\ \widetilde{X}_{21} & \widetilde{X}_{22} & \widetilde{Y}_{21} & \widetilde{Y}_{22} \end{bmatrix} \Pi \tag{3.45}$$

$$T_b = \Pi^T \begin{bmatrix} 0 & Y_{12} \\ Y_{21} & 0 \\ 0 & -X_{12} \\ -X_{21} & 0 \end{bmatrix} \tag{3.46}$$

$$\Phi_{tn} = T_b \begin{bmatrix} N_l & M_l \end{bmatrix} \Pi \tag{3.47}$$

which are stable and affine in $Q$. ∎

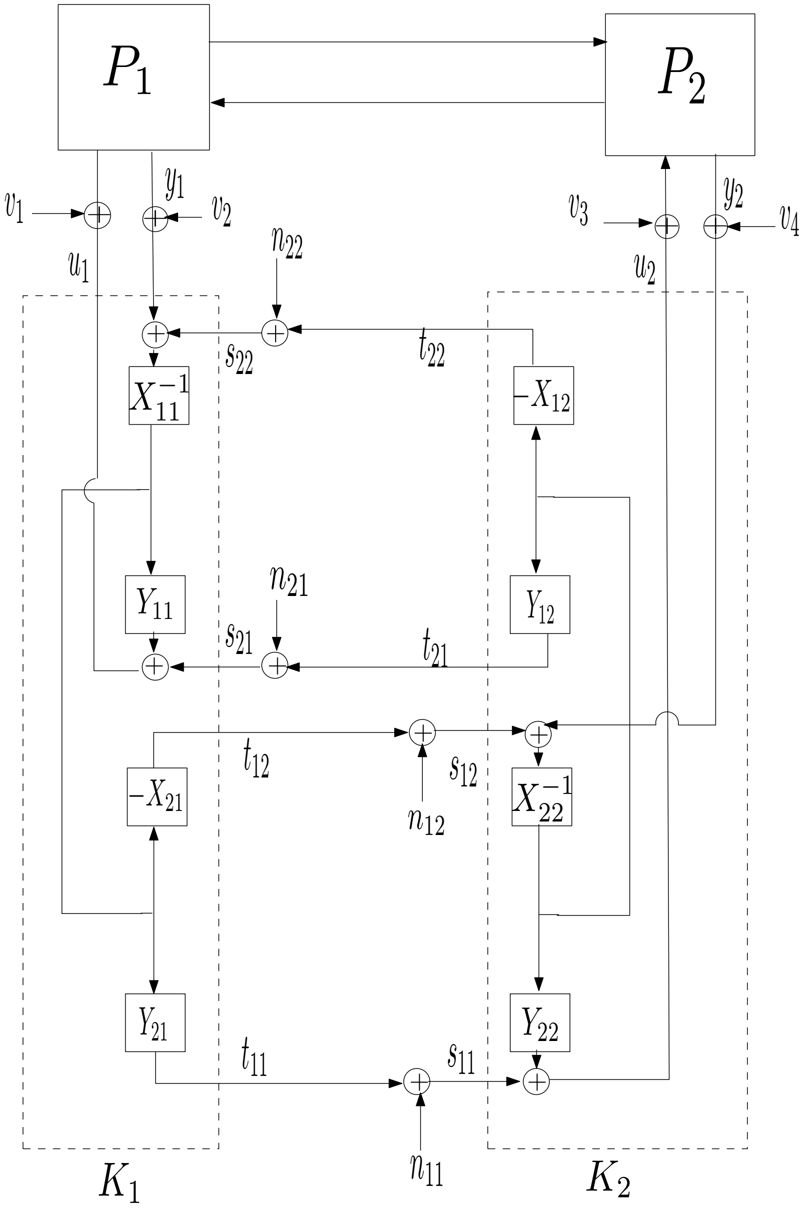


*Figure 3.3 Right-coprime architecture*

The right-coprime architecture obtained in above corollary can be used to reduce the convex performance optimization problem given in Theorem 3.2.2 to the following suboptimal problem:

$$\begin{aligned}
\gamma_R \quad &:= \underbrace{\quad\quad\quad\quad \inf \quad\quad\quad\quad}_{\substack{\text{Right-coprime architecture} \\ Q\text{ - stable}}} \|T(Q)\| &(3.48)\\
&= \underbrace{\quad\quad \inf \quad\quad}_{Q\text{ - stable}} \left\| \begin{bmatrix} H - U * Q * V & H_1^R - U_1^R * Q * V_1^R \\ H_2^R - U_2^R * Q * V_2^R & H_3^R - U_3^R * Q * V_3^R \end{bmatrix} \right\| &(3.49)
\end{aligned}$$

where $H_1^R, U_1^R, V_1^R, H_2^R, U_2^R, V_2^R, H_3^R, U_3^R$ and $V_3^R$ are determined based on left-coprime architecture.

Left-coprime and right-coprime architectures provide a subclass of controllers over which a convex search can be done to obtain controllers which can be implemented distributively, such that desired performance is met including those related to the affect of sub-controller noise and the power of sub-controller transmission signal.

This completes the construction of two architectures for distributed implementation of controller $K$ such that implementation is stable and the performance problem in presence of sub-controller communication is a convex problem.

# CHAPTER 4. DISTRIBUTED DESIGN FOR CONTROLLERS WITH SPECIAL STRUCTURE

The two architectures obtained in the previous chapter for controllers without any structure can be specialized to controllers with banded structure and nested structure.

## 4.1 Banded Structure

All banded structure controllers $K$ given by Equation (2.1) can be parametrized in terms of Youla parameter $Q$ which is also banded by making use of the following Lemma. For the sake of simplicity, $n$ is taken to be equal to 2, and the result can be generalized for any $n$.

**Lemma 4.1.1** *Let the plant $P = G_{22}$ shown in Figure 2.6 that maps the control input $u = (u'_1, u'_2)^T$ to the measured output $y = (y'_1, y'_2)^T$ be described by $(A(\lambda), B(\lambda), C(\lambda), D(\lambda))$ as discussed in Chapter 2. Assuming that $P$ has stabilizable and detectable realization, i.e. there exist $\bar{F}$ and $\bar{L}$ with the properties given above. Then, there exist stable banded parameters $Y_r$, $M_r$, $X_r$, $N_r$, $X_\ell$, $N_\ell$, $Y_\ell$, and $M_\ell$ satisfying the following identity*

$$\begin{pmatrix} X_l & -Y_l \\ -N_l & M_l \end{pmatrix} \begin{pmatrix} M_r & Y_r \\ N_r & X_r \end{pmatrix} = I$$

*such that following statements are equivalent:*

- *$K$ has banded structure and is internally stabilizing for the interconnection shown in Figure 2.6*
- *There exists a stable $Q$ with band structure such that*

$$K = (Y_r - M_r Q)(X_r - N_r Q)^{-1} = (X_\ell - QN_\ell)^{-1}(Y_\ell - QM_\ell)$$

**Proof** This is a 2−input 2−output case of generalized result on parameterization of banded strucutre controller in terms of banded structure $Q$ given in (8) ∎

This results in the parameterization of $K$ in terms of $Q$ having the same banded structure. Let $Q$ be partitioned in conformation with partitioning of $G_{22}$ and $K$:

$$Q = \begin{pmatrix} Q_{11} & \lambda Q_{12} \\ \lambda Q_{21} & Q_{22} \end{pmatrix}.$$

Let, $\overline{Y}_r = Y_r - M_r Q$, $\overline{Y}_l = Y_l - QM_l$, $\overline{X}_r = X_r - N_r Q$ and $\overline{X}_l = X_l - QN_l$. Note that $\overline{Y}_r$, $\overline{Y}_l$, $\overline{X}_r$ and $\overline{X}_l$ are affine in $Q$, stable and have banded structure. Let, $\overline{X}_l = \begin{pmatrix} \widetilde{X}_{11} & \lambda\widetilde{X}_{12} \\ \lambda\widetilde{X}_{21} & \widetilde{X}_{22} \end{pmatrix}$ ; $\overline{Y}_l = \begin{pmatrix} \widetilde{Y}_{11} & \lambda\widetilde{Y}_{12} \\ \lambda\widetilde{Y}_{21} & \widetilde{Y}_{22} \end{pmatrix}$ ; $N_l = \begin{pmatrix} \widetilde{N}_{11} & \lambda\widetilde{N}_{12} \\ \lambda\widetilde{N}_{21} & \widetilde{N}_{22} \end{pmatrix}$ ; $M_l = \begin{pmatrix} \widetilde{M}_{11} & \lambda\widetilde{M}_{12} \\ \lambda\widetilde{M}_{21} & \widetilde{M}_{22} \end{pmatrix}$ ; $\overline{X}_r = \begin{pmatrix} X_{11} & \lambda X_{12} \\ \lambda X_{21} & X_{22} \end{pmatrix}$ ; $\overline{Y}_r = \begin{pmatrix} Y_{11} & \lambda Y_{12} \\ \lambda Y_{21} & Y_{22} \end{pmatrix}$ ; $N_r = \begin{pmatrix} N_{11} & \lambda M_{12} \\ \lambda N_{21} & N_{22} \end{pmatrix}$ and $M_r = \begin{pmatrix} M_{11} & \lambda M_{12} \\ \lambda M_{21} & M_{22} \end{pmatrix}$.

Substitute these coprime factors in the architectures obtained in previous Chapter to obtain two ways of implementing banded structured controller such that the performance problem is a convex problem in presence of sub-controller noise. Thus, the two sub-controllers with left-coprime architecture are given by

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & \lambda\widetilde{X}_{11}^{-1}\widetilde{Y}_{12} & -\lambda\widetilde{X}_{11}^{-1}\widetilde{X}_{12} \\ \hline I & 0 & 0 \\ \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & \lambda\widetilde{X}_{11}^{-1}\widetilde{Y}_{12} & -\lambda\widetilde{X}_{11}^{-1}\widetilde{X}_{12} \end{array}\right] \tag{4.1}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{22}^{-1}\widetilde{Y}_{22} & \lambda\widetilde{X}_{22}^{-1}\widetilde{Y}_{21} & -\lambda\widetilde{X}_{22}^{-1}\widetilde{X}_{21} \\ \hline I & 0 & 0 \\ \widetilde{X}_{22}^{-1}\widetilde{Y}_{22} & \lambda\widetilde{X}_{22}^{-1}\widetilde{Y}_{21} & -\lambda\widetilde{X}_{22}^{-1}\widetilde{X}_{21} \end{array}\right] \tag{4.2}$$

and is shown in Figure 4.1.

Another architecture is obtained by substituting coprime factors in right-coprime architecture. Thus, the two sub-controllers of right-coprime architecture for banded structure controller

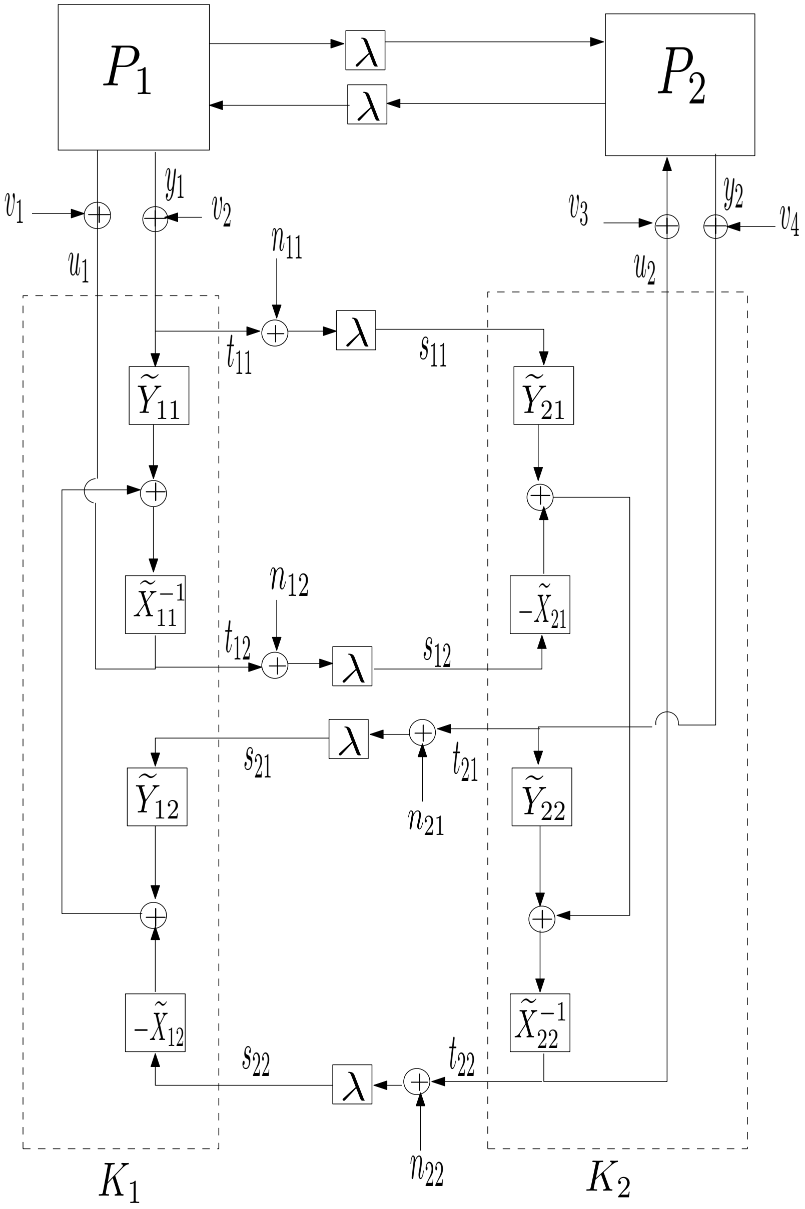


Figure 4.1 Left-coprime architecture for banded structure

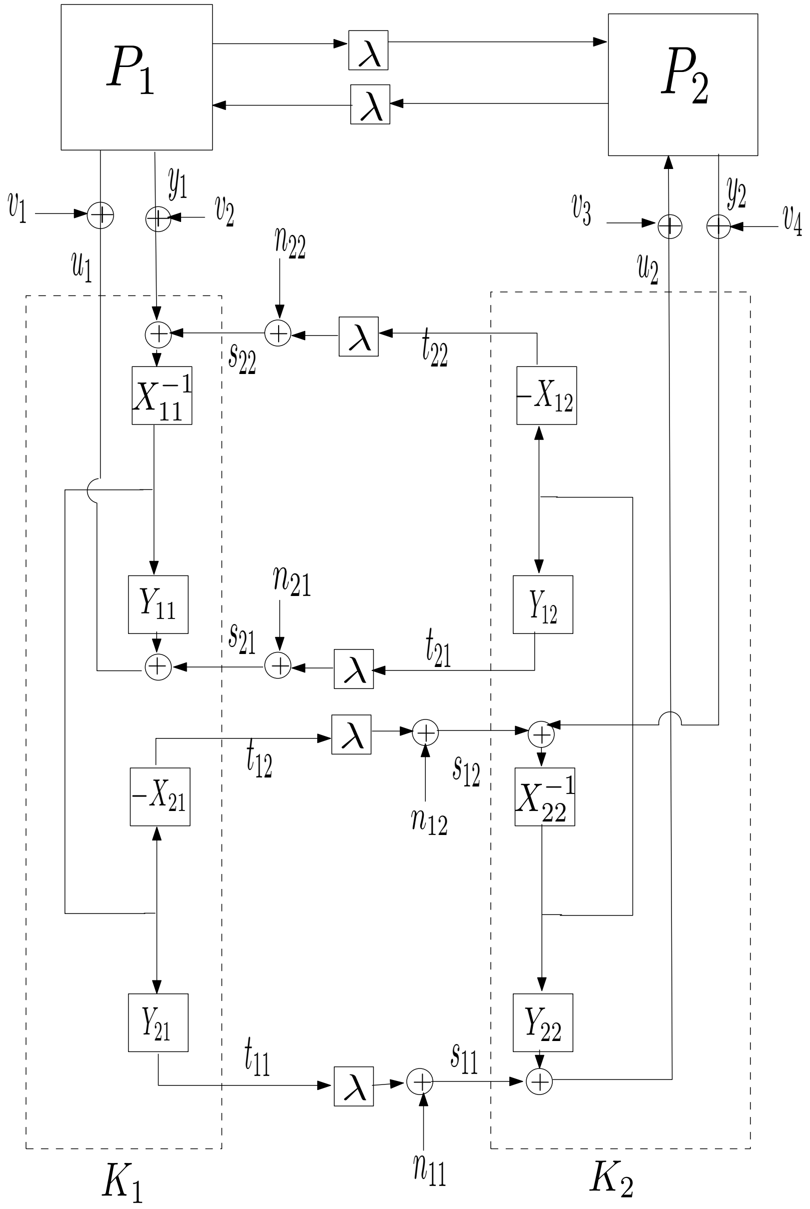


Figure 4.2 Right-coprime architecture for banded structure

are given by

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{11}X_{11}^{-1} & I & Y_{11}X_{11}^{-1} \\ \hline \lambda Y_{21}X_{11}^{-1} & 0 & \lambda Y_{21}X_{11}^{-1} \\ -\lambda X_{21}X_{11}^{-1} & 0 & -\lambda X_{21}X_{11}^{-1} \end{array}\right] \tag{4.3}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{22}X_{22}^{-1} & I & Y_{22}X_{22}^{-1} \\ \hline \lambda Y_{12}X_{22}^{-1} & 0 & \lambda Y_{12}X_{22}^{-1} \\ -\lambda X_{12}X_{22}^{-1} & 0 & -\lambda X_{12}X_{22}^{-1} \end{array}\right] \tag{4.4}$$

and is shown in Figure 4.2.

## 4.2 Nested Structure

All stabilizing nested controllers $K$ as shown in Figure 2.5 can be parameterized using Youla-Kucera parameter $Q$. The following result translates the triangular structure restriction on the controller to the same structure on the Youla parameter $Q$. Once again for the sake of simplicity, $n$ is taken to be equal to 2, and the result can be generalized for any $n$.

**Lemma 4.2.1** *Consider 2-nest $G_{22}-K$ system shown in Figure 2.5, where $G_{22} = \left[\begin{array}{c|c} G_{22a} & 0 \\ \hline G_{22c} & G_{22d} \end{array}\right]$ $:= P$ maps control inputs $u = (u_1', u_2')'$ to the measured output $y = (y_1', y_2')'$. Assume that $P_1 = \left[\begin{array}{cc} G_{22a} & 0 \end{array}\right]$ and $P_2 = \left[\begin{array}{cc} G_{22c} & G_{22d} \end{array}\right]$ have state space realizations $\left[\begin{array}{c|cc} A_1 & B_{11} & 0 \\ \hline C_1 & D_{11} & 0 \end{array}\right]$ and $\left[\begin{array}{c|cc} A_2 & B_{21} & B_{22} \\ \hline C_2 & D_{21} & D_{22} \end{array}\right]$, respectively, and the inherited realizations of $G_{22a}$ and $G_{22d}$ are stabilizable and detectable. Then there exist stable lower triangular parameters $Y_r$, $M_r$, $X_r$, $N_r$, $X_\ell$, $N_\ell$, $Y_\ell$, and $M_\ell$ satisfying the following identity*

$$\begin{pmatrix} X_l & -Y_l \\ -N_l & M_l \end{pmatrix} \begin{pmatrix} M_r & Y_r \\ N_r & X_r \end{pmatrix} = I \tag{4.5}$$

*such that the following statements are equivalent:*

- *$K$ is lower triangular and it internally stabilizes the $G_{22} - K$ inter-connection.*

- *there exists a stable $Q$ that is lower triangular such that*

$$K = (Y_r - M_r Q)(X_r - N_r Q)^{-1} = (X_\ell - QN_\ell)^{-1}(Y_\ell - QM_\ell)$$

**Proof** This is a 2−input 2−output case of generalized result on parameterization of lower triangular controller in terms of lower triangular $Q$ parameter given in (8) ■

This results in the parameterization of $K$ in terms of $Q$ having the same structure. Let $Q$ be partitioned according to the structure of $G_{22}$ and $K$:

$$Q = \begin{pmatrix} Q_{11} & 0 \\ Q_{21} & Q_{22} \end{pmatrix}.$$

Let, $\overline{Y}_r = Y_r - M_r Q$, $\overline{Y}_l = Y_l - QM_l$, $\overline{X}_r = X_r - N_r Q$ and $\overline{X}_l = X_l - QN_l$. Note that $\overline{Y}_r$, $\overline{Y}_l$, $\overline{X}_r$ and $\overline{X}_l$ are affine in $Q$, stable and have lower triangular structure. Let, $\overline{X}_l = \begin{pmatrix} \widetilde{X}_{11} & 0 \\ \widetilde{X}_{21} & \widetilde{X}_{22} \end{pmatrix}$; $\overline{Y}_l = \begin{pmatrix} \widetilde{Y}_{11} & 0 \\ \widetilde{Y}_{21} & \widetilde{Y}_{22} \end{pmatrix}$; $N_l = \begin{pmatrix} \widetilde{N}_{11} & 0 \\ \widetilde{N}_{21} & \widetilde{N}_{22} \end{pmatrix}$; $M_l = \begin{pmatrix} \widetilde{M}_{11} & 0 \\ \widetilde{M}_{21} & \widetilde{M}_{22} \end{pmatrix}$; $\overline{X}_r = \begin{pmatrix} X_{11} & 0 \\ X_{21} & X_{22} \end{pmatrix}$; $\overline{Y}_r = \begin{pmatrix} Y_{11} & 0 \\ Y_{21} & Y_{22} \end{pmatrix}$; $N_r = \begin{pmatrix} N_{11} & 0 \\ N_{21} & N_{22} \end{pmatrix}$ and $M_r = \begin{pmatrix} M_{11} & 0 \\ M_{21} & M_{22} \end{pmatrix}$.

Substitute these coprime factors in the architectures obtained in Chapter 3 to obtain two ways of implementing triangular structured controller such that the performance problem is a convex problem in presence of sub-controller noise. Thus, the two sub-controllers with left-coprime architecture are given by

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & 0 & 0 \\ \hline I & 0 & 0 \\ \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & 0 & 0 \end{array}\right] \tag{4.6}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{22}^{-1}\widetilde{Y}_{22} & \widetilde{X}_{22}^{-1}\widetilde{Y}_{21} & -\widetilde{X}_{22}^{-1}\widetilde{X}_{21} \\ \hline I & 0 & 0 \\ \widetilde{X}_{22}^{-1}\widetilde{Y}_{22} & \widetilde{X}_{22}^{-1}\widetilde{Y}_{21} & -\widetilde{X}_{22}^{-1}\widetilde{X}_{21} \end{array}\right] \tag{4.7}$$

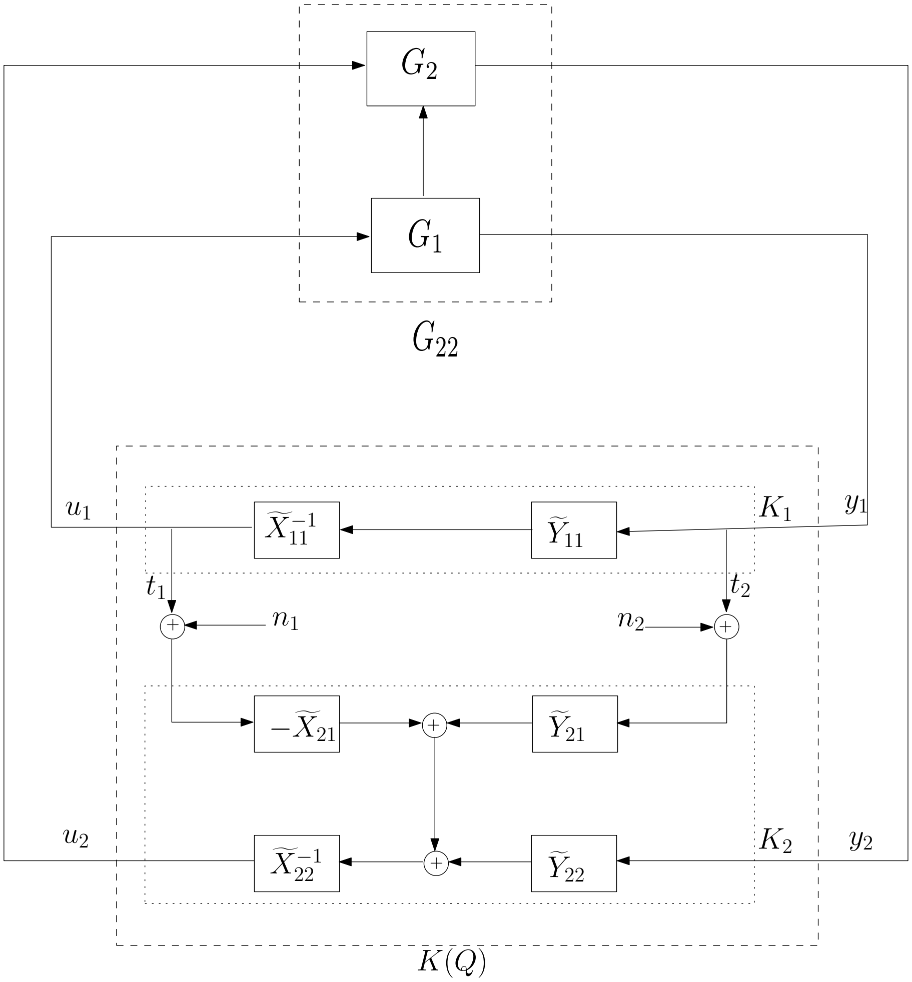


Figure 4.3 Left-coprime architecture for nested structure

and is equivalent to

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & 0 & 0 \\ \hline I & 0 & 0 \\ \widetilde{X}_{11}^{-1}\widetilde{Y}_{11} & 0 & 0 \end{array}\right] \tag{4.8}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} \widetilde{X}_{22}^{-1}\widetilde{Y}_{22} & \widetilde{X}_{22}^{-1}\widetilde{Y}_{21} & -\widetilde{X}_{22}^{-1}\widetilde{X}_{21} \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]. \tag{4.9}$$

The architecture is shown in Figure 4.3. The closed-loop transfer functions are all affine in $Q$ and given by:

$$\begin{array}{lll} \Phi_{u_2n_1} = -M_{22}\widetilde{X}_{21}, & \Phi_{u_2n_2} = M_{22}\widetilde{Y}_{21}, & \Phi_{y_2n_1} = -N_{22}\widetilde{X}_{21}, \\ \Phi_{y_2n_2} = N_{22}\widetilde{Y}_{21}, & \Phi_{t_1v_1} = Y_{11}\widetilde{N}_{11}, & \Phi_{t_2v_1} = X_{11}\widetilde{N}_{11}, \\ \Phi_{t_1v_3} = Y_{11}\widetilde{M}_{11}, & \Phi_{t_2v_3} = X_{11}\widetilde{M}_{11} & . \end{array}$$

Another architecture is obtained by substituting coprime factors in right-coprime architecture. Thus, the two sub-controllers of right-coprime architecture for banded structure controller are given by

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{11}X_{11}^{-1} & I & Y_{11}X_{11}^{-1} \\ \hline Y_{21}X_{11}^{-1} & 0 & Y_{21}X_{11}^{-1} \\ -X_{21}X_{11}^{-1} & 0 & -X_{21}X_{11}^{-1} \end{array}\right] \tag{4.10}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{22}X_{22}^{-1} & I & Y_{22}X_{22}^{-1} \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right] \tag{4.11}$$

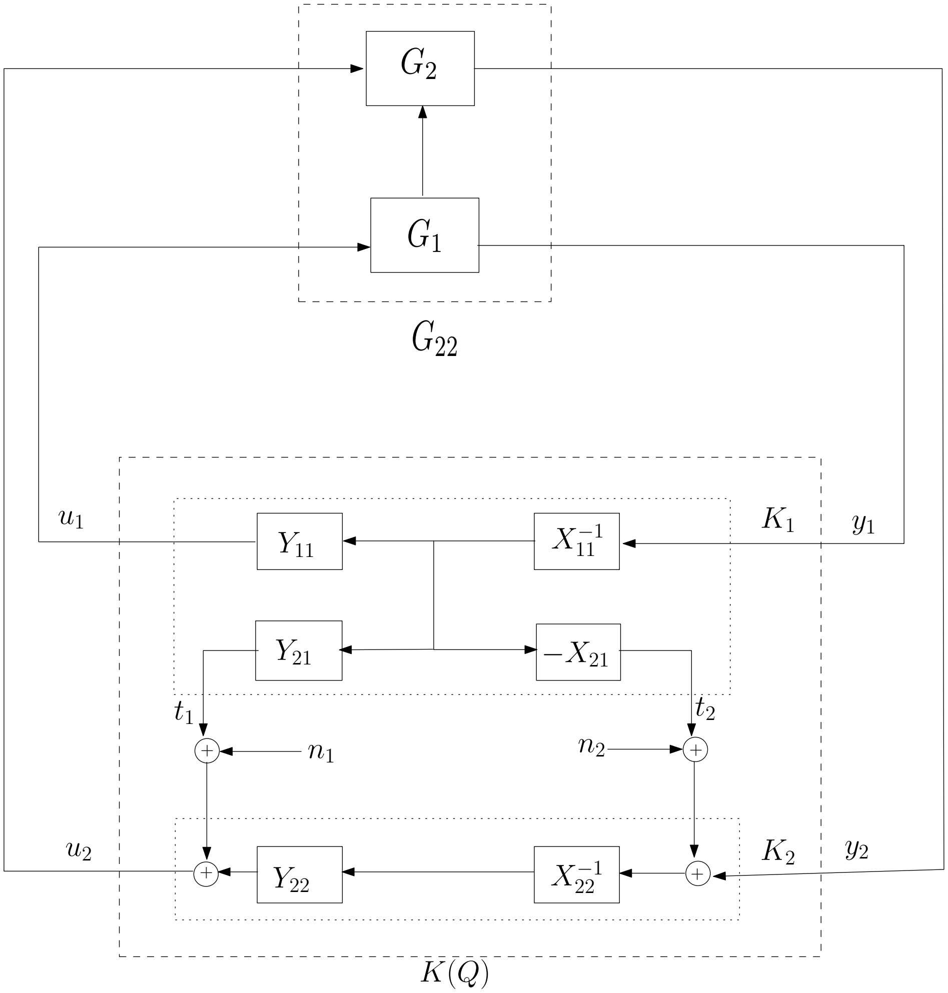


Figure 4.4 Right-coprime architecture for nested structure

and is equivalent to

$$K_1 = \left[\begin{array}{c|c} C^1_{11} & C^1_{12} \\ \hline C^1_{21} & C^1_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{11}X_{11}^{-1} & 0 & 0 \\ \hline Y_{21}X_{11}^{-1} & 0 & 0 \\ -X_{21}X_{11}^{-1} & 0 & 0 \end{array}\right] \tag{4.12}$$

$$K_2 = \left[\begin{array}{c|c} C^2_{11} & C^2_{12} \\ \hline C^2_{21} & C^2_{22} \end{array}\right] = \left[\begin{array}{c|cc} Y_{22}X_{22}^{-1} & I & Y_{22}X_{22}^{-1} \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]. \tag{4.13}$$

The architecture is shown in Figure 4.4. The closed-loop transfer functions are all affine in $Q$ and given by:

$$\begin{array}{lll} \Phi_{u_2n_1} = M_{22}\widetilde{X}_{22}, & \Phi_{u_2n_2} = M_{22}\widetilde{Y}_{22}, & \Phi_{y_2n_1} = -N_{22}\widetilde{X}_{22}, \\ \Phi_{y_2n_2} = N_{22}\widetilde{Y}_{22}, & \Phi_{t_1v_1} = Y_{21}\widetilde{N}_{11}, & \Phi_{t_2v_1} = -X_{21}\widetilde{N}_{11}, \\ \Phi_{t_1v_3} = Y_{21}\widetilde{M}_{11}, & \Phi_{t_2v_3} = -X_{21}\widetilde{M}_{11} & . \end{array}$$

Further, it can be shown that the result presented in (15) to obtain above two architectures for nested system can be derived from (3.34)-(3.35). This is done by by setting $C^1_{11} = \widetilde{X}_{11}^{-1}\widetilde{Y}_{11}$ and $C^2_{11} = \widetilde{X}_{22}^{-1}\widetilde{Y}_{22}$ in 3.34 and by setting $C^1_{11} = Y_{11}X_{11}^{-1}$ and $C^2_{11} = Y_{22}X_{22}^{-1}$, (3.35). As noted in the above two architectures for nested structure controllers, $C^1_{12} = 0, C^1_{22} = 0, C^2_{21} = 0$ and $C^2_{22} = 0$. This is in confirmation with the definition of $C^1_{12}, C^1_{22}, C^2_{21}$ and $C^2_{22}$ for nested structure as there is no signal being transmitted from $K_2$ to $K_1$. Using this and right multiplying (3.34) by $\overline{X}_r$ and left multiplying (3.35) by $\overline{X}_l$, following two equations are obtained:

$$T_aT_b = \begin{bmatrix} 0 \\ \overline{T}_a \end{bmatrix} \begin{bmatrix} \overline{T}_b & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ (\widetilde{Y}_{21}X_{11} - \widetilde{X}_{21}Y_{11}) & 0 \end{bmatrix}$$

$$T_aT_b = \begin{bmatrix} 0 \\ \overline{T}_a \end{bmatrix} \begin{bmatrix} \overline{T}_b & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ (\widetilde{X}_{22}Y_{21} - \widetilde{Y}_{22}X_{21}) & 0 \end{bmatrix}$$

$$\Rightarrow \overline{T}_a\overline{T}_b = \widetilde{Y}_{21}X_{11} - \widetilde{X}_{21}Y_{11}, \text{ and} \tag{4.14}$$

$$\overline{T}_a\overline{T}_b = \widetilde{X}_{22}Y_{21} - \widetilde{Y}_{22}X_{21}. \tag{4.15}$$

where $\overline{T}_a = \widetilde{X}_{22}C^2_{12}$ and $\overline{T}_b = C^1_{21}X_{11}$. Thus, (4.14)-(4.15) are same as the set of equations used in (15) to derive the architectures shown in Figure 4.3 and 4.4. Note that if (4.14) (and (4.15)) can be factorized to obtain affine in $Q$ factors $\overline{T}_a$ and $\overline{T}_b$, that will result in all possible architectures with $C^1_{11} = \widetilde{Y}_{11}\widetilde{X}^{-1}_{11}$ and $C^2_{11} = \widetilde{Y}_{22}\widetilde{X}^{-1}_{22}$ (and architecture with $C^1_{11} = Y_{11}X^{-1}_{11}$ and $C^2_{11} = Y_{22}X^{-1}_{22}$) for which the performance problem in presence of sub-controller communication is a convex problem.

**Theorem 4.2.1** *Consider 2-nest $G_{22} - K$ system shown in Figure 2.5 having double-coprime factorization given by Lemma 4.2.1 with nested-structure $K$ being a stabilizing controller parameterized in terms of nested-structure $Q$. $K$ is implemented distributively such that $C^1_{11} = \widetilde{Y}_{11}\widetilde{X}^{-1}_{11}$ and $C^2_{11} = \widetilde{Y}_{22}\widetilde{X}^{-1}_{22}$. Then there exists an architecture for distributive implementation such that the closed-loop map $T(K)$ is a stable and affine in $Q$ if and only if there exists a factorization of $(\widetilde{Y}_{21}X_{11} - \widetilde{X}_{21}Y_{11})$ such that the two factors are stable and affine in $Q$.*

**Corollary 4.2.1** *Consider 2-nest $G_{22} - K$ system shown in Figure 2.5 having double-coprime factorization given by Lemma 4.2.1 with nested-structure $K$ being a stabilizing controller parameterized in terms of nested-structure $Q$. $K$ is implemented distributively such that $C^1_{11} = Y_{11}X^{-1}_{11}$ and $C^2_{11} = Y_{22}X^{-1}_{22}$. Then there exists an architecture for distributive implementation such that the closed-loop map $T(K)$ is a stable and affine in $Q$ if and only if there exists a factorization of $(\widetilde{Y}_{21}X_{11} - \widetilde{X}_{21}Y_{11})$ such that the two factors are stable and affine in $Q$.*

# CHAPTER 5. ROBUST STABILITY FRAMEWORK

In Chapter 3 and 4, few architectures have been developed for distributed implementation such that the performance problem taking care of affect sub-controller noise and sub-controller communication power constraint can be formulated as a convex problem. In this Chapter, under those architectures the same performance problem for designing distributed controllers is addressed within the framework of robust controller synthesis and a method is developed to obtain optimal controller. This can be done by modelling the communication channels affected by noise as multiplicative channel uncertainty and cast the problem of finding the optimal controller which minimizes the affect of noise on the overall performance of the system as a robust controller synthesis problem. Similarly, the affect of external signals on the interconnection is also taken care by modelling the links between plant and controller as multiplicative channel uncertainty. For the illustration purpose the discussion is restricted to nested structure systems with uncertainty only in sub-controller communication but this formulation can be generalized to banded structure as well as plants with no information structure with uncertainty in all possible external links.

Towards this, consider the descriptions given in Figure 5.1 that shows uncertainty affecting the link from the sub-controller $K_1$ to $K_2$. Figure 5.1 implements $K_1$ and $K_2$ with two-channel transmission from $K_1$ to $K_2$ based on one of the architectures obtained in previous Chapters. The above uncertainty characterizations can be cast into the standard $M - \Delta$ framework as shown in Figure 5.2, where $n = \binom{n_1}{n_2}$, $s = \binom{s_1}{s_2}$,

$$\Delta_n = \begin{pmatrix} \Delta_{n_1} & 0 \\ 0 & \Delta_{n_2} \end{pmatrix}$$

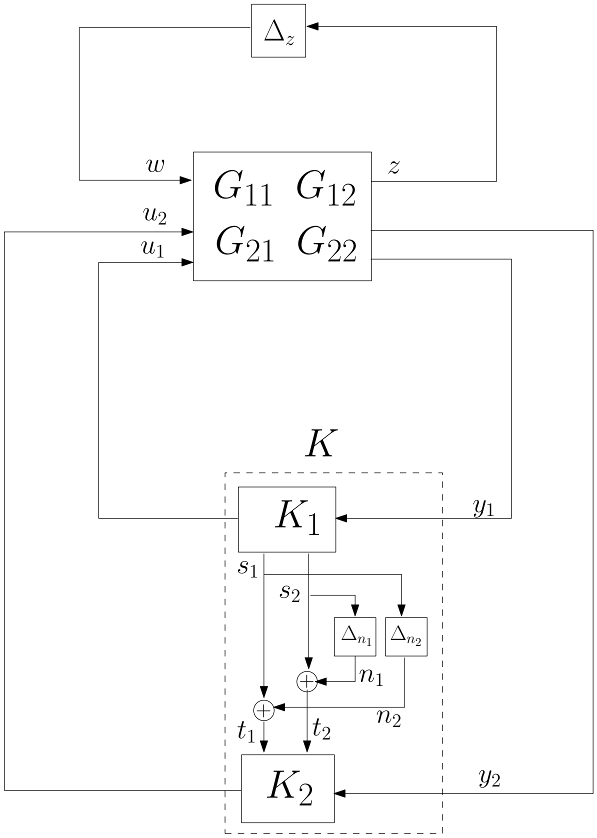


Figure 5.1 $G - K$ nested system with noise modelled as multiplicative uncertainty

and

$$M(Q) : \begin{pmatrix} w \\ n \end{pmatrix} \mapsto \begin{pmatrix} z \\ s \end{pmatrix}$$

$$= \left[ \begin{array}{l|ll} H - U * Q * V & H'_{a1} - U'_{a1} * Q * V'_{a1} & H'_{a2} - U'_{a2} * Q * V'_{a2} \\ \hline H'_{b1} - U'_{b1} * Q * V'_{b1} & 0 & \\ H'_{b2} - U'_{b2} * Q * V'_{b2} & & \end{array} \right]$$

where $Q$ is sable lower triangular transfer function, and

Following class for uncertainty description is considered:

$$\Delta_{LTV} \quad = \quad \{\Delta \in \mathcal{S} \text{ is linear time varying and } \|\Delta\| < \infty\}$$

where $\mathcal{S}$ characterizes the structure and the norm is either $\ell_\infty$ or $\ell_2$ induced norm.

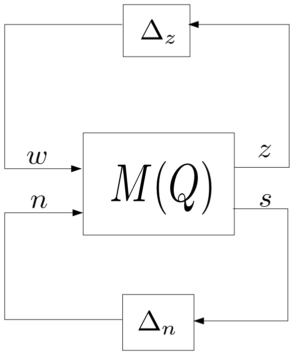


Figure 5.2 The M-$\Delta$ configuration for two channel case shown in Figure 5.1

Note that when $\mathcal{S}$ is given by the block diagonal structure $diag(\Delta_z, \Delta_n)$ with $\Delta_z$, and $\Delta_n$ being unstructured then the $M - \Delta$ interconnection is robustly stable with respect to all $\Delta \in B\Delta_{LTV} := \{\Delta \in \Delta_{LTV} | \|\Delta\| \leq 1\}$ if and only if $\inf_{D \in \mathcal{D}} \|DM(Q)D^{-1}\|_{\ell_1} < 1$ where $\mathcal{D} = \{D = diag(1, d_1, d_2) \text{ with } d_i > 0\}$ (19).

Thus the problem for robust synthesis in this case reduces to the problem

$$\inf_{Q \in \ell_1} \inf_{D \in \mathcal{D}} \|DM_2(Q)D^{-1}\|_{\ell_1}.$$

This problem is nonconvex in the variables $D$ and the Youla parameter $Q$. Recently in (20) a global solution to the above synthesis problem was achieved. This provides an effective procedure to address the problem of synthesizing controllers for $\ell_1$ robust synthesis when there is uncertainty in the nest to nest, sub-controller to sub-controller uncertainty when the uncertainty is described in the $\ell_\infty$ sense.

# CHAPTER 6. EXAMPLES

## 6.1 Optimal Distributed Controller Design for 2-node ABR Network: Robust Synthesis Framework

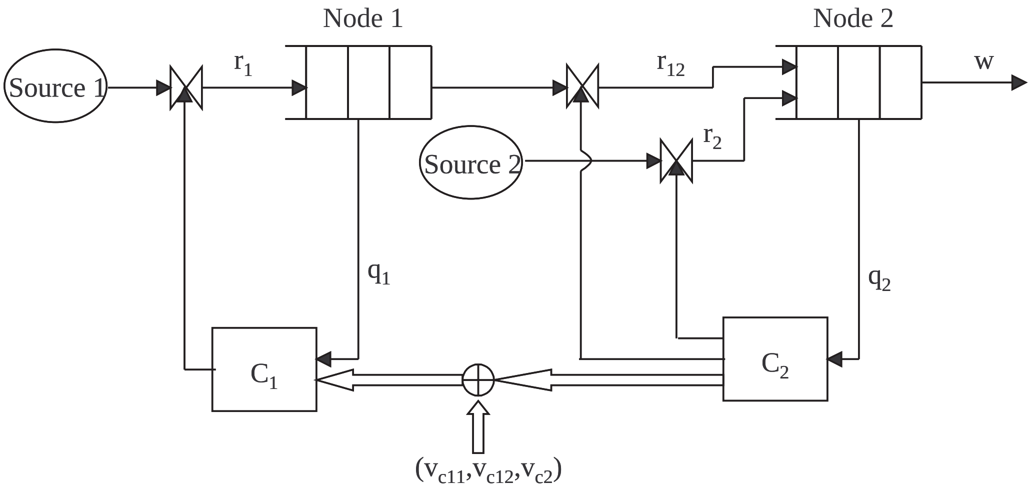


Figure 6.1 2-nodal ABR network with congestion control

Consider the nested system shown in Figure 6.1 of 2-node Available Bit Rate (ABR) communication network, with the problem of congestion of data packets at two nodes. This example illustrates design of an optimal controller with nested structure using robust control synthesis technique as discussed in Chapter 5. The objective is to not only avoid the congestion while keeping the channel utilization ratio as large as possible, but also to minimize the affect of sub-controller to sub-controller noise.

In Figure 6.1, $r_1$ and $r_2$ are the rate with which source 1 and 2 transmit data packets to node 1 and 2, respectively. $r_{12}$ is the rate of flow from node 1 to node 2. $w$ represents the total available capacity (bit-rate) for the two sources. $q_1$ and $q_2$ denote the queue lengths at node 1 and 2, respectively. The network control the network traffic by regulating rates $r_1, r_{12}$ and $r_2$. The overall controller consists of two sub-controllers $C_1$ and $C_2$, controlling $r_1$ and $(r_{12}', r_2')'$, respectively. Note that, the flow of information between controller is only from $C_2$ to $C_1$, similar to the 2-nest system of Figure 2.7, with $K_2 = C_1$ and $K_1 = C_2$. The objectives are to avoid the two queues from overflowing by avoiding congestion, to maximize the utilization factor of the network, i.e. to make $r_1 + r_2$ match $w$ as close as possible, to minimize the affect of sub-controller to sub-controller noise on the queue lengths and rates of transmission, and to regulate the signal power transmitted between two sub-controllers.

The controller $K$ is implemented using the left-coprime architecture derived in Chapter 3 such that $K$ internally stabilizes the network in the presence of noise in the communication channel between $C_1$ and $C_2$ sub-controllers. The information transmitted from $C_2$ to $C_1$ are $(r_{12}', r_2')'$ and $q_2$, and they get corrupted by noise viz. $v_{c11}, v_{c12}$ and $v_{c2}$, respectively. The exogenous signals are identified as the available capacity $w$, and the noise $(v_{c11}', v_c12', v_{c2}')'$. The regulated variables are two queue lengths $q_1$ and $q_2$, and the difference between the data rate of the source and the fraction of $w$ allocated to that source, i.e. $r_1 - a_1 * w$ and $r_2 - a_2 * w$. Thus, $z = [\ (r_2 - a_1 * w)' \quad (r_1 - a_2 * w)' \quad q_2' \quad q_1'\ ]'$. The controlled inputs are the data rates $r_2, r_{12}$ and $r_1$. Thus, $u = [\ r_2' \quad r_{12}' \quad r_1'\ ]'$, with $u_1 = (r_2', r_{12}')'$ and $u_2 = r_1$. And, the measured outputs are the queue lengths $q_1$ and $q_2$, i.e. $y = [\ y_1' \quad y_2'\ ]' = [\ q_2' \quad q_1'\ ]'$. Let $a_1 = a_2 = 0.5$ such that each of the nodes gets half of the available capacity $w$. Further, it is assumed that $w$ is typically step signal to be tracked.

The dynamics of the network is given by:

- Node 1: $q_1(k+1) = q_1(k) + r_1(k) - r_{12}(k)$
- Sub-controller $C_1$: $r_1 = f_1(q_1, r_2 + v_{c11}, r_{12} + v_{c12}, q_2 + v_{c2})$
- Node 2: $q_2(k+1) = q_2(k) + r_2(k) + r_{12}(k) - w(k)$
- Sub-controller $C_2$: $r_2 = f_2(q_2); r_{12} = f_{12}(q_2)$

where $f_1$, $f_2$ and $f_{12}$ are causal and linear operators. Clearly, plant $G_{22}$ (the part of generalized plant $G : [t; u] \mapsto [z; y]$ which maps $u$ to $y$) and controller $K$ in this case are lower triangular operators as shown below:

$$G_{22} \quad := \quad \left[\begin{array}{cc|c} * & * & 0 \\ \hline * & * & * \end{array}\right]$$

$$K \quad := \quad \left[\begin{array}{c|c} * & 0 \\ * & 0 \\ \hline * & * \end{array}\right]$$

The state-space description of $G_{22}$ is given by:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|cc} A_1 & 0 & B_1 & 0 \\ 0 & A_2 & B_{21} & B_2 \\ \hline C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \end{array}\right]$$

where $A_1 = A_2 = 1$, $B_1 = \left[\begin{array}{cc} 1 & 1 \end{array}\right], B_{21} = \left[\begin{array}{cc} 0 & -1 \end{array}\right], B_2 = 1$, and $C_1 = C_2 = 1$.

The state-feedback matrix $F$, and observer gain matrix $L$ for $G_{22}$ are chosen such that $A + LC$ and $A + BF$ are Hurwitz:

$$F = \left[\begin{array}{cc} F_1 & 0 \\ 0 & F_2 \end{array}\right], L = \left[\begin{array}{cc} L_1 & 0 \\ 0 & L_2 \end{array}\right]$$

where $F_1 = \left[\begin{array}{cc} -0.9 & 0 \end{array}\right]^T, F_2 = -0.9, L_1 = L_2 = -0.9$.

Then, the right coprime factors of $G_{22}$ are given by:

$$X_r = \left[\begin{array}{c|c} \frac{1+0.8\lambda}{1-0.1\lambda} & 0 \\ \hline 0 & \frac{1+0.8\lambda}{1-0.1\lambda} \end{array}\right]; Y_r = \left[\begin{array}{c|c} \frac{-0.81\lambda}{1-0.1\lambda} & 0 \\ 0 & 0 \\ \hline 0 & \frac{-0.81\lambda}{1-0.1\lambda} \end{array}\right];$$

$$X_l = \left[\begin{array}{cc|c} \frac{-1-0.8\lambda}{1-0.1\lambda} & \frac{-0.9\lambda}{1-0.1\lambda} & 0 \\ 0 & -1 & 0 \\ \hline 0 & \frac{0.9\lambda}{1-0.1\lambda} & \frac{-1-0.8\lambda}{1-0.1\lambda} \end{array}\right]; Y_l = \left[\begin{array}{c|c} \frac{0.81\lambda}{1-0.1\lambda} & 0 \\ 0 & 0 \\ \hline 0 & \frac{0.81\lambda}{1-0.1\lambda} \end{array}\right];$$

$$M_r = \left[\begin{array}{cc|c} \frac{1-\lambda}{1-0.1\lambda} & \frac{0.9\lambda}{1-0.1\lambda} & 0 \\ 0 & -1 & 0 \\ \hline 0 & \frac{-0.9\lambda}{1-0.1\lambda} & \frac{1-\lambda}{1-0.1\lambda} \end{array}\right]; N_r = \left[\begin{array}{cc|c} \frac{-\lambda}{1-0.1\lambda} & \frac{-\lambda}{1-0.1\lambda} & 0 \\ \hline 0 & \frac{\lambda}{1-0.1\lambda} & \frac{-\lambda}{1-0.1\lambda} \end{array}\right].$$

$$M_l = \left[\begin{array}{c|c} \frac{1-\lambda}{1-0.1\lambda} & 0 \\ \hline 0 & \frac{1-\lambda}{1-0.1\lambda} \end{array}\right]; N_l = \left[\begin{array}{cc|c} \frac{\lambda}{1-0.1\lambda} & \frac{\lambda}{1-0.1\lambda} & 0 \\ \hline 0 & \frac{-\lambda}{1-0.1\lambda} & \frac{-\lambda}{1-0.1\lambda} \end{array}\right].$$

Thus, all right coprime factors are lower triangular, and noting that $K = (Y_r - M_rQ)(X_r - N_rQ)^{-1}$ and $Q = (KN_r - Mr)^{-1}(KX_r - Y_r)$, implies that $Q$ is lower triangular iff $K$ is lower triangular.

This synthesis problem is converted into robust stability problem by formulating noise affected channel as uncertain communication channel (multiplicative uncertainty) as shown in Figure 6.2, which can be written in standard $M - \triangle$ form. Thus, the problem of stabilization and performance of the 2-node ABR system in the presence of sub-controller to sub-controller noise becomes a robust stability problem. To further convert it into robust performance problem, add an uncertainty between the regulated variable $z$ and exogenous input signal $w$ (as done in Figure 6.3). Thus, the problem can be written in standard $M - \triangle$ form, where

$$M(K) = \left[\begin{array}{c|c} \Phi_{zw} & \Phi_{zv} \\ \hline \Phi_{rw} & \Phi_{rv} \end{array}\right].$$

$\Phi_{zw}$ is the $4 \times 1$ closed-loop transfer matrix from $w$ to $z$, $\Phi_{zv}$ is the $4 \times 3$ closed-loop transfer matrix from $v = (v'_{c11}\ v'_{c12}\ v'_{c2})'$ to $z$, $\Phi_{rw}$ is the $3 \times 1$ closed-loop transfer function from $w$ to $r = (u'_1\ y'_1)'$ and $\Phi_{rv}$ is the $3 \times 3$ closed-loop transfer function from $v$ to $r$. $M(K)$ can be written as affine function of Youla parameter $Q$, for $\Phi_{zw}$ and $\Phi_{rw}$ can be written as affine functions of $Q$ using coprime factors of $G_{22}$ and $\Phi_{zv}$ is affine function of $Q$ as established in Chapter 3. $\Phi_{rv}$ is equal 0.

These four transfer functions can be written in terms of Youla parameter $Q$ ( in $H - U * Q * V$

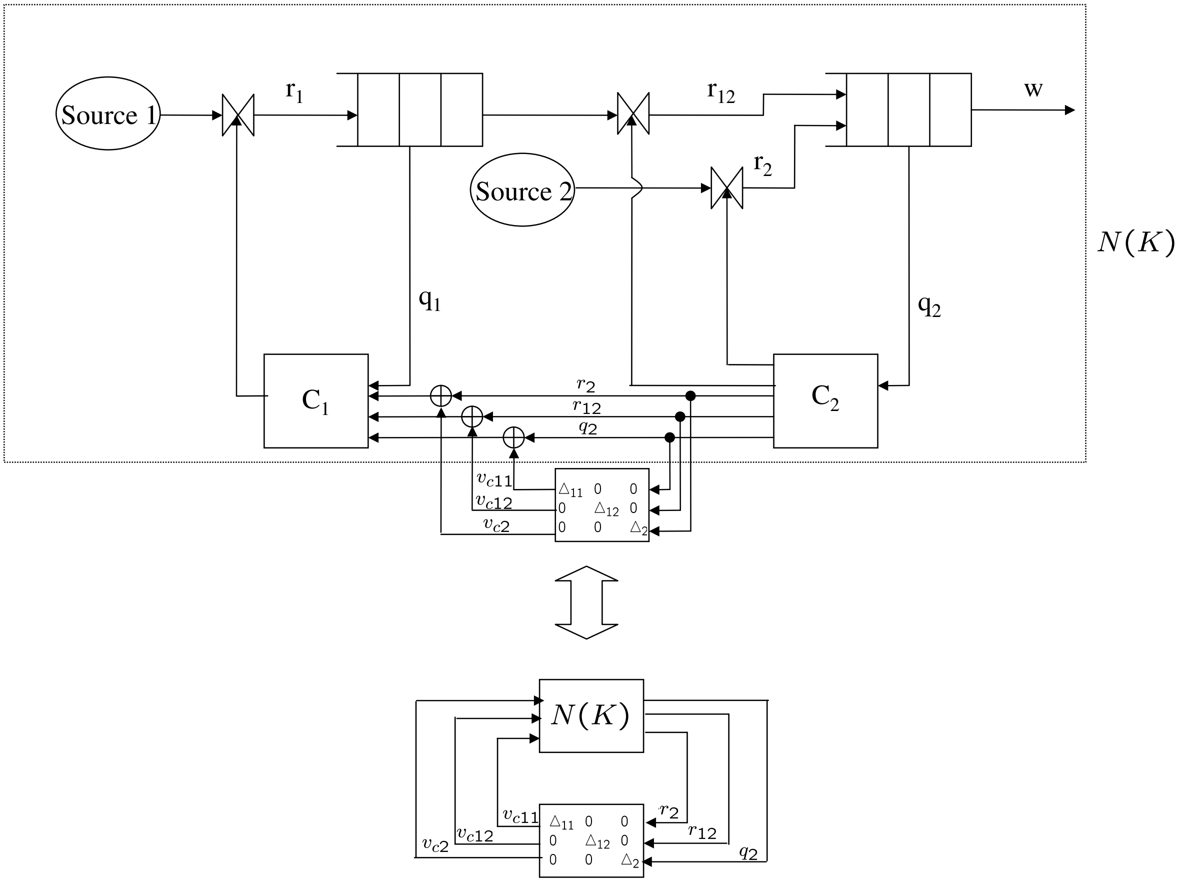


Figure 6.2 2-nodal ABR network with uncertain communication channel

form ) such that the design problem can be written as following optimization problem:

$$\nu = \inf_{Q\in\ell_1,\ lower\ triangular} \inf_{D\in\mathcal{D}} \|DM_2(Q)D^{-1}\|_{\ell_1} \tag{6.1}$$

which can be solved using the technique given in (20) to obtain a global solution.

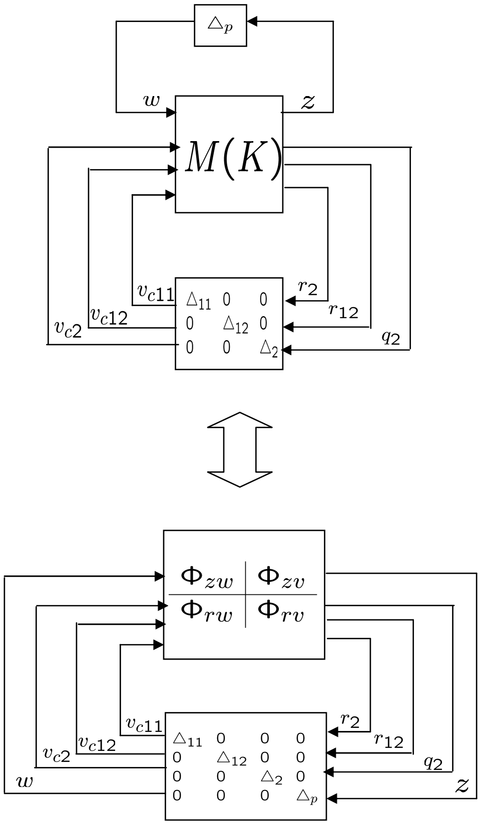


Figure 6.3 $M - \triangle$ form of 2-nodal ABR network

Writing dynamics equations of network in terms of $\lambda$ transform:

$$\begin{pmatrix} z \\ y \end{pmatrix} = \left(\begin{array}{c|ccc} -a_1 & 1 & 0 & 0 \\ -a_2 & 0 & 0 & 1 \\ \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & 0 \\ 0 & 0 & \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} \\ \hline \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & 0 \\ 0 & 0 & \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} \end{array}\right) \begin{pmatrix} w \\ u \end{pmatrix}$$

$$\begin{aligned}
&= \left(\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right)\begin{pmatrix} w \\ u \end{pmatrix} \equiv G\begin{pmatrix} w \\ u \end{pmatrix} \\
&u = Ky \\
&\Rightarrow \Phi_{zw} = \mathrm{LFT}(G, K) = H_1 - U_1 * Q * V_1 \qquad (6.2)
\end{aligned}$$

where, $H_1 = G_{11} + G_{12}Y_rM_lG_{21}$, $U_1 = G_{12}M_r$ and $V_1 = M_lG_{21}$. Using values of coprime factors, $H_1, U_1$ and $V_1$ can be obtained as:

$$H_1 = \begin{bmatrix} -0.5 + \dfrac{0.81\lambda^2}{(1-0.1\lambda)^2} \\ -0.5 \\ \dfrac{\lambda(1-0.8\lambda)}{(1-0.1\lambda)^2} \\ 0 \end{bmatrix}, U_1 = \begin{bmatrix} \dfrac{-(1-\lambda)}{1-0.1\lambda} & \dfrac{0.9\lambda}{1-0.1\lambda} & 0 \\ 0 & \dfrac{-0.9\lambda}{1-0.1\lambda} & \dfrac{-(1-\lambda)}{1-0.1\lambda} \\ \dfrac{-\lambda}{1-0.1\lambda} & \dfrac{-\lambda}{1-0.1\lambda} & 0 \\ 0 & \dfrac{\lambda}{1-0.1\lambda} & \dfrac{-\lambda}{1-0.1\lambda} \end{bmatrix}, V_1 = \begin{bmatrix} \dfrac{-\lambda}{1-0.1\lambda} \\ 0 \end{bmatrix}.$$

Since, controller is implemented using left-coprime architecture of Chapter 3, the transfer function $\Phi_{zv}$ can be written as

$$\Phi_{zv} = \begin{bmatrix} 0 & 0 \\ -M_{22}\widetilde{X}_{21} & M_{22}\widetilde{Y}_{21} \\ 0 & 0 \\ -N_{22}\widetilde{X}_{21} & N_{22}\widetilde{Y}_{21} \end{bmatrix}.$$

Since, coprime factors $X_l$ and $Y_l$ are affine in $Q$, rewrite $\Phi_{zv} = H_2 - U_2 * Q * V_2$ where

$$H_2 = \begin{bmatrix} 0 & 0 \\ -M_{22}\widetilde{X}^0_{21} & M_{22}\widetilde{Y}^0_{21} \\ 0 & 0 \\ -N_{22}\widetilde{X}^0_{21} & N_{22}\widetilde{Y}^0_{21} \end{bmatrix}, U_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & M_{22} \\ 0 & 0 & 0 \\ 0 & 0 & N_{22} \end{bmatrix} \text{ and } V_2 = \begin{bmatrix} -\widetilde{N}_{11} & \widetilde{M}_{11} \\ -\widetilde{N}_{21} & \widetilde{M}_{21} \end{bmatrix}$$

where $\widetilde{X}^0_{21}$ and $\widetilde{Y}^0_{21}$ are lower off-diagonal parts of $X_l$ and $Y_l$. By substituting for values of coprime factors, $H_2, U_2$ and $V_2$ can be written as:

$$H_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{0.9\lambda(1-\lambda)}{(1-0.1\lambda)^2} & 0 \\ 0 & 0 & 0 \\ 0 & \dfrac{0.9\lambda^2}{(1-0.1\lambda)^2} & \end{bmatrix}, U_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \dfrac{-(1-\lambda)}{1-0.1\lambda} \\ 0 & 0 & 0 \\ 0 & 0 & \dfrac{-\lambda}{1-0.1\lambda} \end{bmatrix},$$

$$V_2 = \begin{bmatrix} -\dfrac{\lambda}{1-0.1\lambda} & -\dfrac{\lambda}{1-0.1\lambda} & \dfrac{1-\lambda}{1-0.1\lambda} \\ 0 & \dfrac{\lambda}{1-0.1\lambda} & 0 \end{bmatrix}$$

In order to obtain $Q$ parameterization of $\Phi_{rw}$, the dynamic equations of the network with $r = (r_2' \; r_{12}' \; q_2')'$ as regulated variable is written in terms of $\lambda$ as following:

$$\begin{aligned} \begin{pmatrix} r \\ y \end{pmatrix} &= \left( \begin{array}{c|ccc} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & 0 \\ \hline \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & 0 \\ 0 & 0 & \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} \end{array} \right) \begin{pmatrix} w \\ u \end{pmatrix} && (6.3) \\ &\equiv \left( \begin{array}{c|c} \bar{G}_{11} & \bar{G}_{12} \\ \hline \bar{G}_{21} & \bar{G}_{22} \end{array} \right) \begin{pmatrix} w \\ u \end{pmatrix} \equiv \bar{G} \begin{pmatrix} w \\ u \end{pmatrix} && (6.4) \\ u &= Ky && (6.5) \\ \Rightarrow \Phi_{zw} &= \mathrm{LFT}(\bar{G}, K) = H_3 - U_3 * Q * V_3 && (6.6) \end{aligned}$$

where, $H_3 = \bar{G}_{11} + \bar{G}_{12} Y_r M_l \bar{G}_{21}$, $U_3 = \bar{G}_{12} M_r$ and $V_3 = M_l \bar{G}_{21}$. By substituting for values of coprime factors, $H_3, U_3$ and $V_3$ can be written as:

$$H_3 = \begin{bmatrix} \dfrac{0.81\lambda^2}{(1-0.1\lambda)^2} \\ 0 \\ \dfrac{-\lambda(1+0.8\lambda)}{(1-0.1\lambda)^2} \end{bmatrix}, U_3 = \begin{bmatrix} \dfrac{-(1-\lambda)}{1-0.1\lambda} & \dfrac{0.9\lambda}{1-0.1\lambda} & 0 \\ 0 & -1 & 0 \\ \dfrac{-\lambda}{1-0.1\lambda} & \dfrac{-\lambda}{1-0.1\lambda} & 0 \end{bmatrix}, V_3 = \begin{bmatrix} \dfrac{-\lambda}{1-0.1\lambda} \\ 0 \end{bmatrix}.$$

Thus, $M$ transfer matrix can be written as an affine function of $Q$ as follows:

$$M(K)=\left[\begin{array}{c|c}\Phi_{zw} & \Phi_{zv}\\ \hline \Phi_{rw} & \Phi_{rv}\end{array}\right]=\left[\begin{array}{l|l}H_1-U_1*Q*V_1 & H_2-U_2*Q*V_2\\ \hline H_3-U_3*Q*V_3 & 0\end{array}\right]=M(Q).$$

Consider the following class for uncertainty description:

$$\Delta_{LTV} \quad = \quad \{\Delta \in \mathcal{S} \text{ is linear time varying and } \|\Delta\|_{i\infty} < \infty\},$$

where $\mathcal{S}$ characterizes the structure and subscript $i\infty$ stands for the $\ell_\infty$ and the $\ell_2$ induced norm.

The optimal controller obtained by solving the robust synthesis problem in Equation (6.1)is:

$$Q_{opt}(\lambda)=\left[\begin{array}{l|l}\frac{-0.6-0.283\lambda-0.0689\lambda^2+0.0217\lambda^3+0.016\lambda^4-0.002\lambda^5}{1+0.786\lambda+0.362\lambda^2+0.085\lambda^3+0.005\lambda^4} & 0\\ \frac{0.198\lambda-0.156\lambda^2-0.074\lambda^3-0.018\lambda^4-0.002\lambda^5}{1+0.786\lambda+0.362\lambda^2+0.085\lambda^3+0.005\lambda^4} & 0\\ \hline \frac{-0.001\lambda^3-0.001\lambda^4-0.001\lambda^5}{1+0.786\lambda+0.362\lambda^2+0.085\lambda^3+0.005\lambda^4} & \frac{0.9+0.706\lambda+0.325\lambda^2+0.077\lambda^3+0.004\lambda^4}{1+0.786\lambda+0.362\lambda^2+0.085\lambda^3+0.005\lambda^4}\end{array}\right]$$

The impulse response of $Q_{opt}$ is shown in Figure 6.4 , the lower triangular structure is clearly visible.

The impulse response of closed-loop system from noise to regulated variable $z$ with decentralized lower triangular controller $Q_{opt}$ is shown in Figure 6.5. As expected there is no affect on $q_2$ and $r_2$ while the controller stabilizes $q_1$ and also maintains $r_1$ close to $0.5w$.

## 6.2 Optimal Distributed Controller Design for 2-node ABR Network: Search over two different architectures

In this example, consider a 2-node ABR network as shown in Figure 6.1. It will be shown that the two architectures obtained in Chapter 3 will give different optimal controllers and different values of performance measure. The objective is same as the one in previous example i.e. to design a distributed controller for given ABR network which not only avoids the congestion in the network while keeping the channel utilization ratio as large as possible, but also minimizes the affect of the sub-controller to sub-controller noise on the queue lengths ($q_1$ and $q_2$) and the regulated rates ($r_1, r_{12}$ and $r_2$) of transmission of packets by regulating rates $r_1$ and $r_2$. It should also minimize the signal power of transmitted signal between sub-controllers.

Thus, it is required to find a stabilizing controller $K$ which is lower triangular and minimizes $\|T(K)\|$, where $T(K) : \begin{pmatrix} w \\ n \end{pmatrix} \mapsto \begin{pmatrix} z \\ t \end{pmatrix} = \begin{bmatrix} \Phi_{zw} & \Phi_{zn} \\ \Phi_{tw} & \Phi_{tn} \end{bmatrix}$, where $\Phi_{zw}$ captures the performance requirement of system, $\Phi_{zn}$ captures the affect of sub-controller communication noise on performance and $\Phi_{tw}$ denotes the power of transmitted signal with respect to power of external input signals where $n = v$ and $t = n + v$. From Chapter 3, $T(K) = T(Q)$, where $Q$ is lower triangular and stable. This parametrization makes the closed-loop map $\Phi_{zw}$ affine in $Q$, i.e. $\Phi_{zw} = H - U * Q * V$, where $H = G_{11} + G_{12}Y_r M_l G_{21}, U = G_{12}M$ and $V = M_l G_{21}$. $\Phi_{tn} = I$, but $\Phi_{zn}$ and $\Phi_{tw}$ depend on sub-controllers $C_{1b}$ and $C_{2a}$ which in turn depend on the architecture. Thus, the two performance optimization problems are:

$$I : \mu_l = \underbrace{\inf}_{\substack{K - \text{stabilizing, lower traingular,} \\ \text{based on the first architecture}}} \|T(K)\|_1$$

$$II : \mu_r = \underbrace{\inf}_{\substack{K - \text{stabilizing, lower traingular,} \\ \text{based on the second architecture}}} \|T(K)\|_1$$

Solving above two optimization problems for these two architectures, two different optimal controllers $Q_{opt,l}$ and $Q_{opt,r}$ are obtained as shown in Figure 6.6 - 6.7 with $\mu_l = 1.5$ and $\mu_r = 3.7$, respectively. Clearly, in this example it is better to use second architecture with $\mu_l = 1.5$.

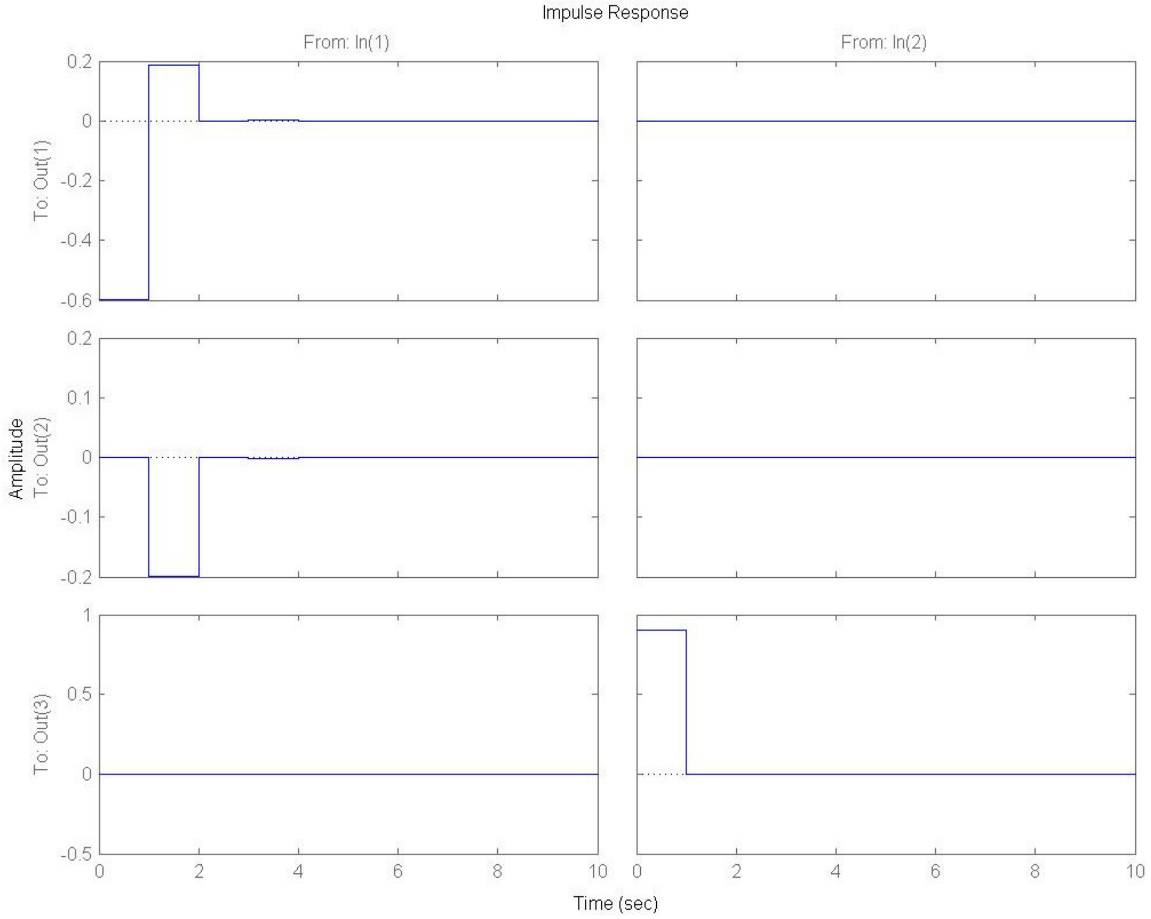


Figure 6.4 Impulse response of structured optimum controller $Q_{opt}$.

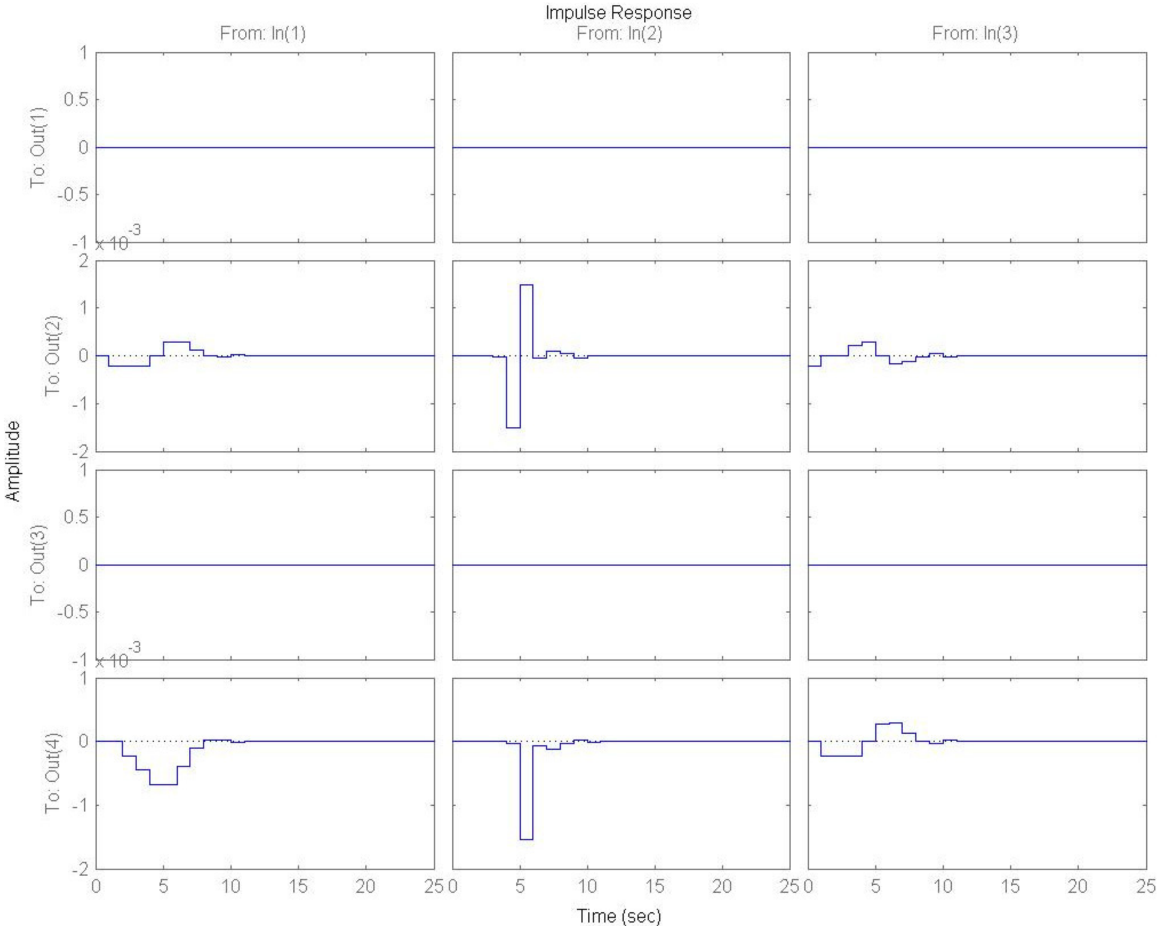


Figure 6.5 Impulse response of closed-loop system from noise to regulated variable $z$ with decentralized lower triangular controller $Q_{opt}$.

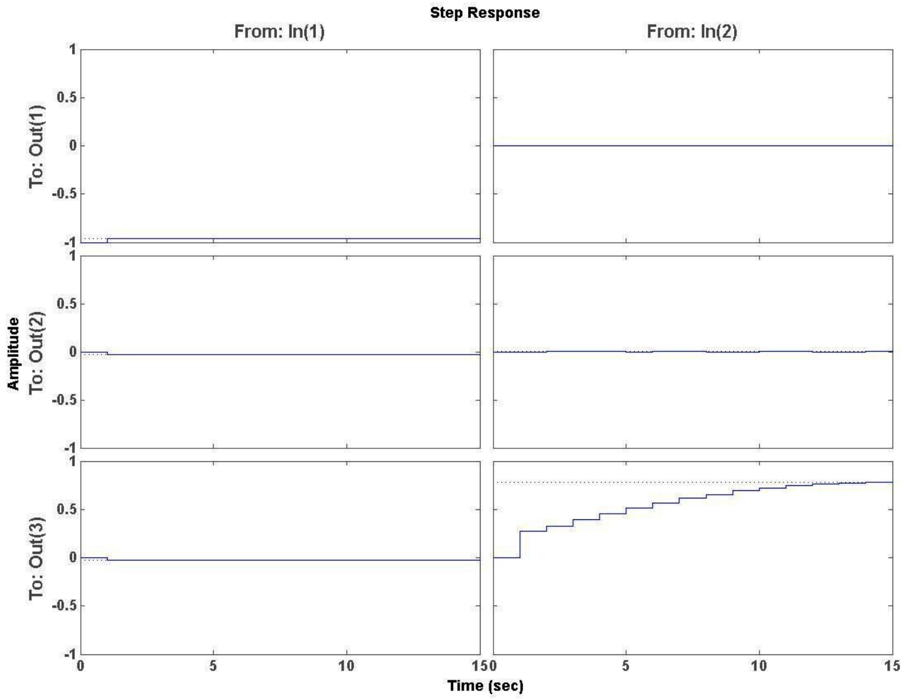


Figure 6.6 $Q_{opt,r}$ optimal controller for first architecture with $\mu_r = 3.74$

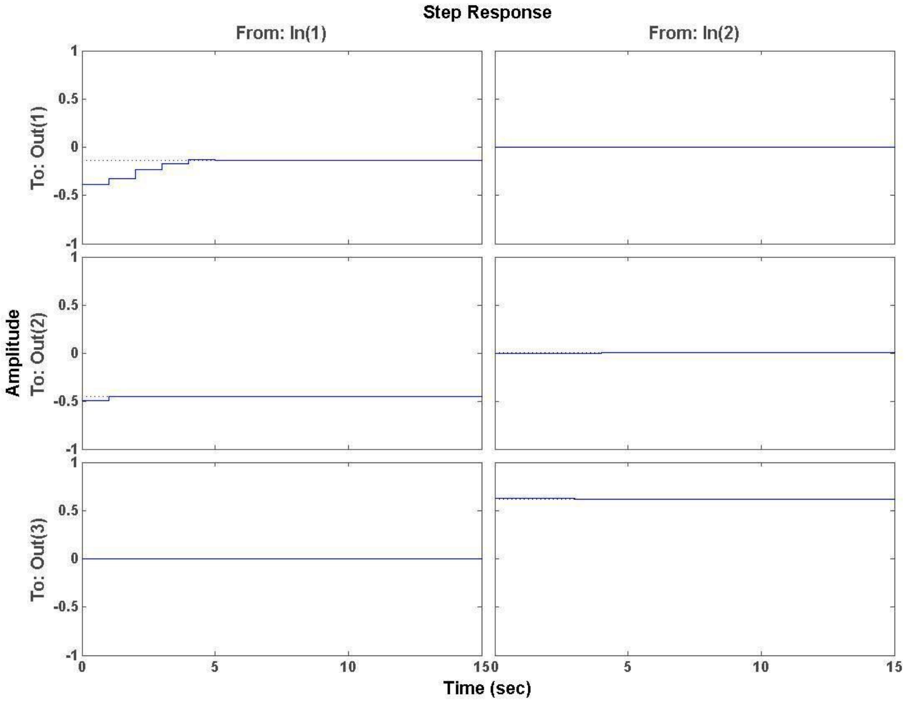


Figure 6.7 $Q_{opt,l}$ optimal controller for second architecture with $\mu_l = 1.5$

# PART II

# DISTRIBUTED DECISION MAKING

# CHAPTER 7. DISTRIBUTED AVERAGING PROBLEM

Consider a system of $N$ nodes or agents connected with each other in an arbitrary manner via communication links. Each node is sensing its local information e.g. local temperature or chemical concentration and is trying to compute the average of that local information over the whole network. The system is modelled as a graph $G := (V, E)$ consisting of a set $V := \{1, 2, ..., N\}$ of elements called vertices or nodes or agents, and a set $E$ of node pairs called edges, with $E \subseteq E_c := \{(i, j) | i, j \in V\}$. If $E = E_c$ i.e. each node is connected to rest of $n - 1$ nodes, it is called a complete graph. A graph is called undirected if for every pair of distinct nodes $i$ and $j$ both $(i, j)$ and $(j, i)$ are in $E$. Otherwise, it is called a directed graph or a digraph. A simple graph is a graph with no self loops, i.e. $(i, j) \notin E$ if $i = j$. A graph is connected if it has a path between each pair of distinct nodes $i$ and $j$, where by a path between nodes $i$ and $j$ we mean a sequence of distinct edges of $G$ of the form $(i, k_1), (k_1, k_2), \ldots, (k_m, j) \in E$. A digraph is called "strongly connected" if there is a directed path between each pair of distinct nodes. Diameter $D$ of a graph is the longest shortest path between any two pair of nodes. Fixed graphs are graphs in which the edge set $E$ does not change with time. In this paper, fixed graphs are considered.

Radius $r$ of node pair $(i, j)$ implies the minimum path length, i.e. the minimum number of edges connecting $i$ to $j$ is equal to $r$. The neighborhood $N_i$ of $i^{th}$ node is a set consisting of all nodes within radius 1 not including the $i^{th}$ node itself. The degree or out-degree of an $i^{th}$ node is $|N_i|$, where $|N_i|$ denotes the number of elements in $N_i$. The maximum degree of the graph is denoted by $\Delta$ and the minimum degree of the graph is denoted by $\delta$. The adjacency matrix $A = \{a_{ij}\}$ of a graph $G$ is an $N \times N$ matrix. $a_{i,j} > 0$ only if the node pair $(i, j) \in E$ and is equal to zero otherwise. The graph $G$ is assumed to be simple, which implies that $a_{i,i} = 0$

for all $i = 1, 2, \cdots N$. The diagonal matrix $\Phi$ is an $N \times N$ diagonal matrix with each diagonal entry $d_{ii} = \sum_{j=1}^{N} a_{ij}$. For undirected graph the graph Laplacian matrix $L$ is defined as $\Phi - A$.

The graph Laplacian matrix $L$ is an important function of the graph $G$. Eigenvalues of $L$ have direct relation to the connectivity of the graph. Let, $\lambda_1 \leq \lambda_2 \leq \cdots \lambda_N$ be $N$ eigenvalues of $L$. Since $L$ has row sum equal to zero (such matrices are called row stochastic), $\lambda_1 = 0$ is a trivial eigenvalue of $L$ with $\bar{1} := [1, \cdots, 1]^T$ as the corresponding eigenvector i.e. $L\bar{1} = 0$. A graph is connected if and only if the second smallest eigenvalue of Laplacian is non-zero i.e. $\lambda_2 > 0$ (26), and larger the $\lambda_2$ better is the connectivity of the graph and faster is the convergence of the distributed consensus protocol. The second smallest eigenvalue $\lambda_2$ is also called the algebraic connectivity of the graph. It is assumed that the communication among nodes is noiseless.

## 7.1 Average consensus protocol

The state vector of node-values for average consensus protocol is defined by column vector $x(k) = (x_1(k)\ x_2(k) \cdots x_N(k))^T$. The average consensus protocol denoted by $AP$ distributively computes the average of a given initial node-values $x(0) = (x_1(0)\ x_2(0) \cdots x_N(0))^T$. It takes $x(0)$ as an input and generates a sequence of node-values $x(k)x(k) = (x_1(k)\ x_2(k) \cdots x_N(k))^T$ such that $\{x(k)\}_{k=1}^{\infty} = AP(x(0))$ based on the following nearest-neighborhood update rule:

$$x_i(k+1) \quad = \quad x_i(k) + \epsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)) \quad \text{for all } i = 1, 2 \cdots N. \tag{7.1}$$

This implies

$$x(k+1) \quad = \quad Px(k) \tag{7.2}$$

where $P = I - \epsilon L$. Since $I - \epsilon L \approx exp(-\epsilon L)$, discrete time average consensus can be seen as the first order approximation of continuous time average consensus problem which is given by $\dot{x} = -\epsilon Lx$. It is known that $P$ with $0 < \epsilon < \frac{1}{d_{max}}$, where $d_{max} = \max d_{ii}$ satisfies following properties (32):

1. $P$ is row-stochastic non-negative matrix with a trivial eigenvalue of 1,

2. All eigenvalues of $P$ are inside the unit circle,

3. If $G$ is strongly connected then $P$ is a primitive matrix,

4. If $G$ is a balanced graph ($\bar{1}^T L = 0$) then $P$ is column-stochastic ($\bar{1}^T P = \bar{1}$). Note that every undirected graph is a balanced graph.

We will make following assumptions throughout the paper:

**Assumption 7.1.1** *(a) $0 < \epsilon < \frac{1}{d_{max}}$, (b) the graph $G$ is connected, and (c) if the graph $G$ is directed graph, then it is "strongly connected" and balanced.*

The average consensus protocol for the graph $G$ given by (7.1) converges asymptotically to average of the initial condition $x(0)$ denoted by $\alpha := \frac{1}{N}\sum_{i=1}^{N} x_i(0)$ (32). The average value $\alpha\bar{1}$ is an invariant quantity of the dynamics given by (7.1) i.e. $P(\alpha\bar{1}) = \alpha\bar{1}$. Further, this convergence is reached exponentially with exponent bounded above by $\mu_2$, which is the second largest eigenvalue of $P$ ($\mu_2 < 1$). Following property of $P$ which relies on the fact that $0 < \epsilon < \frac{1}{d_{max}}$ is needed the rest of the development.

**Proposition 7.1.1** *Let $p_{ij}$ be $(i,j)^{th}$ element of $P$. Then, $0 \leq p_{ij} < 1$ for all $i, j = 1, 2 \cdots N$. Moreover, $p_{ii} > 0$ for all $i$.*

**Proof** Since $P$ is a non-negative matrix, it implies that $p_{ij} \geq 0$. Since, $0 < \epsilon < \frac{1}{d_{max}}$, $0 < p_{ii} = 1 - \epsilon d_{ii} < 1$; and for $i \neq j$, $p_{ij} = \epsilon a_{ij} < \frac{a_{ij}}{d_{max}} \leq 1$. Thus, $p_{ij} < 1$ for all $i, j = 1, 2 \cdots N$ with $p_{ii} > 0$ for all $i$. Also, for all $j \in N_i$, $p_{ij} = \epsilon a_{ij} > 0$. ∎

The average protocol update rule can be rewritten as:

$$x_i(k+1) = \sum_{j=1}^{N} p_{ij} x_j(k) \tag{7.3}$$

at $i^{th}$ node. Thus, each updated node-value is a weighted average of its neighboring node-values such that weights are non-negative and strictly less than one with $\sum_{j=1}^{N} p_{ij} = 1$ for all $i$. This leads to the following conclusion:

**Proposition 7.1.2**

$$x_i(k+1) \leq \max_j x_j(k) \quad \textit{for all } i = 1, 2\cdots, N; \tag{7.4}$$

$$x_i(k+1) \geq \min_j x_j(k) \quad \textit{for all } i = 1, 2\cdots, N. \tag{7.5}$$

*Equalities hold in above equations if and only if* $x_i(k) = x_j(k)$ *for all* $i = 1, 2\cdots, N$.

**Proof** For any node $i$:

$$\begin{aligned} x_i(k+1) &= \sum_{j=1}^{N} p_{ij} x_j(k) \\ &\leq \sum_{j=1}^{N} p_{ij} \max_j x_j(k) = \max_j x_j(k) \sum_{j=1}^{N} p_{ij} = \max_j x_j(k). \end{aligned}$$

Similarly, it can be shown that

$$x_i(k+1) \geq \min_j x_j(k) \quad \text{for all } i = 1, 2\cdots, N.$$

Next, given that the equality holds in (7.4), suppose to the contrary that node-values are not same at time $k$ i.e. $x_i(k) \neq x_j(k)$ for some pair of nodes $(i, j)$. Then there exist a node $m$ with

$$x_m(k) < \max_l x_l(k).$$

Thus,

$$x_m(k+1) = \sum_{j=1}^{N} p_{mj} x_j(k) < \max_l x_l(k).$$

This contradicts the fact that equality holds in (7.4) for all $i = 1, 2, \cdots N$, implying that if equality holds in (7.4) then $x_i(k) = x_j(k)$ for all $i, j = 1, 2\cdots, N$. The other way is straight forward. Similarly, equality condition can be proved for (7.5). Thus, equalities hold in (7.4) and (7.5) if and only if $x_i(k) = x_j(k)$ for all $i = 1, 2\cdots, N$. $\blacksquare$

By taking maximum over all nodes in (7.4) (and minimum over all nodes in (7.5)) it can be shown that:

$$\max x(k+1) := \max_j x_j(k+1) \leq \max_j x_j(k) \tag{7.6}$$

$$\min x(k+1) := \min_j x_j(k+1) \geq \min_j x_j(k) \tag{7.7}$$

where equalities hold in both cases if and only if $x_i(k) = x_j(k)$ for all $i, j = 1, 2 \cdots, N$. Combining this with Proposition 7.1.2, it can be shown that node-value $x_i(k)$ at any time $k$ is bounded from above by the maximum value in network in the past and below by the minimum value in the network in the past. This can be expressed as follows:

$$\min x(k') \leq x_i(k) \leq \max x(k') \quad \text{for all } i = 1, 2 \cdots, N \quad \text{and} \quad \text{for all } k \geq k'. \tag{7.8}$$

The following lemma states that if a node reaches an average consensus protocol node-value that is strictly less than the maximum over the network at some past time instant $k'$ then the node-value at that node at any future time instant $k > k'$ remains strictly less than the maximum over the network at the past time instant $k'$.

**Lemma 7.1.1** *Consider a graph $G$ (undirected or directed “strongly connected”, balanced graph) running an average consensus protocol $AP$ given by (7.1) with an initial condition $x(k')$. Let, $i$ and $i'$ be nodes such that $x_i(k) < \max x(k')$ and $x_{i'}(k) > \min x(k')$, respectively for some time instant $k \geq k'$. Then for all $k'' \geq k$:*

$$x_i(k'') < \max x(k')$$
$$x_{i'}(k'') > \min x(k')$$

**Proof** It is given that for node $i$, $x_i(k) < \max x(k')$ for some time instant $k \geq k'$. It follows that:

$$\begin{aligned}
x_i(k+1) &= \sum_j p_{ij} x_j(k) = p_{ii} x_i(k) + \sum_{j \neq i} p_{ij} x_j(k) \\
&\leq p_{ii} x_i(k) + \sum_{j \neq i} p_{ij} \max x(k) \leq p_{ii} x_i(k) + \sum_{j \neq i} p_{ij} \max x(k') \text{ [From Prop. 7.1.2]} \\
&= p_{ii} x_i(k) - p_{ii} \max x(k') + \sum_j p_{ij} \max x(k') = p_{ii} x_i(k) + (1 - p_{ii}) \max x(k') \\
&< p_{ii} \max x(k') + (1 - p_{ii}) \max x(k') \ [\because p_{ii} > 0] \\
&= \max x(k').
\end{aligned}$$

Thus, $x_i(k+1) < \max x(k')$. It follows that $x_i(k+J) < \max x(k')$ for all $J \geq 1$. Therefore, if node $i$ assumes a node-value $x_i(k) < \max x(k')$, then it remains strictly less than $\max x(k')$

for all future time instances. Similar proof holds for the minimum value case. ■

Next lemma shows that after $D$ time steps the maximum value has to strictly decrease and the minimum value has to strictly increase.

**Lemma 7.1.2** *Consider a graph $G$ (undirected or directed "strongly connected", balanced graph) running an average consensus protocol $AP$ given by (7.1) with an initial condition $x(k')$ such that* $\max x(k') > \min x(k')$*. Then for all $k \geq k' + D$:*

$$\max x(k) < \max x(k'), \ \textit{and} \tag{7.9}$$

$$\min x(k) > \min x(k'). \tag{7.10}$$

**Proof** Consider any particular node $j$. There exists a node $i$ such that $x_i(k') < \max x(k')$ as $\min x(k') < \max x(k')$. The shortest distance between node $i$ and $j$, denoted by $d$, is less than or equal to the diameter $D$ of the graph. Let the path connecting $i$ and $j$ be $(i, m_1), (m_1, m_2), \ldots, (m_{d-1}, j)$. Because of weighted averaging, at time $k = k' + 1, x_{m_1}$ will become strictly less than $\max x(k')$ as shown below:

$$\begin{aligned}
x_{m_1}(k'+1) &= \sum_{n=1}^{N} p_{m_1 n} x_n(k') = p_{m_1 i} x_i(k') + \sum_{n \neq i} p_{m_1 n} x_n(k') \\
&\leq p_{m_1 i} x_i(k') + \sum_{n \neq i} p_{m_1 n} \max x(k') \\
&= p_{m_1 i} x_i(k') - p_{m_1 i} \max x(k') + \sum_{n=1}^{N} p_{m_1 n} \max x(k') \\
&= p_{m_1 i} x_i(k') + (1 - p_{m_1 i}) \max x(k') \\
&< p_{m_1 i} \max x(k') + (1 - p_{ii}) \max x(k') = \max x(k').
\end{aligned}$$

Thus, $x_{m_1}(k'+1) < \max x(k')$. Therefore, from Lemma 7.1.1 for all $k'' \geq k' + 1$, $x_{m_1}(k'') < \max x(k')$. It follows that for all $k'' \geq k' + 2$, $x_{m_2}(k'') < \max x(k')$ and that for all $k'' \geq k' + d_{ij} - 1$, $x_j(k'') < \max x(k')$. Note that $k' + D \geq k' + d_{ij} - 1$ $(D \geq d_{ij})$, therefore for all $k'' \geq k' + D$, $x_j(k'') < \max x(k')$. As $j$ is an arbitrary node, the above result given by (7.9) follows. (7.10) can be derived in a similar manner. ■

Thus from lemma 7.1.2, after a finite time given by the diameter $D$ of the graph, all node-values under averaging consensus protocol become strictly less than the maximum value in network in the past and strictly greater than the minimum value in the network in the past, which in turn means that after a finite time the maximum value in the network decreases and the minimum value in the network increases.

## 7.2 Maximum consensus protocol

The maximum consensus protocol denoted by $MXP$ distributively computes the maximum of a given initial node-values $z(0) = (z_1(0)z_2(0)\cdots z_N(0))^T$. It takes $z(0)$ as an input and generates a sequence of node-values $z(k)$ i.e. $\{z(k)\}_{k=1}^{\infty} = MXP(z(0))$ based on the following update rule:

$$z_i(k+1) = \max_{j \in N_i} z_j(k), \tag{7.11}$$

where $z_i(k)$ is the node-value of $i^{th}$ node for maximum consensus protocol. Each node updates its value to the present maximum value in its neighborhood. The overall state vector for maximum protocol is defined by the column vector $z(k) = (z_1(k)z_2(k)\cdots z_N(k))^T$. Note that $z_i(k)$ is a non-decreasing function with time $k$.

**Proposition 7.2.1** *Maximum consensus protocol $MXP$ given by (7.11) converges to* $\max z(0)$ *in finite time $T \leq D$.*

**Proof** Let $m$ be a node with node-value at $z_m(0) = \max z(0)$. Due to connectedness of graph $G$, each node in graph is connected to node $m$. Let, $\tilde{D}$ be the maximum distance between $m$ and any other node, then $\tilde{D} \leq D$. At time $k = 1$ all nodes connected to $m$ at one unit distance (one hop) will have the maximum value, at time $k = 2$ all nodes connected to $m$ at two unit distance (two hops) will have the maximum value, and so on. Thus, by time $T = \tilde{D}$ all the nodes will have maximum value. ∎

## 7.3 Minimum consensus protocol

The minimum consensus protocol denoted by $MNP$ distributively computes the minimum of a given initial node-values $y(0) = (y_1(0)y_2(0)\cdots y_N(0))^T$. It takes $y(0)$ as an input and generates a sequence of node-values $y(k)$ i.e. $\{y(k)\}_{k=1}^{\infty} = MXP(y(0))$ based on the following update rule:

$$y_i(k+1) = \min_{j \in N_i} y_j(k), \tag{7.12}$$

where $y_i(k)$ is the node-value of $i^{th}$ for minimum consensus protocol. Each node updates its value to the present minimum value in its neighborhood. The overall state vector for minimum protocol is defined by the column vector $y(k) = (y_1(k)y_2(k)\cdots y_N(k))^T$. Further, $y_i(k)$ is a non-increasing function with time $k$.

**Proposition 7.3.1** *Minimum consensus protocol given by (7.12) converges to* $\max y(0)$ *in finite time* $T \leq D$.

**Proof** Similar to the proof of Proposition 7.2.1. ∎

# CHAPTER 8. FINITE TIME CONVERGENCE WITHIN A GIVEN ERROR MARGIN

Consider the graph $G = (V, E)$ with $N$ nodes as defined above, each node running a distributed average consensus protocol $AP$ given by (7.1). In this Chapter, a distributed algorithm is provided which enables each node to detect the occurrence of the convergence in the network within a given error margin in finite time. To achieve this each node runs two more protocols, a maximum consensus protocol $MXP$ and a minimum consensus protocol $MNP$ given by (7.11) and (7.12) , respectively with $z(k_0) = y(k_0) = x(k_0)$, where $k_0$ is the time when maximum and minimum protocols are started. By finite time convergence it is implied that for any given $\rho > 0$, all agents can simultaneously reach to a decision in some finite time $T_c$ that their node-values are $\rho$ close to the desired average value i.e. they are in the interval $[\alpha - \rho, \alpha + \rho]$. From Proposition 7.1.2 and 7.2.1, after time $k = k_0 + D, z(k) = \max x(k_0)\bar{1}$ and $y(k) = \min x(k_0)\bar{1}$. Thus, at $k = k_0 + D$ the difference $z_i(k) - y_i(k)$ will be same at each node.

Define $T(j) = (j-1)D$ for $j = 1, 2, \cdots$, as the set of time instants when $MXP$ and $MNP$ are reset. This is done at $k = T(j)$ by setting their initial conditions $z(T(j))$ and $y(T(j))$ equal to the current node-values $x(T(j))$ from $AP$. Thus, at every time instant $k = T(j+1)$, $MXP$ running at each node with initial value $z(T(j))$ will output $\max z(T(j))$ and $MNP$ running at each node with initial value $y(T(j))$ will output $\min y(T(j))$. Define these outputs of $MXP$ and $MNP$ as $\bar{\alpha}(j) = \max z(T(j))$, $\underline{\alpha}(j) = \min y(T(j))$, respectively and the difference between these two outputs as $\beta(j) = \bar{\alpha}(j) - \underline{\alpha}(j)$. At $k = T(j+1)$ each node will have the same value $\beta(j)$. Following corollary shows that $\bar{\alpha}(j)$ and $\underline{\alpha}(j)$ both converge to $\alpha$, which in turn implies that $\beta(j)$ converges to 0.

**Lemma 8.0.1** *The sequences $\bar{\alpha}(j)$ and $\underline{\alpha}(j)$ converge to $\alpha$ as $j \to \infty$. Further, the sequence $\beta(j)$ converges to $0$ as $j \to \infty$.*

**Proof** From (32) it is given $\{x(k)\}_{k=1}^{\infty}$ converges to $\alpha$ i.e.

$$\lim_{k\to\infty} x_i(k) = \alpha$$

for all $i = 1, 2, \cdots N$. Thus, for any $\epsilon > 0$ there exists $K$ such that for all $k \geq K$ implies:

$$|x_i(k) - \alpha| < \epsilon \text{ for all } i = 1, 2, \cdots N$$

$$\Rightarrow -\epsilon < x_i(k) - \alpha < \epsilon \text{ for all } i = 1, 2, \cdots N$$

$$\Rightarrow -\epsilon < \max x(k) - \alpha < \epsilon$$

$$\Rightarrow |\max x(k) - \alpha| < \epsilon$$

$$\Rightarrow \lim_{k\to\infty} \max x(k) = \alpha$$

$$\text{Similarly, } \lim_{k\to\infty} \min x(k) = \alpha$$

Now, $\alpha(\bar{j}) = \max x_i(jD)$ and $\underline{\alpha}(j) = \min x_i(jD)$. So, they are subsequences of convergent sequences converging to same limit $\alpha$, thus both $\bar{\alpha}(j)$ and $\underline{\alpha}(j)$ converge to $\alpha$ as $\bar{j} \to \infty$. Further, note that $\beta(j) = \bar{\alpha}(j) - \underline{\alpha}(j)$, therefore $\beta(j)$ converges to 0 as $j \to \infty$. ■

This leads to the following distributed algorithm, which is the main result of the paper. It helps each node in deducing the occurrence of convergence in the network in finite time within desired error margin $\rho$.

Algorithm I:

*Initialization*: Given initial condition $x(0)$, set $z(0) = x(0)$ and $y(0) = x(0)$. Start $AP(x(0))$, $MXP(z(0))$ and $MNP(y(0))$. Set $j = 1$.

*Step 1*: At $k = T(j) + D$, let $\bar{\alpha}(j) = MXP(z(T(j))), \underline{\alpha}(j) = MNP(y(T(j)))$ and $\beta(j) = \bar{\alpha}(j) - \underline{\alpha}(j)$. Check at each node if $\beta(j) < \rho$; If yes then Stop, else set $j = j + 1$.

*Step 2*: At $k = T(j)$, set $z(T(j)) = x(T(j))$ and $y(T(j)) = x(T(j))$. Go to Step 1.

Next theorem with help of Lemma 8.0.1 shows that the Algorithm I terminates in finite time.

**Theorem 8.0.1** *Algorithm I terminates in some finite time $T_c < \infty$.*

**Proof** $\beta(j)$ converges to 0 as $j \to \infty$ (from Lemma 8.0.1). Thus, for any given $\rho > 0$, there exists an integer $j_0$ such that $\beta(j) < \rho$ for all $j \geq j_0$. This implies that the Algorithm I converges in finite time $T_c = T(j_0)$. ■

The finite time $T_c$ is not known beforehand because the size of $\beta(j)$'s which is proportional to the algebraic connectivity of the graph is not known to each node beforehand. The significant achievement is that all nodes can deduce in some finite time that the consensus in the network has reached and this happens at the same time at each node without help of any centralized information or source.

In above algorithm, the maximum and minimum protocols are getting reset after every $D$ time. The value of $j$ at the termination of above algorithm gives number of times maximum and minimum protocols are executed. This number can be reduced at the cost of delaying the detection of convergence by choosing $T(j) = (j-1)D + \Delta T_j$ where $\Delta T_j \geq 0$ for all $j = 1, 2, \cdots$. One heuristic way to choose $\Delta T_j$ is by estimating the rate of decrease in the difference between maximum and minimum of node-values and setting $\Delta T_j$ equal to that estimated rate. In fact, above algorithm should work for all the graphs with diameter bounded by $D_{max}$ at the expense of delaying the detection of occurrence of convergence by time bounded by the difference between $D_{max}$ and actual diameter $D$. In other words, for this scheme to work it is not required for each node to know the actual diameter of the graph instead all it needs is some upper bound value on the diameter. In (27) a distributed method for computing the diameter of a graph is presented which uses a maximum of $2N^2$ messages. Each node can first run this protocol to determine $D$ in a distributed manner. Diameter $D$ is the only parameter of the network graph required by each node.

It should be noted that this scheme needs at most three times the amount of data to be communicated between the nodes before the convergence is detected. Communication and computation efforts can further be reduced by adjusting $\Delta T_j$. After the convergence is detected, each node can stop communicating any data to neighboring node (as long as its own node-value does not change from the steady-state value) and just listen for any new information from its neighboring nodes.

Further, this detection of convergence technique can be generalized to distributed protocols such that $x(k)$ satisfies Lemma 7.1.2, i.e. the maximum and minimum of $x(k)$ over all nodes is strictly decreasing and increasing after every finite time $D$.

# CHAPTER 9. NUMERICAL EXAMPLES

In this Chapter, three scenarios of averaging protocol are presented to show how the algorithm presented in the previous Chapter facilitates distributed detection of occurrence of consensus in the network.

Scenario A is of an undirected graph $G_1$ with 25 nodes. The diameter of graph is 4, and the algebraic connectivity of the graph is 1.79. It has maximum degree of 12 and minimum degree of 2. The initial condition $x(0)$ is chosen from a uniform distribution between $+10$ and $-10$. The average value $\alpha = 0.95$. Each node comes to know when the consensus has reached within an error margin of $\rho = 0.02$. The simulation result shown in Figure 9.1 demonstrates that after $k = 45$, each node correctly concludes that the convergence has occurred in the network within an error margin of 0.02.

Scenario B is another undirected graph $G_2$ with 25 nodes, diameter of the graph is 4, the algebraic connectivity of the graph is 1.48. It has maximum degree of 11 and minimum degree of 2. The average value $\alpha = -0.84$. In this case, after consensus has reached and detected, there is some change in one of node-values at $k = 60$, so that the new average value becomes $-0.65$. The algorithm presented in previous Chapter starts automatically to find another occurrence of convergence due to this change. The simulation result shown in Figure 9.2 demonstrates that each node comes to know when the consensus has reached within an error margin of 0.02 at $k = 85$.

Scenario C is of a directed, strongly connected and balanced graph $G_3$ with 25 nodes. The diameter of graph is 11 , and the algebraic connectivity of the graph is 0.17. It has maximum degree of 5 and minimum degree of 1. The initial condition $x(0)$ is chosen from a uniform distribution between $+10$ and $-10$. The average value $\alpha = 1.07$. The simulation result shown in Figure 9.3 demonstrates that each node comes to know when the consensus has reached within an error margin of $\rho = 0.02$ at $k = 96$.

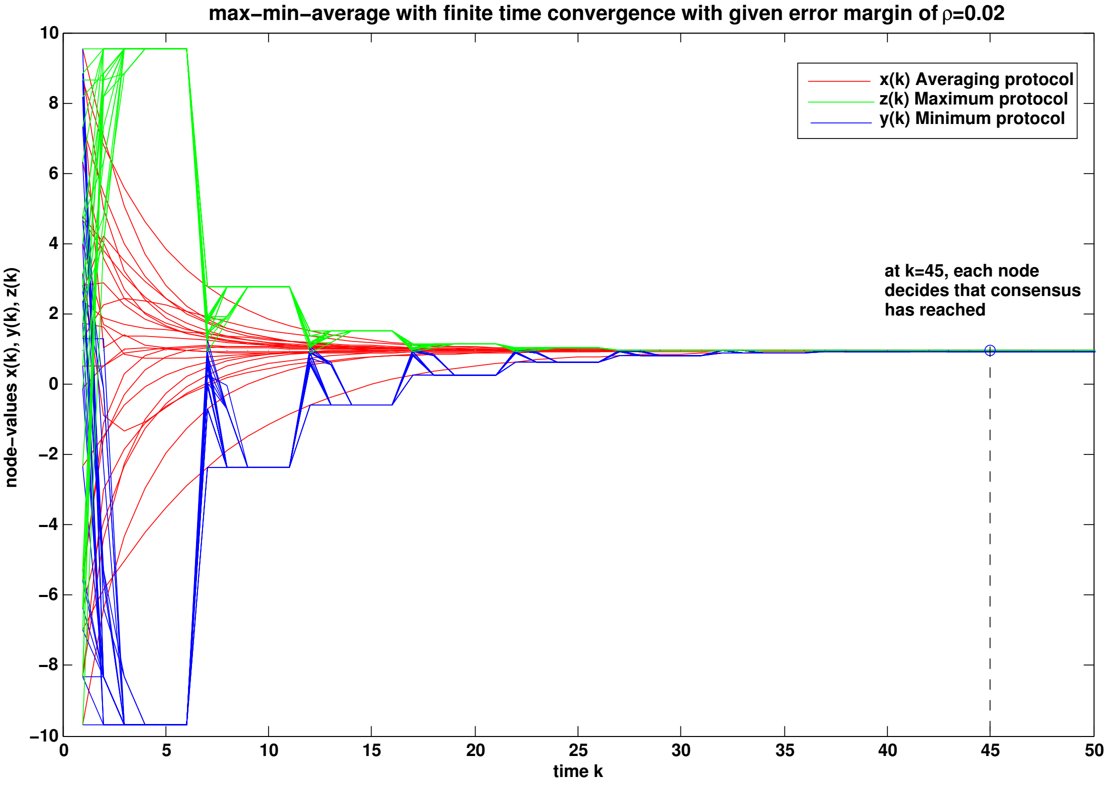


Figure 9.1 A: Maximum-minimum protocol running in parallel with averaging protocol helps individual agents to make a decision about the occurrence of agreement in the network

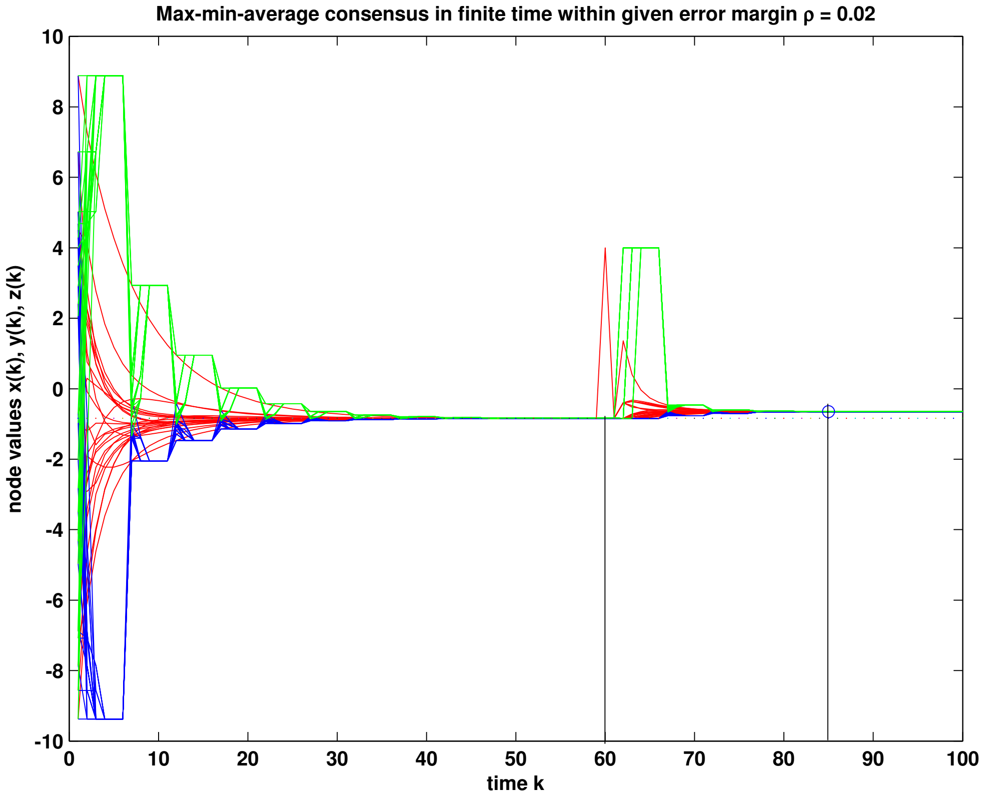


Figure 9.2 B: One node changes it value at $k = 60$ after the agreement has reached in the network. Maximum-minimum protocol and averaging protocol restart to detect next occurrence of agreement in the network

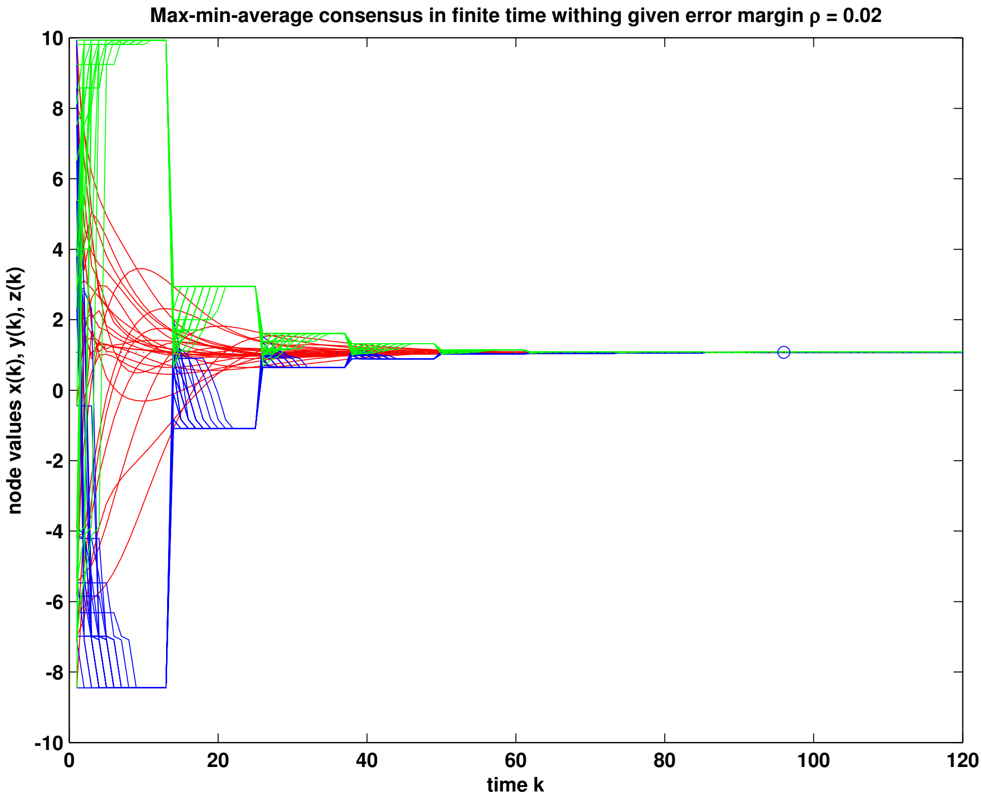


Figure 9.3 C: A case of directed graph. Maximum-minimum protocol running in parallel with averaging protocol helps individual agents to make a decision about the occurrence of agreement in the network

# PART III

# APPENDIX AND BIBLIOGRAPHY

# APPENDIX A.

## A.1 Example: Unstable closed-loop system with sub-controller communication

Consider a generalized plant $G$ with exogenous input $w$ , control input $u$, measured output $y$ and regulated output $z = y$.

$$\begin{aligned} G &= \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} = \left(\begin{array}{c|ccc} \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & 0 \\ 0 & 0 & \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} \\ \hline \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} & 0 \\ 0 & 0 & \frac{-\lambda}{1-\lambda} & \frac{\lambda}{1-\lambda} \end{array}\right) \\ y &= G_{22}u + G_{21}w \\ u &= Ky \end{aligned}$$

The plant has triangular structure as shown in Figure 2.7. The stabilizing controller $K$ (which is also triangular i.e. $K = \begin{bmatrix} K_{11} & 0 \\ K_{21} & K_{22} \end{bmatrix}$) is implemented distributively as shown in Figure 2.7 where $t$ is the sub-controller transmission signal from $K_1$ to $K_2$. The controller is distributively implemented such that $K_1 = \begin{bmatrix} K_{11} \\ I \end{bmatrix}$ and $K_2 = \begin{bmatrix} K_{21} & K_{22} \end{bmatrix}$. In this implementation, $t = K_{21}y$ and $\Phi_{tw} = K_{21}\Phi_{y_1 w}$.

The state space representation of $G_{22}$ is given by:

$$G_{22} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \left(\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{array}\right)$$

with $F = \begin{pmatrix} -0.8 & 0 \\ -0.1 & 0 \\ 0 & -0.9 \end{pmatrix}$ and $L = \begin{pmatrix} -1.9 & 0 \\ 0 & -1.9 \end{pmatrix}$ such that $A+BF$ and $A+LC$ are stable. Using this, $G_{22}$ can be factorize in terms of its coprime factors $X_r, X_l, Y_r, Y_l, M_r, M_l, N_r$ and $N_l$.

$$X_r = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} = \left(\begin{array}{c|c} \frac{1+1.8\lambda}{1-0.1\lambda} & 0 \\ \hline \frac{0.19\lambda^2}{(1-0.1\lambda)^2} & \frac{1+1.8\lambda}{1-0.1\lambda} \end{array}\right), Y_r = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} = \left(\begin{array}{c|c} \frac{-1.52\lambda}{1-0.1\lambda} & 0 \\ \frac{-0.19\lambda}{1-0.1\lambda} & 0 \\ \hline \frac{-0.17\lambda^2}{(1-0.1\lambda)^2} & \frac{-1.71\lambda}{1-0.1\lambda} \end{array}\right)$$

$$X_l = \begin{pmatrix} \widetilde{X}_{11} & \widetilde{X}_{12} \\ \widetilde{X}_{21} & \widetilde{X}_{22} \end{pmatrix} = \left(\begin{array}{cc|c} \frac{-1-1.7\lambda}{1+0.9\lambda} & \frac{-0.8\lambda}{1+0.9\lambda} & 0 \\ \frac{-0.1\lambda}{1+0.9\lambda} & \frac{-1-\lambda}{1+0.9\lambda} & 0 \\ \hline 0 & \frac{0.9\lambda}{1+0.9\lambda} & \frac{-1-1.8\lambda}{1+0.9\lambda} \end{array}\right), Y_l = \begin{pmatrix} \widetilde{Y}_{11} & \widetilde{Y}_{12} \\ \widetilde{Y}_{21} & \widetilde{Y}_{22} \end{pmatrix} = \left(\begin{array}{c|c} \frac{1.52\lambda}{1+0.9\lambda} & 0 \\ \frac{1.9\lambda}{1+0.9\lambda} & 0 \\ \hline 0 & \frac{1.71\lambda}{1+0.9\lambda} \end{array}\right)$$

$$M_l = \begin{pmatrix} \widetilde{M}_{11} & \widetilde{M}_{12} \\ \widetilde{M}_{21} & \widetilde{M}_{22} \end{pmatrix} = \left(\begin{array}{c|c} \frac{1-\lambda}{1+0.9\lambda} & 0 \\ \hline 0 & \frac{1-\lambda}{1+0.9\lambda} \end{array}\right)$$

All stabilizing $K$ can be parameterized in terms of youla parameter $Q$. And, by setting $Q = 0$ one such stabilizing controller can be obtained as $K = Y_r X_r^{-1}$. With this controller, the closed-loop map $\Phi_{zw} = H - UQV = H = G_{11} + G_{12} Y_r M_l G_{21}$ can be written as

$$\Phi_{zw} = \begin{pmatrix} \Phi_{z_1 w} \\ \Phi_{z_2 w} \end{pmatrix} = \begin{pmatrix} \Phi_{y_1 w} \\ \Phi_{y_2 w} \end{pmatrix} = \begin{pmatrix} \frac{-\lambda(1+1.8\lambda)}{(1-0.1\lambda)(1+0.9\lambda)} \\ \frac{-0.19\lambda^3}{(1-0.1\lambda)^2(1+0.9\lambda)} \end{pmatrix}$$

Note that $z = y$ i.e. $z = (z_1', z_2')' = (y_1', y_2')'$ and $\Phi_{y_1 w}$ has only one unstable zero at $z = -1.8$. Thus, the close loop map from $w$ to $z$ is stable. Next, it is shown that with this controller, the distributed implementation is not internally stable by showing that the map $\Phi_{tw}$ from $w$

to sub-controller transmission signal $t$ is not stable.

$$\begin{aligned}\Phi_{tw} &= K_{21} * \Phi_{y_1w} = \widetilde{X}_{22}^{-1}[\widetilde{X}_{22}Y_{21} - \widetilde{Y}_{22}X_{21}]X_{11}^{-1} * \Phi_{y_1w} \\ &= \frac{-\lambda^2(0.17-0.2\lambda)}{(1-0.1\lambda)(1+1.8\lambda)^2} * \frac{-\lambda(1+1.8\lambda)}{(1-0.1\lambda)(1+0.9\lambda)} = \frac{\lambda^3(0.17-0.2\lambda)}{(1-0.1\lambda)^2(1+1.8\lambda)(1+0.9\lambda)}\end{aligned}$$

which is unstable, because $K_{21}$ has two unstable poles at $-1.8$ while $\Phi_{y_1w}$ has only one unstable zero at $z = -1.8$.

## A.2 Proof of Theorem 2.1.1

**Theorem 2.1.1**: *Consider the $G - K_1 - K_2$ interconnection shown in Figure 2.1 where $G = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right]$ is the generalized plant. Let, $K_1$ and $K_2$ have stabilizable and detectable state space realizations such that the induced realization of controller $K$ is stabilizable and detectable. Assuming that the interconnection with distributed implementation using $K_1$ and $K_2$ is well-posed. Given that the inherited realization of $G_{22}$ from $G$ is stabilizable and detectable, $G-K_1-K_2$ interconnection shown in Figure 2.1 is internally stable if and only if $G_{22}-K_1-K_2$ interconnection shown in Figure 2.2 is internally stable.*

**Proof** Assuming that $G - K_1 - K_2$ interconnection shown in Figure 2.1 is internally stable, that is the map from $(w', v_1', v_2', v_3', v_4', v_5', v_6')'$ to $(z', u_1', u_2', y_1', y_2', t_1', t_2')'$ is stable, this implies that the map from $(v_1', v_2', v_3', v_4', v_5', v_6')'$ to $(u_1', u_2', y_1', y_2', t_1', t_2')'$ is also stable. Thus, $G_{22} - K_1 - K_2$ interconnection shown in Figure 2.2 is internally stable.

Now, assume that $G_{22} - K_1 - K_2$ interconnection shown in Figure 2.2 is internally stable. Since, $K_1$ and $K_2$ have stabilizable and detectable state space realizations such that the induced realization of controller $K$ is stabilizable and detectable; and the inherited realization of $G_{22}$ is stabilizable and detectable, the closed-loop $A-$matrix of $G_{22} - K$ interconnection is Hurwitz. It can be shown that the closed-loop $A-$matrix of $G - K_1 - K_2$ interconnection is same as that of $G_{22} - K_1 - K_2$ interconnection. Thus, $G - K_1 - K_2$ interconnection shown in Figure 2.1 is internally stable. ∎

## A.3 Proof of Theorem 2.1.2

**Theorem 2.1.2**: *Consider the $G_{22}-K_1-K_2$ interconnection given in Figure 2.2 where sub-controllers $K_1$ and $K_2$ are distributed implementation of the centralized stabilizing controller $K$. The $G_{22}-K_1-K_2$ interconnection is internally stabilizing if any stabilizable and detectable realization of $K_1$ and $K_2$ is such that the induced realization of $K$ is stabilizable and detectable.*

**Proof** Consider the $G_{22}-K_1-K_2$ interconnection given in Figure 2.2 where sub-controllers $K_1$ and $K_2$ are distributed implementation of the centralized stabilizing controller $K$. Let, $K_1$ and $K_2$ have stabilizable and detectable state space realizations given by $\left[\begin{array}{c|c} A_{C1} & B_{C1} \\ \hline C_{C1} & 0 \end{array}\right]$ and $\left[\begin{array}{c|c} A_{C2} & B_{C2} \\ \hline C_{C2} & 0 \end{array}\right]$, respectively. The interconnection with distributed implementation using $K_1$ and $K_2$ is assumed to be well-posed. Let $v = (v_3', v_4', v_1', v_6', v_2', v_5')'$ be set of all external signals, $r = (u_1', u_2', y_1', t_1', y_2', t_2')'$ be set of all internal signals as shown in Figure 2.1, and $H(G_{22}, K_1, K_2)$ be the closed-loop map from $v$ to $r$. Let, $G_{22}$ have minimal realization given by $\left[\begin{array}{c|c} A & B_2 \\ \hline C_2 & 0 \end{array}\right]$. Then,

$$\begin{aligned}
G_{22} : x^+ &= Ax + B_2 \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \\
\begin{pmatrix} e_3 \\ e_4 \end{pmatrix} &= C_2 x \\
K_1 : x_1^+ &= A_{C1} x_1 + B_{C1} \begin{pmatrix} y_1 \\ t_1 \end{pmatrix} \\
\begin{pmatrix} e_1 \\ e_6 \end{pmatrix} &= C_{C1} x_1 \\
K_2 : x_2^+ &= A_{C2} x_2 + B_{C2} \begin{pmatrix} y_2 \\ t_2 \end{pmatrix} \\
\begin{pmatrix} e_2 \\ e_5 \end{pmatrix} &= C_{C2} x_2
\end{aligned}$$

where $x, x_1$ and $x_2$ are states of $G_{22}, K_1$ and $K_2$, respectively; and $e_1 = u_1 - v_1, e_2 = u_2 - v_2, e_3 = y_1 - v_3, e_4 = y_2 - v_4, e_5 = t_1 - v_5$ and $e_6 = t_2 - v_6$.

Substituting this in above state space equations, the state space realization of map $\bar{H}$ from r to v can be obtained as described below:

$$\begin{array}{rcl} x^{+} & = & Ax + B_2 \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \\ \begin{pmatrix} v_3 \\ v_4 \end{pmatrix} & = & -C_2 x + \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ x_1^{+} & = & A_{C1} x_1 + B_{C1} \begin{pmatrix} y_1 \\ t_1 \end{pmatrix} \\ \begin{pmatrix} v_1 \\ v_6 \end{pmatrix} & = & -C_{C1} x_1 + \begin{pmatrix} u_1 \\ t_2 \end{pmatrix} \\ x_2^{+} & = & A_{C2} x_2 + B_{C2} \begin{pmatrix} y_2 \\ t_2 \end{pmatrix} \\ \begin{pmatrix} v_2 \\ v_5 \end{pmatrix} & = & -C_{C2} x_2 + \begin{pmatrix} u_2 \\ t_2 \end{pmatrix} \end{array}$$

Thus, $\bar{H}$ admits state space realization given by $\bar{H}$

$$= \left[\begin{array}{c|c} \overbrace{\begin{matrix} A & 0 & 0 \\ 0 & A_{C1} & 0 \\ 0 & 0 & A_{C2} \end{matrix}}^{\tilde{A}} & \overbrace{\begin{matrix} B_2 & 0 & 0 \\ 0 & B_{C1} & 0 \\ 0 & 0 & B_{C2} \end{matrix}}^{\tilde{B}} \\ \hline \underbrace{\begin{matrix} -C_2 & 0 & 0 \\ 0 & -C_{C1} & 0 \\ 0 & 0 & -C_{C2} \end{matrix}}_{\tilde{C}} & \underbrace{\begin{matrix} 0 & J_1 & J_3 \\ J_1 & 0 & J_2 \\ J_3' & J_2 & 0 \end{matrix}}_{\tilde{D}} \end{array}\right],$$

where $J_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $J_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ and $J_3 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$. Note that $H = \bar{H}^{-1}$, thus $H = \left[\begin{array}{c|c} \tilde{A} - \tilde{B}\tilde{D}^{-1}\tilde{C} & -\tilde{B}\tilde{D}^{-1} \\ \hline \tilde{D}^{-1}\tilde{C} & \tilde{D}^{-1} \end{array}\right] =: \left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & \bar{D} \end{array}\right]$. It can be shown that $\bar{A}, \bar{B}, \bar{C}$ and $\bar{D}$ is a stabilizable and detectable realization of $H$. Thus, the distributive implementation shown in

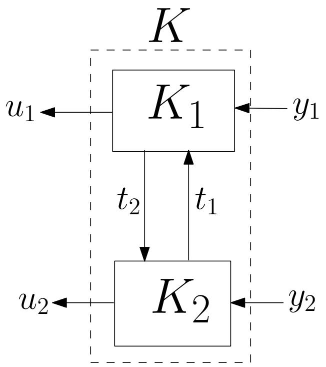


Figure A.1 Distributed implementation of K

Figure 2.2 is internally stable (i.e. $H$ is stable) if and only if $\bar{A}$ is Hurwitz.

Without loss of generality, let $B_2 = \begin{bmatrix} B_{21} & B_{22} \end{bmatrix}$, $B_{C1} = \begin{bmatrix} B_{C11} & B_{C12} \end{bmatrix}$ and $B_{C2} = \begin{bmatrix} B_{C21} & B_{C22} \end{bmatrix}$. Similarly, let $C_2 = \begin{bmatrix} C_{21} \\ C_{22} \end{bmatrix}$, $C_{C1} = \begin{bmatrix} C_{C11} \\ C_{C12} \end{bmatrix}$ and $C_{C2} = \begin{bmatrix} C_{C21} \\ C_{C22} \end{bmatrix}$. Using this we can rewrite $\bar{A}$ as:

$$\bar{A} = \begin{bmatrix} A & B_{21}C_{C11} & B_{22}C_{C21} \\ B_{C11}C_{21} & A_{C1} & B_{C12}C_{C22} \\ B_{C21}C_{22} & B_{C22}C_{C12} & A_{C2} \end{bmatrix} \tag{A.1}$$

The induced realization of $K$ as shown in Figure A.1 can be computed using the stabilizable and detectable realizations of $K_1$ and $K_2$.

$$\begin{aligned} K_1 : x_1^+ &= A_{C1}x_1 + B_{C1}\begin{pmatrix} y_1 \\ t_1 \end{pmatrix} \\ \begin{pmatrix} u_1 \\ t_2 \end{pmatrix} &= C_{C1}x_1 \\ K_2 : x_2^+ &= A_{C2}x_2 + B_{C2}\begin{pmatrix} y_2 \\ t_2 \end{pmatrix} \\ \begin{pmatrix} u_2 \\ t_1 \end{pmatrix} &= C_{C2}x_2 \end{aligned}$$

By substituting $t_1 = C_{C22}x_1$ and $t_2 = C_{C12}x_2$ we get induced realization of K as described below:

$$\begin{aligned} K: x_k^+ &= \overbrace{\begin{pmatrix} A_{C1} & B_{C12}C_{C22} \\ B_{C22}C_{C12} & A_{C2} \end{pmatrix}}^{A_K} x_k + \overbrace{\begin{pmatrix} B_{C11} & 0 \\ 0 & B_{C21} \end{pmatrix}}^{B_K} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} &= \underbrace{\begin{pmatrix} C_{C11} & 0 \\ 0 & C_{C21} \end{pmatrix}}_{C_K} x_k \end{aligned}$$

where $x_k = (x_1', x_2')'$. If induced realization of $K$ is stabilizable and detectable and $K$ is a stabilizing controller i.e. the closed-loop map is stable, then A-matrix of $G_{22} - K$ interconnection, denoted by $A_{cl}$, is Hurwitz. $A_{cl}$ can be written in terms of state space realization of $G_{22}$ and $K$ as

$$A_{cl} = \begin{bmatrix} A & B_2C_K \\ B_KC_2 & A_K \end{bmatrix} = \begin{bmatrix} A & B_{21}C_{C11} & B_{22}C_{C21} \\ B_{C11}C_{21} & A_{C1} & B_{C12}C_{C22} \\ B_{C21}C_{22} & B_{C22}C_{C12} & A_{C2} \end{bmatrix} \qquad \text{(A.2)}$$

which is same as $\bar{A}$, thus $\bar{A}$ is also Hurwitz, and the map $H$ is stable. ∎

## A.4 Proof of Corollary 2.1.3

**Corollary 2.1.3**: *Consider a 2-nest $G_{22} - K$ interconnection where $K$ is a centralized stabilizing controller implemented in distributive manner using sub-controllers $K_1$ and $K_2$ as shown in Figure 2.2 with $t_2 = t$ and no transmission from $K_2$ to $K_1$. Let, $K_1$ and $K_2$ have state space realizations given by Let, $K_1$ and $K_2$ have state space realizations given by* $K_1 = \left[\begin{array}{c|c} A_{C1} & B_{C1} \\ \hline \begin{matrix} C_{C11} \\ C_{C12} \end{matrix} & 0 \end{array}\right]$ *and* $K_2 = \left[\begin{array}{c|cc} A_{C2} & B_{C21} & B_{C22} \\ \hline C_{C2} & \multicolumn{2}{c}{0} \end{array}\right]$ *such that* $(A_{C1}, B_{C1}, C_{C11})$ *and* $(A_{C2}, B_{C21}, C_{C2})$ *are stabilizable and detectable. Then, the induced realization of con-*

*troller $K$ obtained from $K_1$ and $K_2$ is stabilizable and detectable and $G_{22} - K$ interconnection with distributed implementation is internally stable.*

**Proof** Let, $G_{22}$ have minimal realization given by $\left[\begin{array}{c|c} A & B_2 \\ \hline C_2 & 0 \end{array}\right]$, where $B_2 = \begin{bmatrix} B_{21} & B_{22} \end{bmatrix}$, $C_2 = \begin{bmatrix} C_{21} \\ C_{22} \end{bmatrix}$. Then, the induced state-space realization of $K$ is given by

$$\begin{aligned} K : x_k^+ &= \overbrace{\begin{pmatrix} A_{C1} & 0 \\ B_{C22}C_{C12} & A_{C2} \end{pmatrix}}^{A_K} x_k + \overbrace{\begin{pmatrix} B_{C1} & 0 \\ 0 & B_{C21} \end{pmatrix}}^{B_K} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} &= \underbrace{\begin{pmatrix} C_{C11} & 0 \\ 0 & C_{C2} \end{pmatrix}}_{C_K} x_k. \end{aligned}$$

Since, $(A_{C1}, B_{C1})$ and $(A_{C2}, B_{C21})$ are stabilizable pairs, the Popov-Belevitch-Hautus (PBH) test (21) implies that $\begin{bmatrix} \lambda I - A_{C1} & B_{C1} \end{bmatrix}$ and $\begin{bmatrix} \lambda I - A_{C2} & B_{C21} \end{bmatrix}$ have full row rank. Note that

$$\begin{bmatrix} -B_{C22}C_{C12} & \lambda I - A_{C2} & 0 & B_{C21} \end{bmatrix} = \begin{bmatrix} \lambda I - A_{C2} & B_{C21} & B_{C22} \end{bmatrix} \begin{bmatrix} 0 & I & 0 & 0 \\ -C_{C12} & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

where $C_{C12}$ has full row rank, this implies that $\begin{bmatrix} -B_{C22}C_{C12} & \lambda I - A_{C2} & 0 & B_{C21} \end{bmatrix}$ has full row rank. It can be shown that $\begin{bmatrix} \lambda I - A_{C1} & 0 & B_{C1} & 0 \\ -B_{C22}C_{C12} & \lambda I - A_{C2} & 0 & B_{C21} \end{bmatrix}$ has full row rank, implying that $(A_K, B_K)$ is stabilizable pair. Similarly, it can be shown that $(A_K, C_K)$ is detectable. Now, from Lemma 1 it implies that $G_{22} - K$ interconnection with distributed implementation is internally stable. In order to show that $\begin{bmatrix} \lambda I - A_{C1} & 0 & B_{C1} & 0 \\ -B_{C22}C_{C12} & \lambda I - A_{C2} & 0 & B_{C21} \end{bmatrix}$ has full row rank, without loss of any generality, assume that $A_{C1}$ and $A_{C2}$ are in Jordan Canonical form.

The only way this matrix can loose rank is if for some $\lambda$ any row of $\begin{bmatrix} \lambda I - A_{C1} & 0 & B_{C1} & 0 \end{bmatrix}$ $(:= P1)$ becomes similar to one of the rows of $\begin{bmatrix} -B_{C22}C_{C12} & \lambda I - A_{C2} & 0 & B_{C21} \end{bmatrix}$ $(:= P2)$.

- Let the row of $P1$ which corresponds to the last row of a Jordan block of $A_{C1}$ be $R1_{Jl}$, then the part of $R1_{Jl}$ which corresponds to $B_{C1}$ block is non-zero for $(A_{C1}, B_{C1})$ is stabilizable, thus making $R1_{Jl}$ independent of any row of $P2$.
- Similarly, the rows of $P2$ corresponding to the last row of a Jordan blocks of $A_{C2}$ are linearly independent of any row of $P1$.
- Any row of $P1$ which does not correspond to the last row of a Jordan block is also independent of any row of $P2$ which does not correspond to the last row of a Jordan block of $A_{C2}$, for the part of that row corresponding to $\lambda I - A_{C2}$ block is non-zero.

Thus, for all $\lambda$ the rows of $P1$ and $P2$ are linearly independent, which implies that $\begin{bmatrix} \lambda I - A_{C1} & 0 & B_{C1} & 0 \\ -B_{C22}C_{C12} & \lambda I - A_{C2} & 0 & B_{C21} \end{bmatrix}$ has full row rank. ■

*Remark*: The example of unstable closed-loop map in presence of sub-controller communication presented in the Appendix A.1 does not satisfy the sufficiency condition of this Corollary, thus nothing can be concluded from the Corollary and it is required to compute the closed-loop maps to really check if they are stable or not.

## A.5 Overall controller $K$ as obtained from sub-controllers $K_1$ and $K_2$

Consider the system shown in Figure 3.1 where $K$ is implemented distributively using two sub-controllers $K_1$ and $K_2$ as given by (3.3) and (3.4). Ignore the sub-controller noise i.e.

$n_1 = n_2 = 0$, thus $s_1 = t_1$ and $s_2 = t_2$.

$$\begin{array}{rcll}
t_1 & = & C^1_{21}y_1 + C^1_{22}s_2 & \text{[From (3.3)]} \\
 & = & C^1_{21}y_1 + C^1_{22}t_2 & [\because n_2 = 0] \\
t_2 & = & C^2_{21}y_2 + C^2_{22}s_1 & \text{[From (3.4)]} \\
 & = & C^2_{21}y_2 + C^2_{22}t_1 & [\because n_2 = 0] \\
 & = & C^2_{21}y_2 + C^2_{22}(C^1_{21}y_1 + C^1_{22}t_2) & \text{[Substituting for } t_1 \text{ from above]} \\
\Rightarrow t_2 & = & (I - C^2_{22}C^1_{22})^{-1}(C^2_{22}C^1_{21}y_1 + C^2_{21}y_2) & \\
e_1 & = & C^1_{11}y_1 + C^1_{12}s_2 = C^1_{11}y_1 + C^1_{12}t_2 & \text{[From (3.3) and } n_2 = 0] \\
 & = & C^1_{11}y_1 + C^1_{12}(I - C^2_{22}C^1_{22})^{-1}(C^2_{22}C^1_{21}y_1 + C^2_{21}y_2) & \\
 & = & [C^1_{11} + C^1_{12}(I - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{21}]y_1 + [C^1_{12}(I - C^2_{22}C^1_{22})^{-1}C^2_{21}]y_2 & \\
 & = & K_{11}y_1 + K_{12}y_2 &
\end{array}$$

Similarly,

$$\begin{array}{rcll}
e_2 & = & C^2_{11}y_2 + C^2_{12}s_1 = C^2_{11}y_2 + C^2_{12}t_1 & \text{[From (3.4) and } n_1 = 0] \\
 & = & [C^2_{12}(I - C^1_{22}C^2_{22})^{-1}C^1_{21}]y_1 + [C^2_{11} + C^2_{12}(I - C^1_{22}C^2_{22})^{-1}C^1_{22}C^2_{21}]y_2 & \\
 & = & K_{21}y_1 + K_{22}y_2 &
\end{array}$$

Thus,

$$\begin{array}{rcl}
K_{11} & = & C^1_{11} + C^1_{12}(I - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{21} \\
K_{12} & = & C^1_{12}(I - C^2_{22}C^1_{22})^{-1}C^2_{21} \\
K_{21} & = & C^2_{12}(I - C^1_{22}C^2_{22})^{-1}C^1_{21} \\
K_{22} & = & C^2_{11} + C^2_{12}(I - C^1_{22}C^2_{22})^{-1}C^1_{22}C^2_{21}
\end{array}$$

as given by (3.5) and (3.8).

## A.6 Obtaining $K_n, K_t$ and $K_{tn}$ from sub-controllers $K_1$ and $K_2$

Consider the same Figure 3.1 where $K$ is implemented distributively using two sub-controllers $K_1$ and $K_2$ as given by (3.3) and (3.4). $K_n$ is the part which maps $(n_1', n_2')^T$ to $(e_1', e_2')^T$, $K_t$ is the part which maps $(y_1', y_2')^T$ to $t = (t_1', t_2')^T$ and $K_{tn}$ maps $(n_1', n_2')^T$ to $t$. Thus,

$$\begin{aligned}
\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} &= K_n \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \\
\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} &= K_{tn} \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \\
\begin{pmatrix} t_1 \\ t_2 \end{pmatrix} &= K_t \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}
\end{aligned}$$

In order to find $K_n$ and $K_{tn}$, set $y = 0$ in (3.3) and (3.4).

$$\begin{aligned}
e_1 &= C^1_{12} s_2 = C^1_{12}(t_2 + n_2) \\
t_1 &= C^1_{22}(t_2 + n_2) = C^1_{22} t_2 + C^1_{22} n_2 \\
t_2 &= C^2_{22}(t_1 + n_1) = C^2_{22} t_1 + C^2_{22} n_1 \\
&= C^2_{22}(C^1_{22} t_2 + C^1_{22} n_2) + C^2_{22} n_1 \\
\Rightarrow t_2 &= (I - C^2_{22} C^1_{22})^{-1}(C^2_{22} C^1_{22} n_2 + C^2_{22} n_1) \\
\Rightarrow e_1 &= C^1_{12}(I - C^2_{22} C^1_{22})^{-1}(C^2_{22} C^1_{22} n_2 + C^2_{22} n_1) + C^1_{12} n_2 \\
&= [C^1_{12}(I - C^2_{22} C^1_{22})^{-1} C^2_{22}] n_1 + [C^1_{12} + C^1_{12}(I - C^2_{22} C^1_{22})^{-1} C^2_{22} C^1_{22}] n_2 \\
&= [C^1_{12}(I - C^2_{22} C^1_{22})^{-1} C^2_{22}] n_1 + [C^1_{12}(I - C^2_{22} C^1_{22})^{-1}] n_2.
\end{aligned}$$

Similarly

$$\begin{aligned}
e_2 &= [C^2_{12}(I - C^1_{22} C^2_{22})^{-1}] n_1 + [C^2_{12}(I - C^1_{22} C^2_{22})^{-1} C^1_{22}] n_2. \\
\Rightarrow \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} &= \begin{bmatrix} C^1_{12}(1 - C^2_{22} C^1_{22})^{-1} C^2_{22} & C^1_{12}(1 - C^2_{22} C^1_{22})^{-1} \\ C^2_{12}(1 - C^1_{22} C^2_{22})^{-1} & C^2_{12}(1 - C^1_{22} C^2_{22})^{-1} C^1_{22} \end{bmatrix} \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \\
&:= K_n \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}
\end{aligned}$$

From $t_2 = (I - C^2_{22}C^1_{22})^{-1}(C^2_{22}C^1_{22}n_2 + C^2_{22}n_1)$, $t_1$ can be obtained in terms of $n$ as follows:

$$\begin{aligned}
t_1 &= C^1_{22}t_2 + C^1_{22}n_2 \\
&= C^1_{22}((I - C^2_{22}C^1_{22})^{-1}(C^2_{22}C^1_{22}n_2 + C^2_{22}n_1)) + C^1_{22}n_2 \\
&= [C^1_{22}(I - C^2_{22}C^1_{22})^{-1}C^2_{22}]n_1 + [C^1_{22} + C^1_{22}(I - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{22}]n_2 \\
&= [C^1_{22}(I - C^2_{22}C^1_{22})^{-1}C^2_{22}]n_1 + [C^1_{22}(I - C^2_{22}C^1_{22})^{-1}]n_2 \\
\Rightarrow \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} &= \begin{bmatrix} (1 - C^1_{22}C^2_{22})^{-1}C^1_{22}C^2_{22} & (1 - C^1_{22}C^2_{22})^{-1}C^1_{22} \\ (1 - C^2_{22}C^1_{22})^{-1}C^2_{22} & (1 - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{22} \end{bmatrix} \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \\
&:= K_{tn} \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}
\end{aligned}$$

To compute $K_t$ set $n_1 = n_2 = 0$ in (3.3) and (3.4).

$$\begin{aligned}
t_1 &= C^1_{21}y_1 + C^1_{22}s_2 \\
&= C^1_{21}y_1 + C^1_{22}t_2 \\
t_2 &= C^2_{21}y_2 + C^2_{22}s_1 \\
&= C^2_{21}y_2 + C^2_{22}t_1 \\
&= C^2_{21}y_2 + C^2_{22}(C^1_{21}y_1 + C^1_{22}t_2) \\
\Rightarrow t_2 &= (I - C^2_{22}C^1_{22})^{-1}(C^2_{22}C^1_{21}y_1 + C^2_{21}y_2) \\
&= [(I - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{21}]y_1 + [(I - C^2_{22}C^1_{22})^{-1}C^2_{21}]y_2.
\end{aligned}$$

Similarly

$$\begin{aligned}
t_1 &= [(I - C^2_{22}C^1_{22})^{-1}C^1_{21}]y_1 + [(I - C^1_{22}C^2_{22})^{-1}C^1_{22}C^2_{22}]y_2. \\
\Rightarrow \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} &= \begin{bmatrix} (1 - C^2_{22}C^1_{22})^{-1}C^1_{21} & (1 - C^1_{22}C^2_{22})^{-1}C^1_{22}C^2_{21} \\ (1 - C^2_{22}C^1_{22})^{-1}C^2_{22}C^1_{21} & (1 - C^2_{22}C^1_{22})^{-1}C^2_{21} \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\
&:= K_t \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}
\end{aligned}$$

Further, $K_n, K_t$ and $K_{tn}$ can be written as product of two matrices as follows:

$$\begin{aligned}
K_n &= \begin{bmatrix} 0 & C_{12}^1 \\ C_{12}^2 & 0 \end{bmatrix} \begin{bmatrix} I & -C_{22}^1 \\ -C_{22}^2 & I \end{bmatrix}^{-1} && \text{(A.3)} \\
K_t &= \begin{bmatrix} I & -C_{22}^1 \\ -C_{22}^2 & I \end{bmatrix}^{-1} \begin{bmatrix} C_{21}^1 & 0 \\ 0 & C_{21}^2 \end{bmatrix} && \text{(A.4)} \\
K_{tn} &= \begin{bmatrix} 0 & C_{22}^1 \\ C_{22}^2 & 0 \end{bmatrix} \begin{bmatrix} I & -C_{22}^1 \\ -C_{22}^2 & I \end{bmatrix}^{-1}. && \text{(A.5)}
\end{aligned}$$

Let, $K_0 := \begin{bmatrix} C_{11}^1 & 0 \\ 0 & C_{11}^2 \end{bmatrix}$ then overall controller $K$ given by (3.5) and (3.8) can be written in following form:

$$\begin{aligned}
K = K_0 + \begin{bmatrix} 0 & C_{12}^1 \\ C_{12}^2 & 0 \end{bmatrix} \begin{bmatrix} I & -C_{22}^1 \\ -C_{22}^2 & I \end{bmatrix}^{-1} \begin{bmatrix} C_{21}^1 & 0 \\ 0 & C_{21}^2 \end{bmatrix}. && \text{(A.6)} \\
&& \text{(A.7)}
\end{aligned}$$

## A.7 Derivation of important closed-loop maps

Important closed-loop maps from sub-controller noise to internal variables of interconnection as given by equations (3.9)-(3.11) and (3.15) are derived here.

By setting $v_1 = v_2 = v_3 = v_4 = 0$ in the interconnection shown in Figure 3.1, the closed-loop maps from noise $n = (n_1', n_2')^T$ to internal variables $u = (u_1', u_2')^T, y = (y_1', y_2')^T$ and $t = (t_1', t_2')^T$ are obtained as follows:

$$\begin{aligned}
u &= Ky + n = KG_{22}u + K_n n \\
\Rightarrow u &= (I - KG_{22})^{-1} K_n n \\
&:= \Phi_{un} n
\end{aligned}$$

Similarly,

$$\begin{aligned}
y &= G_{22}u = G_{22}(I - KG_{22})^{-1}K_n n \\
&:= \Phi_{yn} n \\
t &= K_t y + K_{tn} n = (K_t G_{22}(I - KG_{22})^{-1}K_n)n + K_{tn} n \\
&= (K_t G_{22}(I - KG_{22})^{-1}K_n + K_{tn})n \\
&:= \Phi_{tn} n
\end{aligned}$$

The closed-loop map from external signals to internal variables at the site of noise injection can be obtained by setting $n = 0$. Then, $t = K_t y$. The closed-loop map form external signals $v = (v_1', v_2', v_3', v_4')^T$ to $y$ is standard map given by $\Phi_{yv} := \left[ \begin{array}{cc} (I - G_{22}K)^{-1}G_{22} & (I - G_{22}K)^{-1} \end{array} \right]$. Thus,

$$\begin{aligned}
t &= K_t y = K_t \Phi_{yv} v \\
&= K_t \left[ \begin{array}{cc} (I - G_{22}K)^{-1}G_{22} & (I - G_{22}K)^{-1} \end{array} \right] v := \Phi_{tv} v
\end{aligned}$$

## A.8 Proof of Lemma 3.2.1

**Lemma 3.2.1**: *The closed-loop map $T(K_1, K_2)$ corresponding to Figure 2.1 interconnection is affine in the Youla parameter $Q$ if the maps $\Phi_{un}$, $\Phi_{tv}$ and $\Phi_{tn}$ are affine in $Q$.*

**Proof** : The closed-loop map $T(K_1, K_2)$ consists of four maps viz. $\Phi_{zw}$, $\Phi_{zn}$, $\Phi_{tw}$ and $\Phi_{tn}$ of which is $\Phi_{zw}$ is the standard map which is stable and affine in $Q$ (18). $\Phi_{tn}$ is affine in $Q$ from the hypothesis.

From $G - K_1 - K_2$ interconnection, the regulated variable $z$ is given by $z = G_{11}w + G_{12}u$. In absence of any exogenous signal other than $n$, $z = G_{12}u$. Thus, $\Phi_{zn} = G_{12}\Phi_{un}$, which is affine in $Q$ if $\Phi_{un}$ is affine in $Q$ for $G_{12}$ does not depend on $Q$.

Further note that $K_t$ is a map from $y$ to $t$. Thus, $\Phi_{tw} = K_t \Phi_{yw}$. From $G - K_1 - K_2$ interconnection, $y$ is given by $y = G_{21}w + G_{22}u = (I - G_{22}K)^{-1}G_{21}w$. This implies that $\Phi_{tw} = K_t(I - G_{22}K)^{-1}G_{21}$ which is is affine in $Q$ if $\Phi_{tv}$ is affine in $Q$ for $G_{21}$ does not depend on $Q$. ∎

# BIBLIOGRAPHY

[1] P. Vettiger, G. Cross, M. Despont, U. Drechsler, U. Durig, B. Gotsmann, W. Haberle, M.A. Lantz, H.E. Rothuizen, R. Stutz, and G.K.; Binnig, “The“millipede” - nanotechnology entering data storage,” *IEEE Transactions on Nanotechnology, Vol. 1, Issue 1, pp. 39-55*, March, 2002.

[2] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S.S.; Sastry, “Distributed control applications within sensor networks,” *Proceedings of IEEE, Vol. 91, Issue 8, pp. 1235-1246*, August, 2003.

[3] D. D. Sourlas and V. Manousiouthakis, “Best achievable decentralized performance,” *IEEE Transactions on Automatic Control, Vol. 40, Issue 11, pp. 1858-1871*, November, 1995.

[4] K. A. Unyelioglu and U. Ozguner, “$\mathcal{H}_\infty$ sensitivity minimization using decentralized feedback: 2-input 2-output systems,” *Systems and Control Letters, pp. 99-109*, 1994.

[5] A. N. Gundes and C. A. Desoer, *Algebraic Theory of Linear Feedback Systems with Full and Decentralized Compensators*, Springer Verlag, Heidelberg, 1990.

[6] P. G. Voulgaris, “Control of nested systems,” *Proceedings of the 2000 IEEE American Control Conference, Vol. 6, pp. 4442-4445, Chicago*, June, 2000.

[7] M. Rotkowitz and S. Lall, “Decentralized control information structures preserved under feedback,” *Proceedings of the 41st IEEE Conference on Decision and Control, Vol. 1, pp. 569-579*, December 2002.

[8] X. Qi, M. V. Salapaka, P. G. Voulgaris, and M. Khammash, "Structured Optimal and Robust Control with Multiple Criteria: A Convex Solution," *IEEE Transactions on Automatic Control, Vol. 49, Issue 10, pp. 1623-1640*, October, 2004.

[9] D. C. Youla, H. A. Jabr, and J. J. Bongiorno, "Modern Wiener-Hopf design of optimal controllers–part 2: The multivariable case," *IEEE Transactions on Automatic Control, Vol. 21, Issue 3, pp. 319-338*, June, 1976.

[10] Raffaello D'Andrea and Geir E. Dullerd;, "Distributed control design for spatially interconnected systems," *IEEE Transactions on Automatic Control, Vol. 48, No. 9, pp. 1478-1495*, September, 2003.

[11] Cedric Langbort, Ramu Sharat Chandra, and Raffaello D'Andrea, "Distributed Control Design for Systems Interconnected Over an Arbitrary Graph," *IEEE Transactions on Automatic Control, Vol. 49, Issue 9, pp. 1502-1519*, September, 2004.

[12] Li Li and Fernando Pagnini;, "LMI relaxation to Riccati equations in the structured $\mathcal{H}_2$ control," *Proceedings of the 25th IEEE American Control Conference, pp. 644-649, Minneapolis, MN*, June, 2006.

[13] Shengxiang Jiang and Petros G. Voulgaris, "Distributed Controller Design Aspects for Guaranteed Performance," *Proceedings of the 44th IEEE Conference on Decision and Control 2005 and 2005 European Control Conference, pp. 4293-4298, Spain*, December, 2005.

[14] V. Yadav, P. G. Voulgaris, and M. V. Salapaka, "Stabilization of nested systems with uncertain subsystem communication channels," *Proceedings of the 42nd IEEE Conference on Decision and Control, Vol. 3, pp. 2853-2858, Hawaii*, December, 2003.

[15] V. Yadav, P. G. Voulgaris, and M. V. Salapaka, "Architectures for distributed control for performance optimization in presence of sub-controller communication noise," *Proceedings of the 25th IEEE American Control Conference, pp. 626-631, Minneapolis, Minnesota*, June, 2006.

[16] V. Yadav, P. G. Voulgaris, and M. V. Salapaka, “Distributed architectures and implementations of observer based controllers for performance optimization,” *Proceedings of the 24th American Control Conference, pp. 4844-4849, Portland, Oregon*, June, 2005.

[17] K. Zhou and J. C. Doyle, *Essentials of Robust Control*, Prentice Hall, Englewood Cliffs, New Jersey, 1998.

[18] B. A. Francis, *A Course in $\mathcal{H}_\infty$ Control Theory*, Springer-Verlag, 1987.

[19] M. H. Khammash and J. B. Pearson, “Performance robustness of discrete-time systems with structured uncertainty,” *IEEE Transactions on Automatic Control, Vol. 36, Issue 4, pp. 398–412*, April, 1991.

[20] M.H. Khammash, M.V. Salapaka, and T. Van Voorhis, “Robust Synthesis in $\ell_1$: A Globally Optimal Solution ,” *IEEE Transactions on Automatic Control, Vol. 46, Issue 11, pp. 1744-1754*, November, 2001.

[21] C. T. Chen, *Linear System Theory and Design*, Oxford University Press, 1999.

[22] D. Angluin, M.J. Fischer, and H. Jiang. Stabilizing Consensus in Mobile Networks. *Proceedings of International Conference on Distributed Computing in Sensor Systems (DCOSS06), pp. 37-50, Springer-Verlag Berlin Heidelberg*, June, 2006.

[23] D. Bauso, L. Giarre, and R. Pesenti. Distributed Consensus in Networks of Dynamic Agents. *Proceedings of the 44th IEEE Conference on Decision and Control 2005 and 2005 European Control Conference, pp. 7054-7059, Spain*, December, 2005.

[24] M. C. DeGennaro and A. Jadbabaie. Decentralized control of connectivity for multi-agent systems. *Proceedings of the 45th IEEE Conference on Decision and Control, pp. 3628-3633, San Diego, California*, December, 2006.

[25] Xiuzhen Cheng Fang Liu and Dechang Chen. Insider Attacker Detection in Wireless Sensor Networks. *IEEE INFOCOM 2007 26th IEEE International Conference on Computer Communication, pp. 1937-1945, Anchorage, Alaska, USA*, May, 2007.

[26] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[27] S. Haldar. An All Pairs Shortest Paths Distributed Algorithm Using 2n2Messages. *Journal of Algorithms, 24(1), pp. 20-36*, 1997.

[28] Y. Hatano, A. K. Das, and M. Mesbahi. Agreement in presence of noise: Pseudogradients on random geometric networks. *Proceedings of the 44th IEE Conference on Decision and Control, pp. 6382-6387*, December, 2005.

[29] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Transactions on Automatic Control, Vol. 50, Issue 11, pp. 1867-1872*, November, 2005.

[30] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control, Vol. 48, Issue 6, pp. 988-1001*, June 2003.

[31] T.K. Moon and W.C. Stirling. *Mathematical methods and algorithms for signal processing.* Prentice Hall, 2000.

[32] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE, Vol. 95, Issue 1, pp. 215-233*, January, 2007.

[33] R. Olfati-Saber and R. M. Murray. Consensus protocols for network of dynamic agents. *IEEE Transactions on Automatic Control, Vol. 49, Issue 9, pp. 1520-1533*, September, 2004.

[34] W. Ren, RW Beard, and EM Atkins. A survey of consensus problems in multi-agent coordination. *Proceedings of the 2005 American Control Conference, pp. 1859-1864, Portland, Oregon*, June, 2005.

[35] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part I: fixed topology. *Proceedings of the 42nd IEEE Conference on Decision and Control, Vol. 2, pp. 2010-2015, Hawaii*, December, 2003.

[36] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part II: dynamical topology. *Proceedings of the 42nd IEEE Conference on Decision and Control, Vol. 2, pp. 2016-2021, Hawaii*, December, 2003.

[37] T. Vicsek, A. Czirok, E. Ben Jacob, I. Cohen, and O. Schochet. Novel type of phase transitions in a system of self-driven particles. *Physical Review Letters, 75:1226-1229*, 1995.

[38] C.W. Wu. Agreement and consensus problems in groups of autonomous agents with linear dynamics. *IEEE International Symposium on Circuits and Systems 2005, pp. 292-295*, 2005.

[39] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Proceedings of the 42nd IEEE Conference on Decision and Control, Vol. 5, pp. 4997-5002, Hawaii*, December, 2003.

# BIOGRAPHY

Vikas Yadav received Bachelor of Technology degree in Electrical Engineering from Indian Institute of Technology, Kanpur (India) in year 2000. From 2000-2001, he worked as a software engineer in a communication software company called Future Software Ltd. in Chennai (India). He had received Master of Science and Doctor of Philosophy degrees in Electrical Engineering (minor in Mathematics) from Iowa State University in year 2007. His research interests concerned distributed control design, studying self organization and phase transition in large scale systems. At the time of writing this biography he is working with Garmin International Inc. based at Olathe, KS (USA). He can be contacted at http://www.vikasyadav.com web site.