A System Level Approach to Optimal Controller Design for Large-Scale Distributed Systems

Thesis by

Yuh-Shyang Wang

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2017
Defended December 7, 2016

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest thank to my advisor, Professor John Doyle, for his enthusiastic guidance and support during my graduate school years. John is an incredibly inspiring scholar. His unique insight into the fundamental problems across a broad range of different fields has broadened my view of scientific research. He gave me many brilliant ideas, encouraged me to pursue my own research directions, and gave me great freedom to work on the most exciting projects that I could ever find. I am so grateful to be his student in my time at Caltech.

I also want to express my thank to my thesis committee: Professors Richard Murray, Adam Wierman, and Soon-Jo Chung. Their comments and suggestions have greatly helped improve the quality of this dissertation. Richard has supported me at each stage of my graduate study, and gave me valuable advices for my career plan. I am also thankful to Professors Steven Low, Lijun Chen, and Venkat Chandrasekaran for introducing me to the field of power network and optimization.

Special thanks to Nikolai Matni for his mentorship in my whole graduate student years. Nikolai works as my “technical advisor" at Caltech. He helped me clarify technical details of my results and guided me step by step in scientific writing and presentation. This thesis would not have been completed without his help. I am also thankful for the guidance from Seungil You in my early time at Caltech, and Changhong Zhao for the questions in power systems.

Thanks to my friends in the CDS department: Yoke Peng, Yorie, Ioannis, Ania, Thomas, Niangjun, Noah, Dimitar, John, and all my friends in the Annenberg community. Thanks to the help from the staff members in the CMS department, especially Maria and Nikki. I would also like to express my thank to all my friends in Caltech, especially from the Association of Caltech Taiwanese and Caltech Badminton Club.

I owe my deepest gratitude to my family: my mother Sheue-Ling Hwang, my father Jhy-Horng Wang, my sister Yung-Hsuan Wang, and my grandparents for their endless love and support. Finally and most importantly, I want to thank my girlfriend and soul-mate, Lily Weng, for her continuous support, care, and love. I dedicate this thesis to my family and Lily for everything they have done for me.

ABSTRACT

Modern cyber-physical systems, such as the smart grid, software-defined networks, and automated highway systems, are large-scale, physically distributed, and interconnected. The scale of these systems poses fundamental challenges for controller design: the traditional optimal control methods are globally centralized, which require solving a large-scale optimization problem with the knowledge of the global plant model, and collecting global measurement instantaneously during implementation. The ultimate goal of distributed control design is to provide a local, distributed, scalable, and coordinated control scheme to achieve centralized control objectives with nearly global transient optimality.

This dissertation provides a novel theoretical and computational contribution to the area of constrained linear optimal control, with a particular emphasis on addressing the scalability of controller design and implementation for large-scale distributed systems. Our approach provides a fundamental rethinking of controller design: we extend a control design problem to a system level design problem, where we directly optimize the desired closed loop behavior of the feedback system. We show that many traditional topics in the optimal control literature, including the parameterization of stabilizing controller and the synthesis of centralized and distributed controller, can all be cast as a special case of a system level design problem. The system level approach therefore unifies many existing results in the field of distributed optimal control, and solves many previously open problems.

Our system level approach has at least the following four technical merits. First, we characterize the broadest known class of constrained linear optimal control problem that admits a convex formulation. Specifically, we show that the set of convex system level design problems is a strict superset of those that can be parameterized using quadratic invariance. Second, we identify a class of system level design problems, which we called the localized optimal control problems, that are scalable to arbitrary large-scale systems. In particular, the parallel synthesis and implementation complexity of the localized optimal controller are $O(1)$ compared to the size of the networked system. Third, we provide a unified framework to simultaneously incorporate user-specified design specification on the closed loop and the hardware implementation constraints on the controller into the optimal controller design process. Lastly, we provide a system level approach that supports the co-design of optimal controller and its sensing and actuating architecture.

We demonstrate the effectiveness of our method on a 51200-state randomized heterogeneous power network model, and show that the system level approach provides superior scalability over the centralized and distributed method. For such a large-scale example, the theoretical computation time for the centralized scheme is more than 200 days, and the distributed optimal control scheme is intractable. In contrast, it only takes 38 minutes to synthesize a localized optimal controller that achieves at least 99% global optimality guarantee.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Yuh-Shyang Wang. “Localized LQR with Adaptive Constraint and Performance Guarantee”. In: *to appear in 2016 55th IEEE Conference on Decision and Control (CDC)*. 2016.
Y.-S. Wang proposed the original idea of the paper. The content of this paper is described in Section 5.5 in this dissertation.

[2] Yuh-Shyang Wang and Nikolai Matni. “Localized Distributed Optimal Control with Output Feedback and Communication Delays”. In: *IEEE 52nd Annual Allerton Conference on Communication, Control, and Computing*. 2014. doi: `10.1109/ALLERTON.2014.7028511`.
Y.-S. Wang proposed the original idea of the paper. This paper is the preliminary version of [3].

[3] Yuh-Shyang Wang and Nikolai Matni. “Localized LQG Optimal Control for Large-Scale Systems”. In: *2016 IEEE American Control Conference (ACC)*. 2016. doi: `10.1109/ACC.2016.7525205`.
Y.-S. Wang proposed the original idea of the paper. The content of this paper is described in Chapter 6 in this dissertation.

[4] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. “A System Level Approach to Controller Synthesis”. In: *submitted to IEEE Transactions on Automatic Control* (2016).
Y.-S. Wang proposed the original idea of the paper. The content of this paper is covered in Chapter 1 to Chapter 4 in this dissertation.

[5] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. “Localized LQR Control with Actuator Regularization”. In: *2016 IEEE American Control Conference (ACC)*. 2016. doi: `10.1109/ACC.2016.7526485`.
Y.-S. Wang proposed the original idea of the paper. The content of this paper is described in Section 5.3 in this dissertation.

[6] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. “Localized LQR Optimal Control”. In: *2014 53rd IEEE Conference on Decision and Control (CDC)*. 2014. doi: `10.1109/CDC.2014.7039638`.
Y.-S. Wang proposed the original idea of the paper. The content of this paper is described in Chapter 5 in this dissertation.

[7] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. “Localized System Level Synthesis for Large-Scale Systems”. In: *submitted to IEEE Transactions on Automatic Control* (2016).
Y.-S. Wang proposed the original idea of the paper. The content of this paper is covered in Chapter 5 to Chapter 7 in this dissertation.

[8] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. “System Level Parameterizations, Constraints and Synthesis”. In: *submitted to 2017 IEEE American*

Control Conference (ACC). 2017.
Y.-S. Wang proposed the original idea of the paper. This paper is the simplified version of [4].

[9] Yuh-Shyang Wang, Seungil You, and Nikolai Matni. “Localized Distributed Kalman Filters for Large-Scale Systems”. In: *5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*. 2015. DOI: 10.1016/j.ifacol.2015.10.306.
Y.-S. Wang proposed the original idea of the paper. The content of this paper is described in Section 5.6 in this dissertation.

[10] Yuh-Shyang Wang et al. “Localized Distributed State Feedback Control with Communication Delays”. In: *2014 IEEE American Control Conference (ACC)*. June 2014. DOI: 10.1109/ACC.2014.6859440.
Y.-S. Wang proposed the original idea of the paper. This paper is the preliminary version of [6].

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

Large-scale networked systems have emerged in extremely diverse application areas recently, with examples including the smart grid, automated highway systems, software-defined networks, Internet of Things, and biological networks in science and medicine. These systems often have limited, sparse, uncertain, and distributed communication and computing in addition to sensing and actuation. Fortunately, the corresponding plants and performance requirements are also sparse and structured, and this must be exploited to make constrained controller design feasible, tractable, and scalable. In this dissertation, we introduce a new “system level" (SL) approach involving three complementary SL elements. System Level Parameterizations (SLPs) generalize state space and Youla parameterizations of all stabilizing controllers and the responses they achieve, and combine with System Level Constraints (SLCs) to parameterize the largest known class of constrained stabilizing controllers that admit a convex characterization, generalizing quadratic invariance (QI). The resulting System Level Synthesis (SLS) problems that arise define the broadest known class of constrained optimal control problems that can be solved using convex programming. Furthermore, we identify a class of SLS problem, which is called the convex localized separable SLS (CLS-SLS) problems, that can be solved with $O(1)$ computational complexity. The class of CLS-SLS problems, which include the localized $\mathcal{H}_2$ optimal control with sensor actuator regularization and the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control problem as special cases, are therefore scalable to systems with arbitrary large-scale. In the following, we review the literature in the field of constrained optimal control, present the key ideas and main contributions of our method, and outline the organization of the rest of this dissertation.

1.1 Motivation and Challenges

The Youla parameterization [77] represented an important shift towards a system level approach to optimal controller synthesis. Youla showed that there exists an isomorphism between a stabilizing controller and the resulting closed loop system response from sensors to actuators; therefore rather than synthesizing the controller itself, this system response (or Youla parameter) could be designed directly. The advantage of this approach is that an affine expression of the Youla parameter describes all achievable responses of the closed loop system, allowing for system behavior to be directly optimized. Together with state-space methods, this contribution played a major role in shifting controller synthesis from an ad hoc, loop-at-a-time tuning process to a principled one with well defined notions of optimality. Indeed, this approach proved very powerful, and paved the way for the foundational results of robust and optimal control that would follow [13].

This dissertation presents an approach that is inspired by the system level thinking pioneered by Youla: rather than directly designing only the feedback loop between sensors and actuators, we propose directly designing the *entire closed loop response of the system*, as captured by the maps from process and measurement disturbances to control actions and states — as such, we call the proposed method a system level approach to controller synthesis. A distinction between our approach and Youla's is that we explicitly model the internal delay structure of the feedback system, whereas Youla (and contemporary state-space methods) hid the internal structure of the controller, and focused instead on its input-output behavior. This focus on controller input-output behavior was natural for the problems of that era (often motivated by aerospace and process control applications), where systems had a single logically centralized controller with global access to sensor measurements and global control over actuators.

In contrast, modern cyber-physical systems (CPS) are large-scale, physically distributed, and interconnected. Rather than a logically centralized controller, these systems are composed of several sub-controllers, each equipped with their own sensors and actuators — these sub-controllers then exchange locally available information (such as sensor measurements or applied control actions) via a communication network. It follows that the information exchanged between sub-controllers is constrained by the delay, bandwidth, and reliability properties of this communication network, ultimately manifesting as information asymmetry among sub-controllers of the system. It is this information asymmetry, as imposed by the underlying communication network, that lies at the heart of what makes distributed optimal controller synthesis challenging [2, 3, 23, 37, 46, 53].

A defining feature of CPS is that controllers have internal delays, as specified by the exchange of information between constituent sub-controllers. These delays thus needed to be reintroduced into Youla and state-space based synthesis methods; methods that aimed to hide the internals of the controller from the system engineer.

Further, there was reason to suspect that introducing such information asymmetry into the optimal control problem lead to intractable synthesis tasks [4, 61, 76].

Despite these apparent technical and conceptual challenges, a body of work [2, 3, 14, 37, 46, 49, 53] that began in the early 2000s, and culminated with the introduction of quadratic invariance (QI) in the seminal paper [53], showed that for a large class of practically relevant systems, such internal structure could be incorporated into the Youla parameterization and still preserve the convexity of the optimal controller synthesis task. Informally, a system is quadratically invariant if sub-controllers are able to exchange information with each other faster than their control actions propagate through the CPS [52]. Even more remarkable is that this condition is tight, in the sense that QI is a necessary [33] and sufficient [53] condition for subspace constraints (defined by, for example, communication delays) on the controller to be enforceable via convex constraints on the Youla parameter.

The identification of QI as a useful condition for determining the tractability of a distributed optimal control problem led to an explosion of synthesis results in this area [27, 29, 30, 32, 34, 42, 55, 57, 59]. These results showed that the robust and optimal control methods that proved so powerful for centralized systems could be ported to distributed settings. However, they also made clear that the synthesis and implementation of QI distributed optimal controllers did not scale gracefully with the size of the underlying CPS. In particular, a QI distributed optimal controller is at least as expensive to compute as its centralized counterpart (c.f., the solutions presented in [27, 29, 30, 32, 34, 42, 55, 57, 59]), and can be more difficult to implement (c.f., the message passing implementation suggested in [29]).

We show in Chapter 2 that the QI framework, which adapts the Youla parameterization to a distributed setting, fails to capture certain constraints that are needed for optimal controller synthesis to scale to arbitrarily large systems. In particular, when the underlying physical system is strongly connected,[1] the QI framework does not allow for localized controllers, in which local sub-controllers only access a subset of system-wide measurements (c.f., Section 4.2.8), to be synthesized using convex programming; perhaps counter-intuitively, this statement holds true even when sub-controllers can exchange information with no delay (c.f., Example 1). Although this may seem surprising, note that implicit to the Youla parameterization is that sub-controllers can only exchange locally collected measurements with each other,

[1] We say that a plant is strongly connected if the state of any subsystem can eventually alter the state of all other subsystems.

and not, for instance, locally applied control actions. This restriction has no consequences in centralized applications, but can complicate controller implementation and synthesis in a distributed setting.

The lack of scalability of the distributed optimal control framework has not gone unnoticed by the community, and techniques based on regularization [17, 35], convex approximation [16, 19, 65], spatial truncation [45], and structural realization [62–64] have been used in hopes of finding a sparse (structured) feedback controller that is scalable to implement. These methods have been successful in extending the size of systems for which a distributed controller can be implemented, but there is still a limit to their scalability as they often rely on an underlying centralized synthesis procedure. Further, it is not clear if these methods can be extended to compute a dynamic controller that incorporates information sharing constraints.

To overcome the above-mentioned limitation, we propose the system level approach to controller synthesis [69, 72, 73], which is composed of three main elements: SLPs, SLCs, and SLS problems. We informally introduce the ideas of the system level approach as below.

1.2 Key Ideas

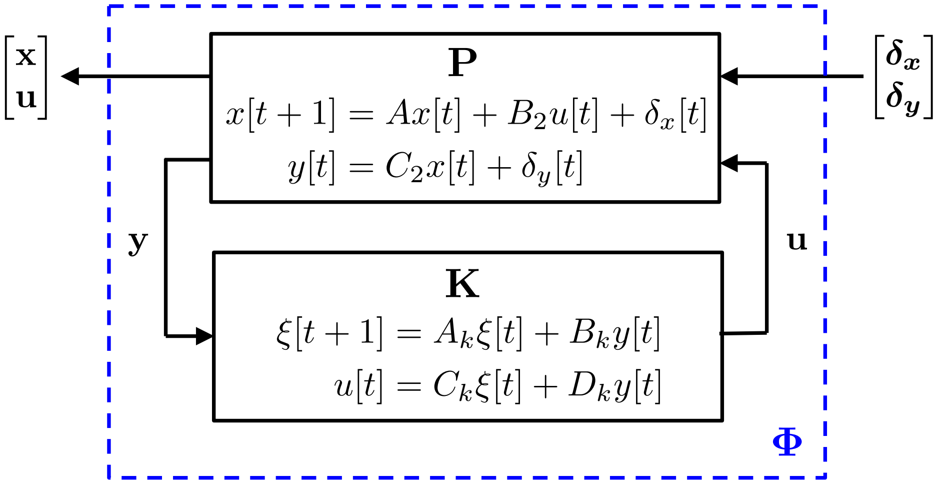

$\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix}$
$\begin{bmatrix}\boldsymbol{\delta_x}\\ \boldsymbol{\delta_y}\end{bmatrix}$
P
$x[t+1] = Ax[t] + B_2u[t] + \delta_x[t]$
$y[t] = C_2x[t] + \delta_y[t]$
y
u
K
$\xi[t+1] = A_k\xi[t] + B_ky[t]$
$u[t] = C_k\xi[t] + D_ky[t]$
$\boldsymbol{\Phi}$

Figure 1.1: Optimal Feedback Control Problem

Consider the optimal control problem shown in Figure 1.1, where **P** is called a plant and **K** is called a controller. Specifically, **P** is a linear time invariant (LTI) dynamical system that we want to control, and **K** is another LTI dynamical system that we want to design for. The interface between the plant and the controller are the measurement signal **y** and control action **u**. Let **Φ** be the system response from the external disturbance $\boldsymbol{\delta}_x$ and $\boldsymbol{\delta}_y$ to regulated output $\mathbf{x}$ and $\mathbf{u}$ in the closed loop. The traditional way to formulate a constrained optimal control problem is typically given by

$$\begin{aligned} \underset{\mathbf{K}}{\text{minimize}} \quad & ||\mathbf{\Phi}|| \\ \text{subject to} \quad & \mathbf{K} \text{ internally stabilizes } \mathbf{P} \\ & \mathbf{K} \in \mathcal{C}, \end{aligned} \tag{1.1}$$

where $\mathcal{C}$ is a structured constraint imposed on the controller. The idea of (1.1) is to find a controller $\mathbf{K}$ to optimize the closed loop system response $\mathbf{\Phi}$, subject to the constraints that the interconnected feedback loop shown in Figure 1.1 is internally stable and the controller satisfies the structured constraint $\mathcal{C}$. The limitation of the formulation (1.1) is that the structured constraint $\mathcal{C}$ usually makes the constrained optimal control problem non-convex, i.e., the QI framework only characterizes a limited class of convex problems in constrained optimal control.

The system level design philosophy approaches the constrained optimal control problem from a different point of view. Instead of designing a controller $\mathbf{K}$ to optimize the system response $\mathbf{\Phi}$, we directly choose our desired system response $\mathbf{\Phi}$ from the set of all stable and achievable system response. After the desired system response $\mathbf{\Phi}$ is chosen, we then reconstruct a controller $\mathbf{K}$ to achieve the desired system response. This design philosophy leads to a system level synthesis (SLS) problem given by

$$\begin{aligned} \underset{\mathbf{\Phi}}{\text{minimize}} \quad & g(\mathbf{\Phi}) & \text{(1.2a)} \\ \text{subject to} \quad & \mathbf{\Phi} \text{ stable and achievable} & \text{(1.2b)} \\ & \mathbf{\Phi} \in \mathcal{S}, & \text{(1.2c)} \end{aligned}$$

where (1.2a) is called the system level objective (SLO), (1.2b) the system level parameterization (SLP), and (1.2c) the system level constraint (SLC). The goal of this dissertation is to show that the SLS framework (1.2) offers significant benefits over the traditional frameowork (1.1) in terms of the generality, simplicity, and scalability. Our specific contributions are outlined below.

1.3 Theoretical Contributions

First, we generalize the objectives of the constrained optimal control problem (1.1) from a system norm to arbitrary SLO (1.2a). This allows us to extend a controller design problem to a system level co-design problem. For instance, we can incorporate the regularizers for sensor actuator placement into the SLO in (1.2a), and use the SLS problem to co-design the controller and its sensing and actuating architecture. The set of all SLS problems is a superset of the set of all constrained optimal control problems. We show this relation using the Venn diagram in Figure 1.2.

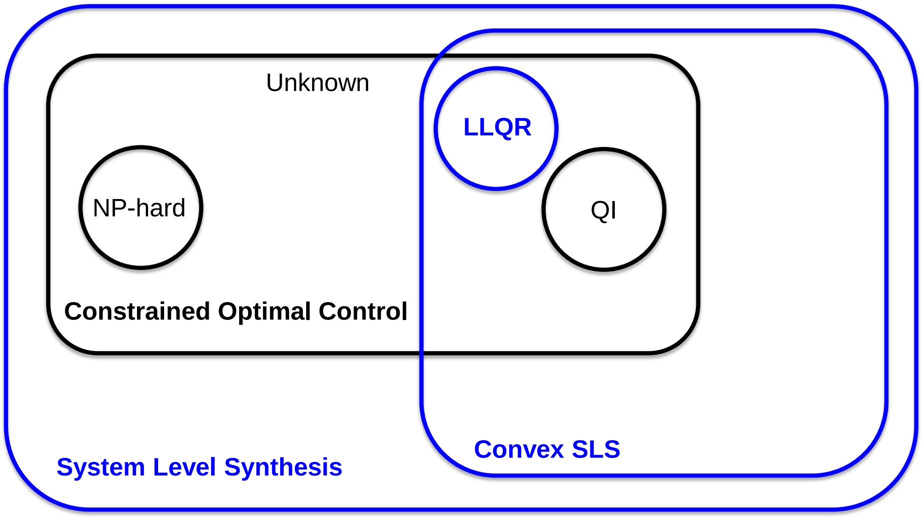


Figure 1.2: Relations between the traditional constrained optimal control framework and the system level synthesis framework

Then, we show that the SLS framework characterizes the broadest known class of constrained optimal control problems that can be solved using convex programming. This is one of the main theoretical contribution of the SLS framework. In particular, we show that the set of constrained stabilizing controllers that can be efficiently parameterized using SLPs (1.2b) and SLCs (1.2c) is a strict superset of those that can be parameterized using quadratic invariance with (1.1), and hence we provide a generalization of the QI framework, characterizing the broadest known class of constrained controllers that admit a convex parameterization. The relation between convex SLS and QI are shown in Figure 1.2.

Furthermore, we identify a class of SLS problems, which we call the CLS-SLS problems, that are scalable to systems with arbitrary large-scale. Specifically, we propose a distributed algorithm to solve all CLS-SLS problems in a localized and scalable way, with $O(1)$ parallel computational complexity. As a concrete example, we demonstrate the localized linear quadratic regulator (LLQR) algorithm on a randomized heterogeneous power network example with 51200 states, and show that the LLQR controller can be computed in 23 minutes using a personal computer.

In contrast, the theoretical computation time for the traditional LQR using the same computer is 200 days, and the distributed LQR is simply intractable. We also use an adaptive constraint update algorithm to design a LLQR controller with at least 99% optimality guarantee on the 51200-state example in 38 minutes. Table 5.2 in Chapter 5 shows the superior scalability of LLQR (a special case of a CLS-SLS problem) over the centralized and distributed approach. As shown in Figure 1.3, examples of CLS-SLS problems include localized linear quadratic Gaussian (LLQG) (the output feedback version of LLQR), localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control, and LLQG with sensor actuator regularization using the regularization for design (RFD) framework [40, 41].

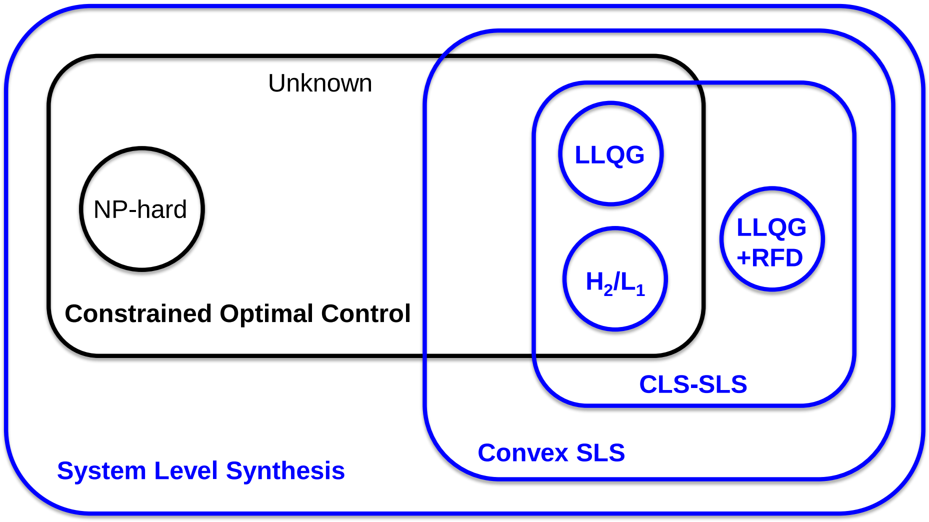


Figure 1.3: Examples of CLS-SLS problems

Finally, the SLS formulation provides a unified framework to simultaneously incorporate user-specified design specifications on the closed loop and structured constraints imposed on the controller into the optimal controller design process — both the design specification and structured constraints are in the form of SLCs. In Chapter 4, we provide a catalog of SLCs that admit a convex representation: highlights include general convex constraints on the Youla parameter (QI subspace constraints being a special case thereof), robustness and architectural constraints on the controller, as well as multi-objective performance constraints and spatiotemporal constraints on the system response. We also show that the constrained system responses can be used to directly implement a controller achieving them — in particular, any SLC imposed on the system response imposes a corresponding SLC on the internal structure of the resulting controller. This offers a convex way to design a structured controller that is scalable to implement for large-scale systems, which cannot be done using the traditional optimal control framework (1.1).

1.4 Organization of the Dissertation

The rest of this dissertation is structured as follows. In Chapter 2, we define the system model considered in the thesis, and review relevant results from the distributed optimal control and QI literature. We then provide a motivating example as to why moving beyond QI systems may be desirable, before presenting a survey of our main results. In Chapter 3 we define and analyze SLPs (1.2b) for state and output feedback problems, and provide a characterization of stable and achievable system responses. We show that SLPs also give a characterization of all internally stabilizing controllers, and propose a structural realization of the controller to achieve the desired system response. In Chapter 4, we define and analyze the SLS problem (1.2), which incorporates SLPs and SLCs into an optimization problem. We provide a catalog of SLCs that can be imposed on the system responses parameterized by the SLPs described in the previous chapter — in particular, we show that by appropriately selecting these SLCs, we can provide convex characterizations of all stabilizing controllers satisfying QI subspace constraints, convex constraints on the Youla parameter, finite impulse response (FIR) constraints, sparsity constraints, spatiotemporal constraints [67, 68, 71, 75], controller internal robustness constraints, multi-objective performance constraints, controller architecture constraints [40, 41, 70], and any combination thereof. In addition, we show that the constrained optimal control problem (1.1) is a special case of SLS (1.2).

In Chapters 5 - 7, we move our focus to the class of CLS-SLS problems for large-scale systems. We introduce LLQR control in Chapter 5, and show that the synthesis and implementation of a LLQR controller are scalable to systems with arbitrary large-scale, i.e., with $O(1)$ parallel computational complexity. We demonstrate the LLQR algorithm on a randomized heterogeneous power network example with 51200 states, and show that the LLQR controller achieves superior scalability over the centralized and distributed methods. In Chapter 6, we generalize LLQR to output feedback localized linear quadratic Gaussian (LLQG). We combine distributed optimization algorithm such as alternating direction method of multipliers (ADMM) with LLQR to solve a LLQG in a localized yet iterative manner. In Chapter 7, we generalize LLQG to the class of CLS-SLS problems, and propose a distributed optimization algorithm to solve all CLS-SLS problems in a localized and scalable way, with $O(1)$ parallel computational complexity. Finally, we end with conclusions and future works in Chapter 8.

1.5 Mathematical Notations

We use lower and upper case Latin letters such as x and A to denote vectors and matrices, respectively, and lower and upper case boldface Latin letters such as $\mathbf{x}$ and $\mathbf{G}$ to denote signals and transfer matrices, respectively. We use calligraphic letters such as $\mathcal{S}$ to denote sets.

In the interest of clarity, we work with discrete time linear time invariant systems, but unless stated otherwise, all results extend naturally to the continuous time setting. We use standard definitions of the Hardy spaces $\mathcal{H}_2$ and $\mathcal{H}_\infty$, and denote their restriction to the set of real-rational proper transfer matrices by $\mathcal{RH}_2$ and $\mathcal{RH}_\infty$. We use $G[i]$ to denote the ith spectral component of a transfer function $\mathbf{G}$, i.e., $\mathbf{G}(z) = \sum_{i=0}^{\infty} \frac{1}{z^i} G[i]$ for $|z| > 1$. Finally, we use $\mathcal{F}_T$ to denote the space of finite impulse response (FIR) transfer matrices with horizon T, i.e., $\mathcal{F}_T := \{\mathbf{G} \in \mathcal{RH}_\infty \,|\, \mathbf{G} = \sum_{i=0}^{T} \frac{1}{z^i} G[i]\}$.

Chapter 2

PROBLEM STATEMENT AND MAIN RESULTS

We begin by introducing the system model and some preliminaries on optimal control and Youla parameterization. We then introduce the distributed optimal control framework and discuss its limitations. This chapter ends with the summary of the main result of this dissertation.

2.1 System Model

We consider discrete time linear time invariant (LTI) systems of the form

$$x[t+1] = Ax[t] + B_1 w[t] + B_2 u[t] \tag{2.1a}$$

$$\bar{z}[t] = C_1 x[t] + D_{11} w[t] + D_{12} u[t] \tag{2.1b}$$

$$y[t] = C_2 x[t] + D_{21} w[t] + D_{22} u[t], \tag{2.1c}$$

where x, u, w, y, and $\bar{z}$ are the state vector, control action, external disturbance, measurement, and regulated output, respectively. The frequency domain representation of (2.1) is given by

$$\begin{aligned} z\mathbf{x} &= A\mathbf{x} + B_1\mathbf{w} + B_2\mathbf{u} \\ \bar{\mathbf{z}} &= C_1\mathbf{x} + D_{11}\mathbf{w} + D_{12}\mathbf{u} \\ \mathbf{y} &= C_2\mathbf{x} + D_{21}\mathbf{w} + D_{22}\mathbf{u}, \end{aligned} \tag{2.2}$$

where z is the variable of z-transform. Equation (2.1) can be written in state space form as

$$\mathbf{P} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}, \tag{2.3}$$

where $\mathbf{P}_{ij} = C_i(zI - A)^{-1}B_j + D_{ij}$. We refer to $\mathbf{P}$ as the open loop plant model.

Remark 1. *We will occasionally discuss continuous time system in this dissertation. A continuous time LTI system is given in the form*

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 u(t) \\ \bar{z}(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t) \\ y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t). \end{aligned} \tag{2.4}$$

The frequency domain representation of the continuous time system is given by (2.2) *by changing z to s (the variable for Laplace transform). The state space representation is given by* (2.3)*, with* $\mathbf{P}_{ij} = C_i(sI - A)^{-1}B_j + D_{ij}$.

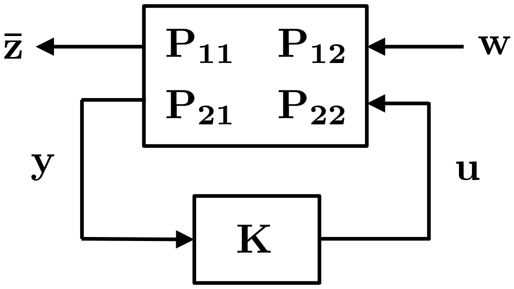


Figure 2.1: Interconnection of the plant **P** and controller **K**.

Consider a dynamic output feedback control law $\mathbf{u} = \mathbf{K}\mathbf{y}$. The controller **K** is assumed to have the state space realization

$$\xi[t+1] = A_k\xi[t] + B_k y[t] \tag{2.5a}$$

$$u[t] = C_k\xi[t] + D_k y[t], \tag{2.5b}$$

where ξ is the internal state of the controller. We have $\mathbf{K} = C_k(zI - A_k)^{-1}B_k + D_k$. A schematic diagram of the interconnection of the plant **P** and the controller **K** is shown in Figure 2.1.

The following assumptions are made throughout the dissertation.

Assumption 1. *The interconnection in Figure 2.1 is well-posed — the matrix* $(I - D_{22}D_k)$ *is invertible.*

Assumption 2. *Both the plant and the controller realizations are stabilizable and detectable; i.e.,* (A, B_2) *and* (A_k, B_k) *are stabilizable, and* (A, C_2) *and* (A_k, C_k) *are detectable.*

The internal stability of the interconnection in Figure 2.1 is defined as follows [81].

Definition 1. *The interconnection in Figure 2.1 is said to be internally stable if the origin* $(x, \xi) = (0, 0)$ *is asymptotically stable, i.e.,* $x[t], \xi[t] \to 0$ *for* $t \to \infty$ *from all initial states.*

We say that a controller **K** is an internally stabilizing controller for plant **P** if the interconnection of **P** and **K** in Figure 2.1 is internally stable. We summarize some useful lemmas to determine the internal stability of the interconnection in Figure 2.1 in Appendix 2.A in the end of this chapter.

The aim of the optimal control problem is to find a controller $\mathbf{K}$ to stabilize the plant $\mathbf{P}$ and minimize a suitably chosen norm[1] of the closed loop transfer matrix from external disturbance $\mathbf{w}$ to regulated output $\bar{\mathbf{z}}$. From the relations $\bar{\mathbf{z}} = \mathbf{P}_{11}\mathbf{w} + \mathbf{P}_{12}\mathbf{u}$, $\mathbf{y} = \mathbf{P}_{21}\mathbf{w} + \mathbf{P}_{22}\mathbf{u}$, and $\mathbf{u} = \mathbf{K}\,\mathbf{y}$, we can express $\bar{\mathbf{z}}$ as a function of $\mathbf{w}$ as

$$\bar{\mathbf{z}} = (\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21})\mathbf{w}.$$

The optimal control problem can then be formulated as

$$\begin{aligned} \underset{\mathbf{K}}{\text{minimize}} \quad & ||\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}|| \\ \text{subject to} \quad & \mathbf{K} \text{ internally stabilizes } \mathbf{P}. \end{aligned} \tag{2.6}$$

We refer to (2.6) the unconstrained (centralized) optimal control problem.

2.2 Youla Parameterization

Youla parameterization is a common technique to characterize the set of all internally stabilizing controller for a given plant, i.e., the constraint set in (2.6). The Youla parameterization technique is based on a doubly co-prime factorization of the plant, which is defined as follows.

Definition 2. *A collection of stable transfer matrices,* $\mathbf{U}_r$*,* $\mathbf{V}_r$*,* $\mathbf{X}_r$*,* $\mathbf{Y}_r$*,* $\mathbf{U}_l$*,* $\mathbf{V}_l$*,* $\mathbf{X}_l$*,* $\mathbf{Y}_l \in \mathcal{RH}_\infty$ *defines a doubly co-prime factorization of* $\mathbf{P}_{22}$ *if* $\mathbf{P}_{22} = \mathbf{V}_r\mathbf{U}_r^{-1} = \mathbf{U}_l^{-1}\mathbf{V}_l$ *and*

$$\begin{bmatrix} \mathbf{X}_l & -\mathbf{Y}_l \\ -\mathbf{V}_l & \mathbf{U}_l \end{bmatrix} \begin{bmatrix} \mathbf{U}_r & \mathbf{Y}_r \\ \mathbf{V}_r & \mathbf{X}_r \end{bmatrix} = I.$$

Such doubly co-prime factorizations can always be computed if $\mathbf{P}_{22}$ is stabilizable and detectable [81].

Let $\mathbf{Q}$ be the Youla parameter. From [81], the centralized optimal control problem (2.6) can be reformulated in terms of the Youla parameter as

$$\begin{aligned} \underset{\mathbf{Q}}{\text{minimize}} \quad & ||\mathbf{T}_{11} + \mathbf{T}_{12}\mathbf{Q}\mathbf{T}_{21}|| \\ \text{subject to} \quad & \mathbf{Q} \in \mathcal{RH}_\infty \end{aligned} \tag{2.7}$$

with $\mathbf{T}_{11} = \mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{Y}_r\mathbf{U}_l\mathbf{P}_{21}$, $\mathbf{T}_{12} = -\mathbf{P}_{12}\mathbf{U}_r$, and $\mathbf{T}_{21} = \mathbf{U}_l\mathbf{P}_{21}$. In (2.7), we search over all stable proper real-rational transfer matrix $\mathbf{Q}$ to minimize the norm of the

[1] Typical choices for the norm include $\mathcal{H}_2$ and $\mathcal{H}_\infty$.

closed loop map. Once the optimal Youla parameter $\mathbf{Q}$ is found, we reconstruct the controller $\mathbf{K}$ by the formula

$$\mathbf{K} = (\mathbf{Y}_r - \mathbf{U}_r\mathbf{Q})(\mathbf{X}_r - \mathbf{V}_r\mathbf{Q})^{-1}.$$

The set $\{(\mathbf{Y}_r-\mathbf{U}_r\mathbf{Q})(\mathbf{X}_r-\mathbf{V}_r\mathbf{Q})^{-1}|\mathbf{Q} \in \mathcal{RH}_\infty\}$ is the parameterization of all internally stabilizing controller.

With a change of variable from controller $\mathbf{K}$ to Youla parameter $\mathbf{Q}$, we note that (2.7) is in the form of a convex optimization problem, which can then be solved using efficient convex programming algorithms.

2.3 Distributed Optimal Control and Quadratic Invariance

Distributed optimal control problems arise when there are information asymmetry among sub-controllers in the network. In this section, we follow the paradigm adopted in [27, 30, 32, 34, 42, 53, 55, 57, 59], and focus on information asymmetry introduced by delays in the communication network — this is a reasonable modeling assumption when one has dedicated physical communication channels (e.g., fiber optic channels), but may not be valid under wireless settings. In the references cited above, locally acquired measurements are exchanged between sub-controllers subject to delays imposed by the communication network,[2] which manifest as subspace constraints on the controller itself.

The distributed optimal control problem is then formulated as a constrained optimal control problem in the following form [31, 33, 53, 54]:

$$\begin{aligned} \underset{\mathbf{K}}{\text{minimize}} \quad & \|\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\| && \text{(2.8a)} \\ \text{subject to} \quad & \mathbf{K} \text{ internally stabilizes } \mathbf{P} && \text{(2.8b)} \\ & \mathbf{K} \in \mathcal{C}, && \text{(2.8c)} \end{aligned}$$

for $\mathcal{C}$ a subspace. This subspace can enforce, for instance, the information sharing constraints imposed on the controller $\mathbf{K}$ by the underlying communication network, as described above.

In order to determine the tractability (convexity) of the constrained optimal control problem (2.8), we first reformulate (2.8) in terms of the Youla parameter $\mathbf{Q}$. The

[2] Note that this delay may range from 0, modeling instantaneous communication between sub-controllers, to infinite, modeling no communication between sub-controllers.

authors in [54] show that (2.8) can be equivalently formulated as

$$\begin{aligned}\underset{\mathbf{Q}}{\text{minimize}} \quad & \|\mathbf{T}_{11} + \mathbf{T}_{12}\mathbf{Q}\mathbf{T}_{21}\| && \text{(2.9a)}\\ \text{subject to} \quad & \mathbf{Q} \in \mathcal{RH}_\infty && \text{(2.9b)}\\ & \mathfrak{M}(\mathbf{Q}) \in C_Q, && \text{(2.9c)}\end{aligned}$$

where $\mathfrak{M}$ is an invertible affine map defined in terms of an arbitrary doubly co-prime factorization of the plant[3], and C_Q is given by the set $C_Q = \{\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1} | \mathbf{K} \in C\}$. Note that the convexity of (2.9) (and/or (2.8)) depends solely on the convexity of the set C_Q.

A synthesis of the main results of the distributed optimal control literature [27, 30–34, 42, 53–55, 57, 59] can be expressed as follows: if the subspace C is quadratically invariant (QI)[4] with respect to $\mathbf{P}_{22}$ [53], then we have $C_Q = C$, and therefore the constraint (2.9c) can be replaced by the subspace constraint $\mathfrak{M}(\mathbf{Q}) \in C$. In this case, problem (2.9) is a convex optimization problem. Further, the QI condition can be viewed as tight, in the sense that quadratic invariance is also a *necessary* condition [31, 33] for a subspace constraint C on the controller $\mathbf{K}$ to be enforced on the Youla parameter $\mathbf{Q}$ in a convex manner.

2.4 Beyond Quadratic Invariance

We now present a simple example showing how the above framework, built around the Youla parameterization, fails to capture an “obvious” structured controller. We return to this example at the end of this chapter to show that our system level approach naturally recovers said obvious controller.

Example 1. *Consider the optimal control problem:*

$$\begin{aligned}\underset{u}{\textit{minimize}} \quad & \lim_{T\to\infty} \tfrac{1}{T} \textstyle\sum_{t=0}^{T} \mathbb{E}\|x[t]\|_2^2 \\ \textit{subject to} \quad & x[t+1] = Ax[t] + u[t] + w[t],\end{aligned} \qquad (2.10)$$

with zero mean unit covariance additive white Gaussian noise (AWGN) vector $w[t]$, i.e., $w[t] \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, I)$. We assume full state-feedback, i.e., the control action at time t can be expressed as $u[t] = f(x[0:t])$ for some function f. An optimal control policy $u^\star$ for this linear quadratic regulator (LQR) problem is easily seen to be given by $u^\star[t] = -Ax[t]$.

[3] We have $\mathfrak{M}(\mathbf{Q}) = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1} = (\mathbf{Y}_r - \mathbf{U}_r\mathbf{Q})\mathbf{U}_l$. By definition, we have $\mathbf{P}_{22} = \mathbf{V}_r\mathbf{U}_r^{-1} = \mathbf{U}_l^{-1}\mathbf{V}_l$. This implies that the transfer matrices $\mathbf{U}_r$ and $\mathbf{U}_l$ are both invertible. Therefore, $\mathfrak{M}$ is an invertible affine map of the Youla parameter $\mathbf{Q}$.

[4] The subspace C is quadratically invariant with respect to $\mathbf{P}_{22}$ if $\mathbf{K}\mathbf{P}_{22}\mathbf{K} \in C$ for all $\mathbf{K} \in C$

Further suppose that the state matrix A is sparse and let its support define the adjacency matrix of a graph $\mathcal{G}$ for which we identify the ith node with the corresponding state/control pair (x_i, u_i). In this case, we have that the optimal control policy $u^\star$ can be implemented in a localized *manner. In particular, in order to implement the state feedback policy for the ith actuator u_i, only those states x_j for which $A_{ij} \neq 0$ need to be collected — thus only those states corresponding to immediate neighbors of node i in the graph $\mathcal{G}$, i.e., only* local *states, need to be collected to compute the corresponding control action, leading to a localized implementation. As we discuss in more detail in Section 4.2.8 and in Chapter 5 - 7, the idea of locality is essential to allowing controller synthesis and implementation to scale to arbitrarily large systems, and hence such a structured controller is desirable.*

Now suppose that we naively attempt to solve optimal control problem (2.10) *by converting it to its equivalent $\mathcal{H}_2$ optimal control problem and constraining the controller $\mathbf{K}$ to have the same support as A, i.e., $\mathbf{K} = \sum_{t=0}^{\infty} \frac{1}{z^t} K[t]$, $\text{supp}\,(K[t]) \subset \text{supp}\,(A)$. If the graph $\mathcal{G}$ is strongly connected, then the conditions in [52] imply that the corresponding distributed optimal control problem is not quadratically invariant. The results of [33] further allow us to conclude that computing such a structured controller cannot be done using convex programming when using the Youla parameterization described in the previous section.*

In addition, we note that the QI framework is developed under the assumption that C in (2.8c) is a *subspace* constraint. When C is not a subspace constraint, no general method exists to determine the convexity of the constrained optimal control problem (2.8). Further, we note that the optimization problem (2.9) is convex as long as the set C_Q in (2.9c) is a convex set — in particular, the set C_Q does not need to satisfy the QI condition, nor does C_Q need to be a subspace. In other words, there are some convex constrained optimal control problems that cannot be identified using the theory of QI. The motivation of this thesis is to generalize the QI framework to characterize a broader class of convex constrained optimal control problems, and show that some of these problems are extremely favorable for large-scale applications.

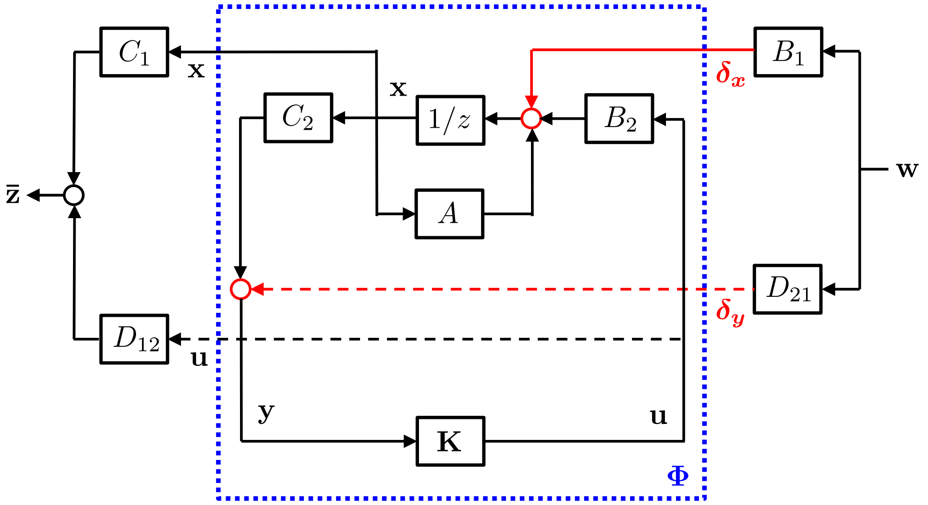
2.5 Summary of Main Results

The rest of the thesis is devoted to defining and analyzing the system level approach to controller synthesis, centered around the notion of a *system response*. We collect here a summary of our main results, and show how they can be used to pose a novel System Level Synthesis (SLS) problem that significantly generalizes that defined in equation (2.8).

For a LTI system with dynamics given by (2.1), we define a *system response* $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ to be the closed loop maps satisfying

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_x \\ \boldsymbol{\delta}_y \end{bmatrix}, \tag{2.11}$$

where $\boldsymbol{\delta}_x = B_1\mathbf{w}$ is the disturbance on the state vector, and $\boldsymbol{\delta}_y = D_{21}\mathbf{w}$ is the disturbance on the measurement. We illustrate the system response using the block diagram shown in Figure 2.2.



$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_x \\ \boldsymbol{\delta}_y \end{bmatrix} \qquad \boldsymbol{\Phi} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix}$$

Figure 2.2: An illustration of the system response in the block diagram

We say that a system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ is *stable and achievable* with respect to a plant $\mathbf{P}$ if there exists an internally stabilizing controller $\mathbf{K}$ such that the interconnection illustrated in Figure 2.1 leads to closed loop behavior consistent with equation (2.11).

In Chapter 3, Theorem 2, we show that a system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ is stable and achievable with respect to a strictly proper plant $\mathbf{P}$ with realization (2.1) if and only if it lies in the affine subspace described by:

$$\begin{bmatrix} zI-A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{2.12}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} zI-A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{2.13}$$

$$\mathbf{R}, \mathbf{M}, \mathbf{N} \in \frac{1}{z}\mathcal{RH}_\infty, \quad \mathbf{L} \in \mathcal{RH}_\infty. \tag{2.14}$$

As the above characterizes all stable and achievable system responses, we call it a system level parameterization (SLP).

In addition, for such a stable achievable system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$, a controller that leads to these closed loop maps is given by $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$, and can be implemented as:

$$\begin{aligned} z\boldsymbol{\beta} &= z(I - z\mathbf{R})\boldsymbol{\beta} - z\mathbf{N}\mathbf{y} \\ \mathbf{u} &= z\mathbf{M}\boldsymbol{\beta} + \mathbf{L}\mathbf{y}, \end{aligned} \tag{2.15}$$

for $\boldsymbol{\beta}$ the internal state of the stabilizing controller.[5] A block diagram of the controller structure is shown in Figure 2.3.

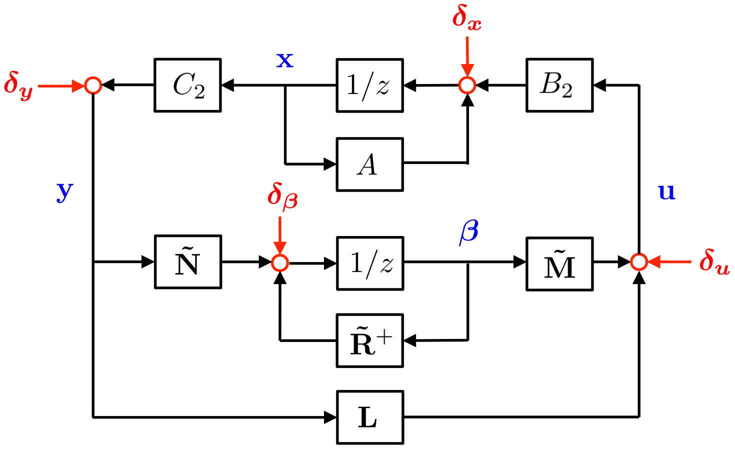


Figure 2.3: The proposed output feedback controller structure, with $\tilde{\mathbf{R}}^+ = z\tilde{\mathbf{R}} = z(I - z\mathbf{R})$, $\tilde{\mathbf{M}} = z\mathbf{M}$, and $\tilde{\mathbf{N}} = -z\mathbf{N}$.

Notice that any sparsity structure imposed on the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ translates directly to the sparsity structure of the controller implementation, and hence information sharing constraints on the measured output $\mathbf{y}$ and controller state $\boldsymbol{\beta}$ can be imposed via subspace constraints on the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$. As the

[5] Although not apparent, $z(I - z\mathbf{R}) \in \mathcal{RH}_\infty$, and hence the suggested controller implementation is causal. See Section 3.2 for further details.

controller is implemented directly using these transfer matrices, we are furthermore no longer limited to subspace constraints, and can in fact impose arbitrary system level constraints (SLCs) on the closed loop response of the system, and by extension the controller implementation. In Section 4.2, we provide a catalog of useful SLCs. We also show in Section 4.2.1 and 4.2.2 that by combining appropriate SLCs with the SLP (2.12) - (2.14), we recover all structured controllers that can be parameterized using the Youla parameter and quadratic invariance.

Let $\mathcal{S}$ denote such a SLC, and assume that it admits a convex representation. Furthermore, let $g(\cdot)$ be a convex functional. This gives a convex system level synthesis (SLS) problem

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & g(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}) && (2.16a) \\ \text{subject to} \quad & \text{equations } (2.12) - (2.14) && (2.16b) \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{S}. && (2.16c) \end{aligned}$$

We show in Chapter 4 that the SLS problem (2.16) characterize the broadest known class of convex problems in optimal control — in particular, the distributed optimal control problem (2.8) is a special case of a SLS problem. Beside (2.8), we show that the mixed objective optimal control problem and the sensor actuator regularized optimal control problem can all be cast as a SLS problem.

In Chapters 5 - 7, we focus on a special class of SLS problems, which we call the convex localized separable SLS (CLS-SLS) problems. By imposing suitable localized SLC in (2.16c), we show that the CLS-SLS problems can be solved in an extremely scalable manner, i.e., with $O(1)$ parallel computational and implementation complexity relative to the size of the overall system. This allows us to synthesize and implement localized optimal controller for systems with arbitrary large-scale, which is extremely favorable for large-scale applications.

Example 2 (Example 1 cont'd)**.** *We now return to the motivating example introduced above to provide a preview of the usefulness of the system level approach to controller synthesis. In the case of a full control (*$B_2 = I$*) state-feedback (*$C_2 = I$*,* $D_{21} = 0$*) problem, the conditions* (2.12) - (2.14) *simplify to* $(zI - A)\mathbf{R} - \mathbf{M} = I$*,* $\mathbf{R}, \mathbf{M} \in \frac{1}{z}\mathcal{RH}_\infty$ *(c.f. Section 3.1), and a controller achieving the desired response is given by* $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$*. Further, this controller can be implemented as*

$$\hat{\mathbf{w}} = \mathbf{x} - \hat{\mathbf{x}}, \ \mathbf{u} = z\mathbf{M}\hat{\mathbf{w}}, \ \hat{\mathbf{x}} = (z\mathbf{R} - I)\hat{\mathbf{w}}.$$

Again, suppose that we wish to synthesize an optimal controller that has a communication topology given by the support of A — from the above implementation, it suffices to constrain the support of transfer matrices $\mathbf{R}$ *and* $\mathbf{M}$ *to be a subset of that of A. It can be checked that* $\mathbf{R} = \frac{1}{z}I$*, and* $\mathbf{M} = -\frac{1}{z}A$ *satisfy the above constraints, and recover the globally optimal controller* $\mathbf{K} = -A$*. Recall that this controller cannot be computed using quadratic invariance and the Youla parameterization if the graph with adjacency matrix defined by the support of A is strongly connected and sparse.*

APPENDIX

2.A Lemmas for Internal Stability

The following definitions and lemmas are useful to determine the internal stability of the closed loop system for a given controller [81].

Definition 3 (stable matrix)**.** *For the discrete time system* (2.1)*, the square matrix A is said to be a* stable matrix *if the spectral radius of A is smaller than* 1*. For the continuous time system* (2.4)*, the square matrix A is said to be a* stable matrix *if every eigenvalue of A has strictly negative real part.*

Lemma 1. *The interconnection in Figure 2.1 is internally stable if and only if the matrix*

$$A_{cl} = \begin{bmatrix} A & 0 \\ 0 & A_k \end{bmatrix} + \begin{bmatrix} B_2 & 0 \\ 0 & B_k \end{bmatrix} \begin{bmatrix} I & -D_{22} \\ -D_k & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & C_k \\ C_2 & 0 \end{bmatrix}$$

is a stable matrix. In particular, when $D_{22} = 0$, the equation above can be simplified into

$$A_{cl} = \begin{bmatrix} A + B_2 D_k C_2 & B_2 C_k \\ B_k C_2 & A_k \end{bmatrix}. \tag{2.17}$$

Lemma 2. *The interconnection in Figure 2.1 is internally stable if and only if the four closed loop transfer matrices from $(\boldsymbol{\delta_y}, \boldsymbol{\delta_u})$ to $(\mathbf{u}, \mathbf{y})$ in Figure 2.4 are in $\mathcal{RH}_\infty$ (real rational, stable, and proper).*

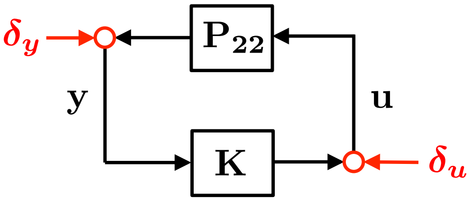


Figure 2.4: Internal stability analysis diagram

Chapter 3

SYSTEM LEVEL PARAMETERIZATION OF STABILIZING CONTROLLERS

In this chapter, we propose a system level approach to parameterize the set of all internally stabilizing controller. Specifically, we show that the affine subspace defined by the constraints (2.12) - (2.14) parameterizes all stable achievable system responses $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$. In addition, the controller $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$, which admits a realization as described in (2.15), parameterizes all internally stabilizing controllers for a strictly proper plant $\mathbf{P}_{22}$.[1] The results of this chapter provide an alternative to the traditional Youla parameterization, but is far more amenable for constrained optimal control problems, as will be shown in the next few chapters.

We begin by analyzing the state feedback case, as it admits a simpler characterization and allows us to provide intuition about the construction of a controller that achieves a desired system response. With this intuition in hand, we present our results for the output feedback setting, which is the main focus of this chapter.

3.1 State Feedback

We consider a state feedback problem with plant model given by

$$\mathbf{P} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ I & 0 & 0 \end{array}\right]. \tag{3.1}$$

The z-transform of the state dynamics (2.1a) is given by

$$(zI - A)\mathbf{x} = B_2\mathbf{u} + \boldsymbol{\delta_x}, \tag{3.2}$$

where we let $\boldsymbol{\delta_x} := B_1\mathbf{w}$ denote the disturbance affecting the state.

We define $\mathbf{R}$ to be the system response mapping the external disturbance $\boldsymbol{\delta_x}$ to the state $\mathbf{x}$, and $\mathbf{M}$ to be the system response mapping the disturbance $\boldsymbol{\delta_x}$ to the control action $\mathbf{u}$. By substituting a dynamic state feedback control rule $\mathbf{u} = \mathbf{K}\mathbf{x}$ into (3.2), we can write the system response $\{\mathbf{R}, \mathbf{M}\}$ as a function of the controller $\mathbf{K}$ as

$$\begin{aligned} \mathbf{R} &= (zI - A - B_2\mathbf{K})^{-1} \\ \mathbf{M} &= \mathbf{K}(zI - A - B_2\mathbf{K})^{-1}. \end{aligned} \tag{3.3}$$

[1] The non-strictly proper case is discussed in Section 3.3.

The main result of this section is an algebraic characterization of the set $\{\mathbf{R}, \mathbf{M}\}$ of state-feedback system responses that are achievable by an internally stabilizing controller $\mathbf{K}$, as stated in the following theorem.

Theorem 1 (System Level Parameterization for State Feedback Systems)**.** *For the state feedback system* (3.1)*, the following are true:*

(a) The affine subspace defined by

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \tag{3.4a}$$

$$\mathbf{R}, \mathbf{M} \in \frac{1}{z}\mathcal{RH}_\infty \tag{3.4b}$$

parameterizes all system responses from $\boldsymbol{\delta_x}$ to $(\mathbf{x}, \mathbf{u})$*, as defined in* (3.3)*, achievable by an internally stabilizing state feedback controller* $\mathbf{K}$*.*

(b) For any transfer matrices $\{\mathbf{R}, \mathbf{M}\}$ *satisfying* (3.4)*, the controller* $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$ *is internally stabilizing and achieves the desired system response* (3.3)*.*

The rest of this section is devoted to proving the claims made in Theorem 1.

3.1.1 Necessity

The necessity of a stable and achievable system response $\{\mathbf{R}, \mathbf{M}\}$ lying in the affine subspace (3.4) follows from rote calculation. Here we provide some intuition about the conditions (3.4). Note that (3.4a) can be derived by substituting the definition $\mathbf{x} = \mathbf{R}\boldsymbol{\delta_x}$ and $\mathbf{u} = \mathbf{M}\boldsymbol{\delta_x}$ into (3.2). This condition must hold for any achievable system response $(\mathbf{R}, \mathbf{M})$. In addition, for a proper controller $\mathbf{K}$, the relations in (3.3) imply that both $\mathbf{R}$ and $\mathbf{M}$ are strictly proper. Intuitively, the state feedback system (3.2), or the matrix pair (A, B_2), is stabilizable if and only if there exists strictly proper stable transfer matrices $\mathbf{R}, \mathbf{M} \in \frac{1}{z}\mathcal{RH}_\infty$ lie in the subspace described by (3.4a). This idea is formally stated and proved by the following lemma.

Lemma 3 (Stabilizability)**.** *The pair* (A, B_2) *is stabilizable if and only if the affine subspace defined by* (3.4) *is non-empty.*

Proof. We first show that the stabilizability of (A, B_2) implies that there exist transfer matrices $\mathbf{R}, \mathbf{M} \in \frac{1}{z}\mathcal{RH}_\infty$ satisfying equation (3.4a). From the definition of stabilizability, there exists a matrix F such that $A + B_2F$ is a stable matrix. Substituting the state feedback control law $u = Fx$ into (3.2), we have $\mathbf{x} = (zI - A - B_2F)^{-1}\boldsymbol{\delta}_{\boldsymbol{x}}$ and $\mathbf{u} = F(zI - A - B_2F)^{-1}\boldsymbol{\delta}_{\boldsymbol{x}}$. The system response is given by $\mathbf{R} = (zI - A - B_2F)^{-1}$ and $\mathbf{M} = F(zI - A - B_2F)^{-1}$, which lie in $\frac{1}{z}\mathcal{RH}_\infty$ and are a solution to (3.4a).

For the opposite direction, we note that $\mathbf{R}, \mathbf{M} \in \mathcal{RH}_\infty$ implies that these transfer matrices do not have poles outside the unit circle $|z| \geq 1$. From (3.4a), we further observe that $\begin{bmatrix} zI - A & -B_2 \end{bmatrix}$ is right invertible in the region where $\mathbf{R}$ and $\mathbf{M}$ do not have poles, with $\begin{bmatrix} \mathbf{R}^\top & \mathbf{M}^\top \end{bmatrix}^\top$ being its right inverse. This then implies that $\begin{bmatrix} zI - A & -B_2 \end{bmatrix}$ has full row rank for all $|z| \geq 1$. This is equivalent to the PBH test [15] for stabilizability, proving the claim. □

Thus Lemma 3 provides an alternative definition of (state feedback) stabilizability via the conditions described in (3.4) — in particular, stable achievable responses exist only if the state feedback system is stabilizable.

The necessity of conditions (3.4) is provided in the following lemma.

Lemma 4 (Necessity of conditions (3.4))**.** *Consider the state feedback system* (3.1). *Let* $(\mathbf{R}, \mathbf{M})$ *be the system response achieved by an internally stabilizing controller* $\mathbf{K}$. *Then,* $(\mathbf{R}, \mathbf{M})$ *is a solution of* (3.4).

Proof. See Appendix 3.A in the end of this chapter. □

3.1.2 Sufficiency

Here we show that for any system response $\{\mathbf{R}, \mathbf{M}\}$ lying in the affine subspace (3.4), we can construct an internally stabilizing controller $\mathbf{K}$ that leads to the desired system response (3.3).

A partial solution is provided in our prior work [71], where we give a construction for finite impulse response (FIR) system responses $\{\mathbf{R}, \mathbf{M}\}$. Here we extend these results to infinite impulse response (IIR) system responses, and provide a proof of internal stability for the proposed controller structure.

In [71], we considered FIR system responses of horizon T, i.e., $\mathbf{R}, \mathbf{M} \in \mathcal{F}_T$, and proposed the following disturbance-based controller implementation:

$$\hat{\delta_x}[t] = x[t] - \hat{x}[t] \tag{3.5a}$$

$$u[t] = \sum_{\tau=0}^{T-1} M[\tau+1]\hat{\delta_x}[t-\tau] \tag{3.5b}$$

$$\hat{x}[t+1] = \sum_{\tau=0}^{T-2} R[\tau+2]\hat{\delta_x}[t-\tau]. \tag{3.5c}$$

The internal states of the controller (3.5) should be interpreted as follows: $\hat{\delta_x}$ is the controller estimate of the state disturbance, and $\hat{x}$ is a desired or reference state trajectory. The estimated disturbance $\hat{\delta_x}[t]$ is computed by taking the difference between the current state measurement $x[t]$ and the current reference state value $\hat{x}[t]$. The control action $u[t]$ and the next reference state value $\hat{x}[t+1]$ are then computed using past estimated disturbances $\hat{\delta_x}[t-T+1], \ldots, \hat{\delta_x}[t]$.

Taking the z-transform of equations (3.5), we obtain their representation in the frequency domain

$$\hat{\boldsymbol{\delta_x}} = \mathbf{x} - \hat{\mathbf{x}} \tag{3.6a}$$

$$\mathbf{u} = z\mathbf{M}\hat{\boldsymbol{\delta_x}} \tag{3.6b}$$

$$\hat{\mathbf{x}} = (z\mathbf{R} - I)\hat{\boldsymbol{\delta_x}}. \tag{3.6c}$$

Combining equations (3.6) with (3.2) and (3.4), one can verify that the estimated disturbance $\hat{\delta_x}[t]$ indeed reconstructs the true disturbance $\delta_x[t-1]$ that perturbed the plant at time $t-1$; hence $\hat{\boldsymbol{\delta_x}} = z^{-1}\boldsymbol{\delta_x}$. It is then straightforward to show that the desired system response $\{\mathbf{R}, \mathbf{M}\}$ satisfying $\mathbf{x} = \mathbf{R}\boldsymbol{\delta_x}$ and $\mathbf{u} = \mathbf{M}\boldsymbol{\delta_x}$ is achieved. Note that the previous argument holds for any FIR horizon T as well as for $T = \infty$.

Remark 2. *From* (3.6)*, the control action* $\mathbf{u}$ *can be expressed as* $\mathbf{u} = \mathbf{M}\mathbf{R}^{-1}\mathbf{x}$*. We can therefore also implement the controller defined in* (3.6) *via the dynamic state feedback gain* $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$*.*[2] *However, we argue that the disturbance-based implementation in* (3.6) *has significant advantages over a traditional state feedback implementation — specifically, this implementation allows us to connect constraints imposed on the system response to constraints on the controller implementation. The implementation* (3.6) *is the key to make a localized linear quadratic regulator (LLQR) optimal controller (cf., Chapter 5) scalable to implement.*

[2] As $\mathbf{R}$ is strictly proper, $\mathbf{R}^{-1}$ is not proper. However, $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$ can be verified to always be proper.

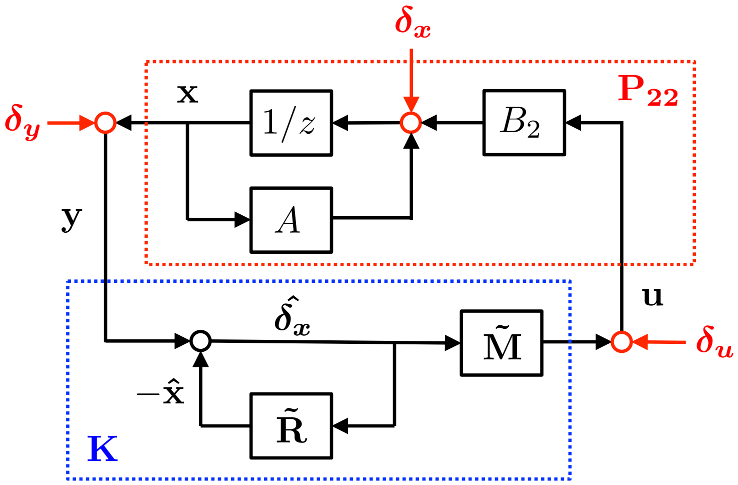


Figure 3.1: The proposed state feedback controller structure, with $\tilde{\mathbf{R}} = I - z\mathbf{R}$ and $\tilde{\mathbf{M}} = z\mathbf{M}$.

It remains to be shown that the controller implementation (3.6) internally stabilizes the plant (3.1). We consider the block diagram shown in Figure 3.1, where here $\tilde{\mathbf{R}} = I - z\mathbf{R}$ and $\tilde{\mathbf{M}} = z\mathbf{M}$. It can be checked that $\tilde{\mathbf{R}} = I - z\mathbf{R} = -A\mathbf{R} - B_2\mathbf{M} \in \frac{1}{z}\mathcal{RH}_\infty$ and $\tilde{\mathbf{M}} = z\mathbf{M} \in \mathcal{RH}_\infty$, and hence the internal feedback loop between $\hat{\boldsymbol{\delta}}_{\boldsymbol{x}}$ and the reference state trajectory $\hat{\mathbf{x}}$ is well defined.

As is standard, we introduce external perturbations $\boldsymbol{\delta}_{\boldsymbol{x}}, \boldsymbol{\delta}_{\boldsymbol{y}}$, and $\boldsymbol{\delta}_{\boldsymbol{u}}$ into the system and note that the perturbations entering other links of the block diagram can be expressed as a combination of $(\boldsymbol{\delta}_{\boldsymbol{x}}, \boldsymbol{\delta}_{\boldsymbol{y}}, \boldsymbol{\delta}_{\boldsymbol{u}})$ being acted upon by some stable transfer matrices.[3] Therefore, the standard definition of internal stability applies, and we can use a bounded-input bounded-output argument (e.g., Lemma 2 in Appendix 2.A) to conclude that it suffices to check the stability of the nine closed loop transfer matrices from perturbations $(\boldsymbol{\delta}_{\boldsymbol{x}}, \boldsymbol{\delta}_{\boldsymbol{y}}, \boldsymbol{\delta}_{\boldsymbol{u}})$ to the internal variables $(\mathbf{x}, \mathbf{u}, \hat{\boldsymbol{\delta}}_{\boldsymbol{x}})$ to determine the internal stability of the structure as a whole.

As all blocks in Figure 3.1 are stable filters, it follows that if the origin $(x, \hat{\delta}_x) = (0, 0)$ is asymptotically stable then any other signals in the block diagram will decay asymptotically. This is equivalent to the conventional notion of internal stability [81], which we recall here for the reader before stating and proving that the proposed controller implementation is internally stabilizing.

Definition 4. *The interconnection in Figure 3.1 is internally stable if the origin* $(x, \hat{\delta}_x) = (0, 0)$ *is asymptotically stable, i.e.,* $x[t], \hat{\delta}_x[t] \to 0$ *for* $t \to \infty$ *for all initial*

[3] The matrix A may define an unstable system, but viewed as an element of $\mathcal{F}_0$, defines a stable (FIR) transfer matrix.

conditions when the external perturbations $\delta_x, \delta_y, \delta_u$ *in Figure 3.1 are set to* 0.

Lemma 5 (Sufficiency of conditions (3.4))**.** *Consider the state feedback system* (3.1)*. Given any system response* $\{\mathbf{R}, \mathbf{M}\}$ *lying in the affine subspace described by* (3.4)*, the state feedback controller* $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$*, with structure shown in Figure 3.1, internally stabilizes the plant. In addition, the desired system response, as specified by* $\mathbf{x} = \mathbf{R}\boldsymbol{\delta}_{\boldsymbol{x}}$ *and* $\mathbf{u} = \mathbf{M}\boldsymbol{\delta}_{\boldsymbol{x}}$*, is achieved.*

Proof. We first note that from Figure 3.1, we can express the state feedback controller $\mathbf{K}$ as $\mathbf{K} = \tilde{\mathbf{M}}(I - \tilde{\mathbf{R}})^{-1} = (z\mathbf{M})(z\mathbf{R})^{-1} = \mathbf{M}\mathbf{R}^{-1}$. Now, for any system response $\{\mathbf{R}, \mathbf{M}\}$ lying in the affine subspace described by (3.4), we construct a controller using the structure given in Figure 3.1. To show that the constructed controller internally stabilizes the plant, we list the following equations from Figure 3.1:

$$z\mathbf{x} = A\mathbf{x} + B_2\mathbf{u} + \boldsymbol{\delta}_{\boldsymbol{x}} \tag{3.7a}$$

$$\mathbf{u} = \tilde{\mathbf{M}}\hat{\boldsymbol{\delta}}_{\boldsymbol{x}} + \boldsymbol{\delta}_{\boldsymbol{u}} \tag{3.7b}$$

$$\hat{\boldsymbol{\delta}}_{\boldsymbol{x}} = \mathbf{x} + \boldsymbol{\delta}_{\boldsymbol{y}} + \tilde{\mathbf{R}}\hat{\boldsymbol{\delta}}_{\boldsymbol{x}}. \tag{3.7c}$$

Routine calculations (see Appendix 3.A for details) show that the closed loop transfer matrices from $(\boldsymbol{\delta}_{\boldsymbol{x}}, \boldsymbol{\delta}_{\boldsymbol{y}}, \boldsymbol{\delta}_{\boldsymbol{u}})$ to $(\mathbf{x}, \mathbf{u}, \hat{\boldsymbol{\delta}}_{\boldsymbol{x}})$ are given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \hat{\boldsymbol{\delta}}_{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\tilde{\mathbf{R}} - \mathbf{R}A & \mathbf{R}B_2 \\ \mathbf{M} & \tilde{\mathbf{M}} - \mathbf{M}A & I + \mathbf{M}B_2 \\ \frac{1}{z}I & I - \frac{1}{z}A & \frac{1}{z}B_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{\boldsymbol{x}} \\ \boldsymbol{\delta}_{\boldsymbol{y}} \\ \boldsymbol{\delta}_{\boldsymbol{u}} \end{bmatrix}. \tag{3.8}$$

As all nine transfer matrices in (3.8) are stable, the implementation in Figure 3.1 is internally stable. Furthermore, the desired system response $\{\mathbf{R}, \mathbf{M}\}$, from $\boldsymbol{\delta}_{\boldsymbol{x}}$ to $(\mathbf{x}, \mathbf{u})$, is achieved. □

3.1.3 Summary and corollary

The proof of Theorem 1 is then straightforward.

Proof of Theorem 1. The statements follow directly by combining the results of Lemma 4 and 5. □

Theorem 1 provides a necessary and sufficient condition for the system response $\{\mathbf{R}, \mathbf{M}\}$ to be stable and achievable, in that elements of the affine subspace defined by (3.4) parameterize all stable system responses achievable via state-feedback, as well as the internally stabilizing controllers that achieve them.

We note that the analysis for the state feedback problem can be applied to the state estimation problem by considering the dual to a full control system (c.f., §16.5 in [81]). For instance, the following corollary to Lemma 3 gives an alternative definition of the detectability of pair (A, C_2) [74].

Corollary 1 (Detectability)**.** *The pair* (A, C_2) *is detectable if and only if the following conditions are feasible:*

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = I \tag{3.9a}$$

$$\mathbf{R}, \mathbf{N} \in \frac{1}{z} \mathcal{RH}_\infty. \tag{3.9b}$$

A parameterization of all detectable observers can be constructed using the affine subspace (3.9) in a manner analogous to that described above. We will give a more detailed discussion about state estimation application in Section 5.6.

Finally, we note that Theorem 1 can be extended to continuous time system immediately by replacing z-transform variable z to the Laplace transform variable s. The corollary of Theorem 1 for continuous time system is therefore given as follows:

Corollary 2 (Theorem 1 for continuous time systems)**.** *For a continuous time state feedback system with state space realization* (3.1)*, the following are true:*

(a) The affine subspace defined by

$$\begin{bmatrix} sI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \tag{3.10a}$$

$$\mathbf{R}, \mathbf{M} \in \frac{1}{s} \mathcal{RH}_\infty \tag{3.10b}$$

parameterizes all system responses from $\boldsymbol{\delta_x}$ *to* $(\mathbf{x}, \mathbf{u})$ *achievable by an internally stabilizing state feedback controller* $\mathbf{K}$*.*

(b) For any transfer matrices $\{\mathbf{R}, \mathbf{M}\}$ *satisfying* (3.10)*, the controller* $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$ *is internally stabilizing and achieves the desired system response* $\mathbf{x} = \mathbf{R}\, \boldsymbol{\delta_x}$ *and* $\mathbf{u} = \mathbf{M}\, \boldsymbol{\delta_x}$.

3.2 Output Feedback for Strictly Proper Systems

We now extend the arguments of the previous section to the output feedback setting, and begin by considering the case of a strictly proper plant

$$\mathbf{P} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right]. \tag{3.11}$$

Letting $\delta_x[t] = B_1 w[t]$ denote the disturbance on the state, and $\delta_y[t] = D_{21} w[t]$ denote the disturbance on the measurement, the dynamics defined by plant (3.11) can be written as

$$\begin{aligned} x[t+1] &= Ax[t] + B_2 u[t] + \delta_x[t] \\ y[t] &= C_2 x[t] + \delta_y[t]. \end{aligned} \tag{3.12}$$

Analogous to the state-feedback case, we define a system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ from perturbations $(\boldsymbol{\delta_x}, \boldsymbol{\delta_y})$ to state and control inputs $(\mathbf{x}, \mathbf{u})$ via the following relation:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta_x} \\ \boldsymbol{\delta_y} \end{bmatrix}. \tag{3.13}$$

Substituting the output feedback control law $\mathbf{u} = \mathbf{K}\mathbf{y}$ into the z-transform of system equation (3.12), we obtain

$$(zI - A - B_2\mathbf{K}C_2)\mathbf{x} = \boldsymbol{\delta_x} + B_2\mathbf{K}\boldsymbol{\delta_y}.$$

For a proper controller $\mathbf{K}$, the transfer matrix $(zI - A - B_2\mathbf{K}C_2)$ is always invertible because its leading coefficient zI is invertible, hence we obtain the following expressions for the system response (3.13) in terms of an output feedback controller $\mathbf{K}$:

$$\begin{aligned} \mathbf{R} &= (zI - A - B_2\mathbf{K}C_2)^{-1} \\ \mathbf{M} &= \mathbf{K}C_2\mathbf{R} \\ \mathbf{N} &= \mathbf{R}B_2\mathbf{K} \\ \mathbf{L} &= \mathbf{K} + \mathbf{K}C_2\mathbf{R}B_2\mathbf{K}. \end{aligned} \tag{3.14}$$

We now present one of the main results of this chapter: an algebraic characterization of the set $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ of output-feedback system responses that are achievable by an internally stabilizing controller $\mathbf{K}$, as stated in the following theorem.

Theorem 2 (System Level Parameterization for Output Feedback Systems with Strictly Proper Plants)**.** *For the output feedback system* (3.11)*, the following are true:*

(a) The affine subspace described by:

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{3.15a}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{3.15b}$$

$$\mathbf{R}, \mathbf{M}, \mathbf{N} \in \frac{1}{z}\mathcal{RH}_\infty, \quad \mathbf{L} \in \mathcal{RH}_\infty \tag{3.15c}$$

parameterizes all system responses (3.14) *achievable by an internally stabilizing controller* $\mathbf{K}$*.*

(b) For any transfer matrices $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ *satisfying* (3.15)*, the controller* $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$ *is internally stabilizing and achieves the desired response* (3.14)*.*

As the equations (3.15a) - (3.15c) characterize the set of all stable and achievable system responses, we call it a system level parameterization (SLP).

3.2.1 Necessity

As was the case for the state-feedback setting, the necessity of a stable and achievable system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ lying in the affine subspace (3.15) follows from rote calculation. We first provide some intuition about how the equality constraints (3.15a) - (3.15b) are derived from the relations (3.14). Using the identity $(zI - A - B_2\mathbf{K}C_2)\mathbf{R} = I$ and the relation $\mathbf{M} = \mathbf{K}C_2\mathbf{R}$, we get $(zI - A)\mathbf{R} - B_2\mathbf{M} = I$. Likewise, we have the relation $(zI - A)\mathbf{N} - B_2\mathbf{L} = 0$. Therefore, the system response must satisfy the equality constraint in (3.15a). Similarly, using the identity $\mathbf{R}(zI - A - B_2\mathbf{K}C_2) = I$, we know that the system response must also satisfy (3.15b). Besides, from the relations in (3.14), we note that $\mathbf{R}, \mathbf{M}$, and $\mathbf{N}$ must be strictly proper, and all the system response are stable because $\mathbf{K}$ is an internally stabilizing controller. This leads to the constraint (3.15c). Intuitively, the equations (3.15a) - (3.15c) are feasible as long as the system matrices (A, B_2) is stabilizable and (A, C_2) is detectable. This is formally stated in the following lemma.

Lemma 6 (Stabilizability and Detectability)**.** *The triple* (A, B_2, C_2) *is stabilizable and detectable if and only if the affine subspace described by* (3.15) *is non-empty.*

Proof. See Appendix 3.A in the end of this chapter. □

Lemma 6 provides an alternative characterization of stabilizability and detectability via the conditions described in (3.15) — in particular, stable achievable system responses (3.14) exist only if the output feedback system is stabilizable and detectable. The next lemma shows that (3.15) is a necessary condition for the system response to be stable and achievable.

Lemma 7 (Necessity of conditions (3.15))**.** *Consider the output feedback system* (3.11). *Let* $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$*, with* $\mathbf{x} = \mathbf{R}\boldsymbol{\delta}_x + \mathbf{N}\boldsymbol{\delta}_y$ *and* $\mathbf{u} = \mathbf{M}\boldsymbol{\delta}_x + \mathbf{L}\boldsymbol{\delta}_y$*, be the system response achieved by an internally stabilizing control law* $\mathbf{u} = \mathbf{K}\mathbf{y}$*. Then,* $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ *lies in the affine subspace described by* (3.15).

Proof. See Appendix 3.A in the end of this chapter. □

3.2.2 Sufficiency

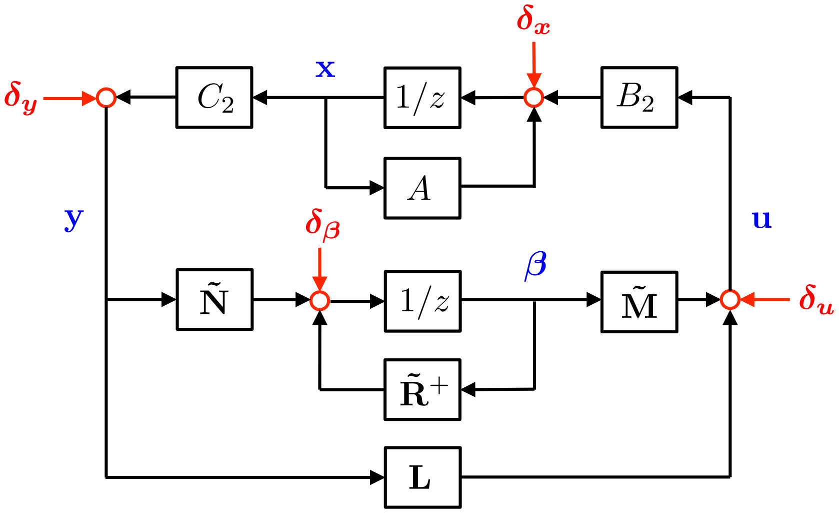


Figure 3.2: The proposed output feedback controller structure, with $\tilde{\mathbf{R}}^+ = z\tilde{\mathbf{R}} = z(I - z\mathbf{R})$, $\tilde{\mathbf{M}} = z\mathbf{M}$, and $\tilde{\mathbf{N}} = -z\mathbf{N}$.

Here we show that for any system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ lying in the affine subspace (3.15), there exists an internally stabilizing controller $\mathbf{K}$ that leads to the desired system response (3.14). From the relations in (3.14), we notice the identity $\mathbf{K} = \mathbf{L} - \mathbf{K}C_2\mathbf{R}B_2\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$. This relation leads to the controller structure given in Figure 3.2, with $\tilde{\mathbf{R}}^+ = z\tilde{\mathbf{R}} = z(I - z\mathbf{R})$, $\tilde{\mathbf{M}} = z\mathbf{M}$, and $\tilde{\mathbf{N}} = -z\mathbf{N}$. As was the case for the state feedback setting, it can be verified that $\tilde{\mathbf{R}}^+, \tilde{\mathbf{M}}$, and $\tilde{\mathbf{N}}$ are all in

$\mathcal{RH}_\infty$. Therefore, the structure given in Figure 3.2 is well defined. The controller implementation of Figure 3.2 is governed by the following equations:

$$\begin{aligned} z\boldsymbol{\beta} &= \tilde{\mathbf{R}}^{+}\boldsymbol{\beta} + \tilde{\mathbf{N}}\mathbf{y} \\ \mathbf{u} &= \tilde{\mathbf{M}}\boldsymbol{\beta} + \mathbf{L}\mathbf{y}. \end{aligned} \tag{3.16}$$

The control implementation equations (3.16) can be interpreted as an extension of the state-space realization (2.5) of a controller $\mathbf{K}$. In particular, in the realization equations (3.16) we allow the constant matrices A_K, B_K, C_K, D_K of the state-space realization (2.5) to be stable proper transfer matrices $\tilde{\mathbf{R}}^{+}$, $\tilde{\mathbf{M}}$, $\tilde{\mathbf{N}}$, and $\mathbf{L}$. The benefit of this implementation is that arbitrary convex constraints imposed on the transfer matrices $\tilde{\mathbf{R}}^{+}, \tilde{\mathbf{M}}, \tilde{\mathbf{N}}, \mathbf{L}$ carry over directly to the controller implementation. We show in Chapter 4.2 that this allows for a class of structural (locality) constraints to be imposed on the system response (and hence the controller) that are crucial for extending controller synthesis methods to large-scale systems (this will be discussed in details in Chapters 6 - 7). In contrast, we recall that imposing general convex constraints on the controller $\mathbf{K}$ or its state-space realization A_K, B_K, C_K, D_K cannot be done in a computationally efficient manner.

What remains to be shown is that the proposed controller implementation (3.16) is internally stabilizing and achieves the desired system response (3.14). As was the case for the state feedback setting, all of the blocks in Figure 3.2 are stable filters — thus, as long as the origin $(x, \beta) = (0, 0)$ is asymptotically stable, all signals internal to the block diagram will decay to zero. To check the internal stability of the structure, we introduce external perturbations $\boldsymbol{\delta_x}$, $\boldsymbol{\delta_y}$, $\boldsymbol{\delta_u}$, and $\boldsymbol{\delta_\beta}$ to the system. The perturbations appearing on other links of the block diagram can all be expressed as a combination of the perturbations $(\boldsymbol{\delta_x}, \boldsymbol{\delta_y}, \boldsymbol{\delta_u}, \boldsymbol{\delta_\beta})$ being acted upon by some stable transfer matrices, and so it suffices to check the input-output stability of the closed loop transfer matrices from perturbations $(\boldsymbol{\delta_x}, \boldsymbol{\delta_y}, \boldsymbol{\delta_u}, \boldsymbol{\delta_\beta})$ to controller signals $(\mathbf{x}, \mathbf{u}, \mathbf{y}, \boldsymbol{\beta})$ to determine the internal stability of the structure [81].

With this discussion in mind, we formally define internal stability for the controller structure of Figure 3.2, and state and prove the sufficiency of the conditions stated in Theorem 2.

Definition 5. *The interconnection in Figure 3.2 is said to be internally stable if the origin* $(x, \beta) = (0, 0)$ *is asymptotically stable, i.e.,* $x[t], \beta[t] \to 0$ *for* $t \to \infty$ *from any initial condition when the perturbations* $\delta_x, \delta_y, \delta_u, \delta_\beta$ *in Figure 3.2 are* 0.

Lemma 8. *Consider the output feedback system* (3.11). *For any system response* $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ *lying in the affine subspace defined by* (3.15), *the controller* $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$ *(with structure shown in Figure 3.2) internally stabilizes the plant. In addition, the desired system response, as specified by* $\mathbf{x} = \mathbf{R}\boldsymbol{\delta}_x + \mathbf{N}\boldsymbol{\delta}_y$ *and* $\mathbf{u} = \mathbf{M}\boldsymbol{\delta}_x + \mathbf{L}\boldsymbol{\delta}_y$, *is achieved.*

Proof. For any system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ lying in the affine subspace defined by (3.15), we construct a controller using the structure given in Figure 3.2. We now check the stability of the closed loop transfer matrices from the perturbations $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y, \boldsymbol{\delta}_u, \boldsymbol{\delta}_\beta)$ to the internal variables $(\mathbf{x}, \mathbf{u}, \mathbf{y}, \boldsymbol{\beta})$. We have the following equations from Figure 3.2:

$$z\mathbf{x} = A\mathbf{x} + B_2\mathbf{u} + \boldsymbol{\delta}_x \tag{3.17a}$$

$$\mathbf{y} = C_2\mathbf{x} + \boldsymbol{\delta}_y \tag{3.17b}$$

$$z\boldsymbol{\beta} = \tilde{\mathbf{R}}^+\boldsymbol{\beta} + \tilde{\mathbf{N}}\mathbf{y} + \boldsymbol{\delta}_\beta \tag{3.17c}$$

$$\mathbf{u} = \tilde{\mathbf{M}}\boldsymbol{\beta} + \mathbf{L}\mathbf{y} + \boldsymbol{\delta}_u. \tag{3.17d}$$

Combining these equations with the relations in (3.15a) - (3.15b), we summarize the closed loop transfer matrices from $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y, \boldsymbol{\delta}_u, \boldsymbol{\delta}_\beta)$ to $(\mathbf{x}, \mathbf{u}, \mathbf{y}, \boldsymbol{\beta})$ in Table 3.1 (see Appendix 3.A for a detailed derivation).

Table 3.1: Closed Loop Maps from Perturbations to Internal Variables

	$\boldsymbol{\delta}_x$	$\boldsymbol{\delta}_y$	$\boldsymbol{\delta}_u$	$\boldsymbol{\delta}_\beta$
$\mathbf{x}$	$\mathbf{R}$	$\mathbf{N}$	$\mathbf{R}B_2$	$\frac{1}{z}\mathbf{N}C_2$
$\mathbf{u}$	$\mathbf{M}$	$\mathbf{L}$	$I + \mathbf{M}B_2$	$\frac{1}{z}\mathbf{L}C_2$
$\mathbf{y}$	$C_2\mathbf{R}$	$I + C_2\mathbf{N}$	$C_2\mathbf{R}B_2$	$\frac{1}{z}C_2\mathbf{N}C_2$
$\boldsymbol{\beta}$	$-\frac{1}{z}B_2\mathbf{M}$	$-\frac{1}{z}B_2\mathbf{L}$	$-\frac{1}{z}B_2\mathbf{M}B_2$	$\frac{1}{z}I - \frac{1}{z^2}(A + B_2\mathbf{L}C_2)$

Equation (3.15c) implies that all sixteen transfer matrices in Table 3.1 are stable, so the implementation in Figure 3.2 is internally stable. Furthermore, the desired system response from $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y)$ to $(\mathbf{x}, \mathbf{u})$ is achieved. □

3.2.3 Alternative controller implementations

Before we summarize the results for Theorem 2, we first show that care must be taken when dealing with open loop unstable systems. In particular, we demonstrate the necessity of considering perturbations on the internal controller state $\boldsymbol{\beta}$ and control input $\mathbf{u}$ for unstable plants. Such perturbations can arise, for instance, from using floating point arithmetic within the controller, or from quantization at the actuators.

If we set $\boldsymbol{\delta}_u$ and $\boldsymbol{\delta}_\beta$ to 0, it follows that $\boldsymbol{\beta} = -\frac{1}{z}B_2\mathbf{u}$ from Table 3.1. This leads to a simpler controller implementation given by $\mathbf{u} = \mathbf{L}\mathbf{y} - \mathbf{M}B_2\mathbf{u}$, with the corresponding controller structure shown in Figure 3.3a. This implementation can also be obtained from the identity $\mathbf{K} = (I + \mathbf{M}B_2)^{-1}\mathbf{L}$, which follows from the relations in (3.14). Unfortunately, as shown below, this implementation is internally stable only when the open loop plant is stable.

For the controller implementation and structure shown in Figure 3.3a, the closed loop transfer matrices from perturbations to the internal variables are given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{N} & \mathbf{R}B_2 & (zI - A)^{-1}B_2 \\ \mathbf{M} & \mathbf{L} & I + \mathbf{M}B_2 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_x \\ \boldsymbol{\delta}_y \\ \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\beta \end{bmatrix}. \tag{3.18}$$

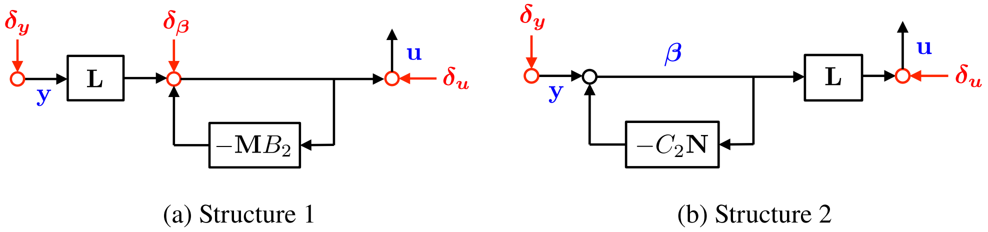
When A defines a stable system, the implementation in Figure 3.3a is internally stable. However, when the open loop plant is unstable (and the realization (A, B_2) is stabilizable), the transfer matrix $(zI - A)^{-1}B_2$ is unstable. From (3.18), the effect of the perturbation $\boldsymbol{\delta}_\beta$ can lead to instability of the closed loop system. This structure thus shows the necessity of introducing and analyzing the effects of perturbations $\boldsymbol{\delta}_\beta$ on the controller internal state.

Alternatively, if we start with the identity $\mathbf{K} = \mathbf{L}(I + C_2\mathbf{N})^{-1}$, which also follows from (3.14), we obtain the controller structure shown in Figure 3.3b. The closed loop map from perturbations to internal signals is then given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{N} & \mathbf{R}B_2 \\ \mathbf{M} & \mathbf{L} & I + \mathbf{M}B_2 \\ C_2(zI - A)^{-1} & I & C_2(zI - A)^{-1}B_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_x \\ \boldsymbol{\delta}_y \\ \boldsymbol{\delta}_u \end{bmatrix}.$$

As can be seen, the controller implementation is once again internally stable only when the open loop plant is stable (if the realization (A, C_2) is detectable). This structure thus shows the necessity of introducing and analyzing the effects of perturbations on the controller internal state $\boldsymbol{\beta}$.

Of course, when the open loop system is stable, the controller structures illustrated below may be appealing as they are simpler and easier to implement. In fact, when the open loop system is stable, the system response $\mathbf{L}$ is equal to the Youla parameter $\mathbf{Q}$ (this will be shown in Lemma 9 in Section 4.2). In this case, the controller implementation $\mathbf{u} = \mathbf{L}\mathbf{y} - \mathbf{M}B_2\mathbf{u}$ can be written as $\mathbf{u} = \mathbf{Q}\mathbf{y} - \mathbf{Q}\mathbf{P}_{22}\mathbf{u}$, and the controller structure shown in Figure 3.3a is just the implementation of Internal Model Control (IMC) [21, 51].



(a) Structure 1

(b) Structure 2

Figure 3.3: Alternative controller structures for stable systems.

3.2.4 Summary

The proof of Theorem 2 is straightforward.

Proof of Theorem 2. The statements follow directly by combining the results of Lemma 7 and 8. □

Theorem 2 provides a necessary and sufficient condition for the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ to be stable and achievable, in that elements of the affine subspace defined by (3.15) parameterize all stable achievable system responses, as well as all internally stabilizing controllers that achieve them.

Theorem 2 can be extended to continuous time system immediately by replacing z-transform variable z to the Laplace transform variable s. The corollary of Theorem 2 for continuous time system is therefore given as follows:

Corollary 3 (Theorem 2 for continuous time systems)**.** *For a continuous time output feedback system with state space realization* (3.11)*, the following are true:*

(a) The affine subspace described by:

$$\begin{bmatrix} sI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{3.19a}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} sI - A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{3.19b}$$

$$\mathbf{R}, \mathbf{M}, \mathbf{N} \in \frac{1}{s}\mathcal{RH}_\infty, \quad \mathbf{L} \in \mathcal{RH}_\infty \tag{3.19c}$$

parameterizes all system responses (3.13) *achievable by an internally stabilizing controller* **K**.

(b) For any transfer matrices $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ *satisfying* (3.19)*, the controller* $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$ *is internally stabilizing and achieves the desired response* (3.13).

3.3 Output Feedback for Proper Systems

Finally, for a general proper plant model (2.1) with $D_{22} \neq 0$, we define a new measurement $\bar{y}[t] = y[t] - D_{22}u[t]$. This leads to the controller structure shown in Figure 3.4. In this case, the closed loop transfer matrices from $\boldsymbol{\delta_u}$ to the internal variables become

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \mathbf{y} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{R}B_2 + \mathbf{N}D_{22} \\ I + \mathbf{M}B_2 + \mathbf{L}D_{22} \\ C_2\mathbf{R}B_2 + D_{22} + C_2\mathbf{N}D_{22} \\ -\frac{1}{z}B_2(\mathbf{M}B_2 + \mathbf{L}D_{22}) \end{bmatrix} \boldsymbol{\delta_u}.$$

The remaining entries of Table 3.1 remain the same. Therefore, the controller structure shown in Figure 3.4 internally stabilizes the plant. The parameterization of all stable achievable system responses as well as the set of well-posed internally stabilizing controllers follow immediately from Theorem 2.

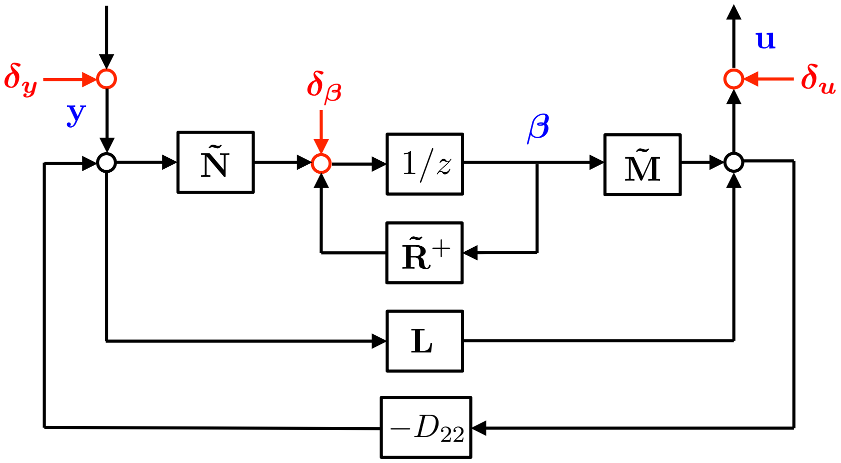


Figure 3.4: The proposed output feedback controller structure for $D_{22} \neq 0$.

APPENDIX

3.A Proof of Lemmas

Proof of Lemma 4. Consider an internally stabilizing controller $\mathbf{u} = \mathbf{K}\mathbf{x}$ with its state space realization given by

$$\begin{aligned} z\boldsymbol{\xi} &= A_K\boldsymbol{\xi} + B_K\mathbf{x} \\ \mathbf{u} &= C_K\boldsymbol{\xi} + D_K\mathbf{x}. \end{aligned} \tag{3.20}$$

Combining (3.20) and (3.2), we have the closed loop system dynamics given by

$$\begin{bmatrix} z\mathbf{x} \\ z\boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} A + B_2 D_k & B_2 C_k \\ B_k & A_k \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} \boldsymbol{\delta}_{\boldsymbol{x}}. \tag{3.21}$$

As the controller is internally stabilizing, we know that the state matrix in (3.21) is a stable matrix (Lemma 1 in Appendix 2.A). The system response achieved by the control law $\mathbf{u} = \mathbf{K}\mathbf{x}$ is given by

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = \left[\begin{array}{cc|c} A + B_2 D_k & B_2 C_k & I \\ B_k & A_k & 0 \\ \hline I & 0 & 0 \\ D_k & C_k & 0 \end{array}\right]. \tag{3.22}$$

It is clear that the system response (3.22) is strictly proper and stable, and thus (3.4b) is satisfied. From (3.22), we have

$$\begin{aligned} (zI - A)\mathbf{R} - B_2\mathbf{M} &= z\mathbf{R} - A\mathbf{R} - B_2\mathbf{M} \\ &= \left[\begin{array}{cc|c} A + B_2 D_k & B_2 C_k & I \\ B_k & A_k & 0 \\ \hline A + B_2 D_k & B_2 C_k & I \end{array}\right] - \left[\begin{array}{cc|c} A + B_2 D_k & B_2 C_k & I \\ B_k & A_k & 0 \\ \hline A & 0 & 0 \end{array}\right] \\ &\quad - \left[\begin{array}{cc|c} A + B_2 D_k & B_2 C_k & I \\ B_k & A_k & 0 \\ \hline B_2 D_k & B_2 C_k & 0 \end{array}\right] \\ &= I, \end{aligned}$$

which shows that the equality constraint (3.4a) is satisfied for arbitrary (A_k, B_k, C_k, D_k). This completes the proof. □

Proof of Lemma 5 (from (3.7) *to* (3.8)*).* Recall that we have $\tilde{\mathbf{R}} = I - z\mathbf{R}$ and $\tilde{\mathbf{M}} = z\mathbf{M}$. We can rewrite (3.7a) - (3.7c) as

$$(zI - A)\mathbf{x} = B_2\mathbf{u} + \boldsymbol{\delta_x} \tag{3.24a}$$

$$\mathbf{u} = z\mathbf{M}\hat{\boldsymbol{\delta_x}} + \boldsymbol{\delta_u} \tag{3.24b}$$

$$\mathbf{x} = z\mathbf{R}\hat{\boldsymbol{\delta_x}} - \boldsymbol{\delta_y}. \tag{3.24c}$$

Substituting (3.24b) and (3.24c) into (3.24a), we have

$$z(zI - A)\mathbf{R}\hat{\boldsymbol{\delta_x}} - (zI - A)\boldsymbol{\delta_y} = zB_2\mathbf{M}\hat{\boldsymbol{\delta_x}} + B_2\boldsymbol{\delta_u} + \boldsymbol{\delta_x}.$$

Moving $\hat{\boldsymbol{\delta_x}}$ to the left-hand-side and using the identity (3.4a) from the assumption, we have

$$z\hat{\boldsymbol{\delta_x}} = (zI - A)\boldsymbol{\delta_y} + B_2\boldsymbol{\delta_u} + \boldsymbol{\delta_x},$$

which can be simplified into

$$\hat{\boldsymbol{\delta_x}} = \frac{1}{z}\boldsymbol{\delta_x} + (I - \frac{1}{z}A)\boldsymbol{\delta_y} + \frac{1}{z}B_2\boldsymbol{\delta_u}. \tag{3.25}$$

Substituting (3.25) into (3.24b) and (3.24c) leads to equation (3.8). □

Proof of Lemma 6. For one direction, note that the feasibility of (3.15) implies the feasibility of both (3.4) and (3.9). Using Lemma 3 and Corollary 1, we know that the triple (A, B_2, C_2) is stabilizable and detectable.

For the opposite direction, given a stabilizable (A, B_2) and a detectable (A, C_2), let $(\mathbf{R}_1, \mathbf{M}_1)$ be a feasible solution of (3.4) and $(\mathbf{R}_2, \mathbf{N}_2)$ be a feasible solution of (3.9). We use these to construct the feasible solution to (3.15)

$$\mathbf{R} = \mathbf{R}_1 + \mathbf{R}_2 - \mathbf{R}_1(zI - A)\mathbf{R}_2 \tag{3.26a}$$

$$\mathbf{M} = \mathbf{M}_1 - \mathbf{M}_1(zI - A)\mathbf{R}_2 \tag{3.26b}$$

$$\mathbf{N} = \mathbf{N}_2 - \mathbf{R}_1(zI - A)\mathbf{N}_2 \tag{3.26c}$$

$$\mathbf{L} = -\mathbf{M}_1(zI - A)\mathbf{N}_2, \tag{3.26d}$$

which completes the proof. □

Proof of Lemma 7. Consider an internally stabilizing controller $\mathbf{K}$ with state space realization

$$\begin{aligned} z\boldsymbol{\xi} &= A_K\boldsymbol{\xi} + B_K\mathbf{y} \\ \mathbf{u} &= C_K\boldsymbol{\xi} + D_K\mathbf{y}. \end{aligned} \tag{3.27}$$

Combining (3.27) with the system equation (3.12), we obtain the closed loop dynamics

$$\begin{bmatrix} z\mathbf{x} \\ z\boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} A + B_2 D_k C_2 & B_2 C_k \\ B_k C_2 & A_k \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} I & B_2 D_k \\ 0 & B_k \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta_x} \\ \boldsymbol{\delta_y} \end{bmatrix}.$$

From the assumption that $\mathbf{K}$ is internally stabilizing, we know that the state matrix of the above equation is a stable matrix (Lemma 1 in Appendix 2.A). The system response achieved by $\mathbf{u} = \mathbf{K}\mathbf{y}$ is given by

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \left[\begin{array}{cc|cc} A + B_2 D_k C_2 & B_2 C_k & I & B_2 D_k \\ B_k C_2 & A_k & 0 & B_k \\ \hline I & 0 & 0 & 0 \\ D_k C_2 & C_k & 0 & D_k \end{array}\right], \tag{3.28}$$

which satisfies (3.15c). From (3.28), we have

$$\begin{aligned} (zI - A)\mathbf{R} - B_2\mathbf{M} &= z\mathbf{R} - A\mathbf{R} - B_2\mathbf{M} \\ &= \left[\begin{array}{cc|c} A + B_2 D_k C_2 & B_2 C_k & I \\ B_k C_2 & A_k & 0 \\ \hline A + B_2 D_k C_2 & B_2 C_k & I \end{array}\right] - \left[\begin{array}{cc|c} A + B_2 D_k C_2 & B_2 C_k & I \\ B_k C_2 & A_k & 0 \\ \hline A & 0 & 0 \end{array}\right] \\ &\quad - \left[\begin{array}{cc|c} A + B_2 D_k C_2 & B_2 C_k & I \\ B_k C_2 & A_k & 0 \\ \hline B_2 D_k C_2 & B_2 C_k & 0 \end{array}\right] \\ &= I. \end{aligned}$$

Using similar derivation, we can verify that (3.28) satisfies both (3.15a) and (3.15b) for arbitrary (A_k, B_k, C_k, D_k). This completes the proof. □

Proof of Lemma 8 (from (3.17) *to Table 3.1).* Recall that we have $\tilde{\mathbf{R}}^+ = z(I - z\mathbf{R})$, $\tilde{\mathbf{M}} = z\mathbf{M}$, and $\tilde{\mathbf{N}} = -z\mathbf{N}$. We can rewrite (3.17a), (3.17c), and (3.17d) as

$$(zI - A)\mathbf{x} = B_2\mathbf{u} + \boldsymbol{\delta_x} \tag{3.30a}$$

$$z\mathbf{R}\boldsymbol{\beta} = -\mathbf{N}\mathbf{y} + \frac{1}{z}\boldsymbol{\delta_\beta} \tag{3.30b}$$

$$z\mathbf{M}\boldsymbol{\beta} = \mathbf{u} - \mathbf{L}\mathbf{y} - \boldsymbol{\delta_u}. \tag{3.30c}$$

Multiplying $(zI - A)$ and B_2 on both sides of (3.30b) and (3.30c) respectively, we obtain

$$z(zI - A)\mathbf{R}\boldsymbol{\beta} = -(zI - A)\mathbf{N}\mathbf{y} + (I - \frac{1}{z}A)\boldsymbol{\delta_\beta} \tag{3.31a}$$

$$zB_2\mathbf{M}\boldsymbol{\beta} = B_2\mathbf{u} - B_2\mathbf{L}\mathbf{y} - B_2\boldsymbol{\delta_u}. \tag{3.31b}$$

Taking the difference between (3.31a) and (3.31b) then substituting the equality constraint (3.15a) in, we can express $\boldsymbol{\beta}$ as a function of $\mathbf{u}$ and the perturbations as

$$z\boldsymbol{\beta} = -B_2\mathbf{u} + (I - \frac{1}{z}A)\boldsymbol{\delta_\beta} + B_2\boldsymbol{\delta_u}. \tag{3.32}$$

Next, we multiply $\mathbf{R}$ on both sides of (3.30a) to obtain

$$\mathbf{R}(zI - A)\mathbf{x} = \mathbf{R}B_2\mathbf{u} + \mathbf{R}\boldsymbol{\delta_x}. \tag{3.33}$$

Substituting (3.17b) into (3.30b) and rearranging some terms, we have

$$\mathbf{N}C_2\mathbf{x} = -z\mathbf{R}\boldsymbol{\beta} - \mathbf{N}\boldsymbol{\delta_y} + \frac{1}{z}\boldsymbol{\delta_\beta}. \tag{3.34}$$

Taking the difference between (3.33) and (3.34) then substituting the equality constraint (3.15b) in, we have

$$\mathbf{x} = \mathbf{R}B_2\mathbf{u} + z\mathbf{R}\boldsymbol{\beta} + \mathbf{R}\boldsymbol{\delta_x} + \mathbf{N}\boldsymbol{\delta_y} - \frac{1}{z}\boldsymbol{\delta_\beta}. \tag{3.35}$$

Substituting (3.32) into (3.35), we can express $\mathbf{x}$ as a function of external perturbations as

$$\mathbf{x} = \mathbf{R}\boldsymbol{\delta_x} + \mathbf{N}\boldsymbol{\delta_y} + \mathbf{R}B_2\boldsymbol{\delta_u} + \left[\mathbf{R}(I - \frac{1}{z}A) - \frac{1}{z}I\right]\boldsymbol{\delta_\beta}. \tag{3.36}$$

We have $\mathbf{R}(I - \frac{1}{z}A) - \frac{1}{z}I = \frac{1}{z}\mathbf{N}C_2$ from (3.15b), and thus (3.36) can be rewritten as

$$\mathbf{x} = \mathbf{R}\boldsymbol{\delta_x} + \mathbf{N}\boldsymbol{\delta_y} + \mathbf{R}B_2\boldsymbol{\delta_u} + \frac{1}{z}\mathbf{N}C_2\boldsymbol{\delta_\beta}, \tag{3.37}$$

which is the first row of Table 3.1. The third row of Table 3.1 follows immediately by substituting (3.37) into (3.17b).

Then, we eliminate $B_2\mathbf{u}$ by adding (3.30a) and (3.32) as

$$z\boldsymbol{\beta} = -(zI - A)\mathbf{x} + \boldsymbol{\delta_x} + (I - \frac{1}{z}A)\boldsymbol{\delta_\beta} + B_2\boldsymbol{\delta_u}. \tag{3.38}$$

In addition, we substitute (3.17b) into (3.30c) and rearrange some terms to obtain

$$\mathbf{u} = z\mathbf{M}\boldsymbol{\beta} + \mathbf{L}C_2\mathbf{x} + \mathbf{L}\boldsymbol{\delta_y} + \boldsymbol{\delta_u}. \tag{3.39}$$

Multiplying $\mathbf{M}$ on both sides of (3.38) and substituting into (3.39), we have

$$\begin{aligned}\mathbf{u} &= \left[\mathbf{L}C_2 - \mathbf{M}(zI - A)\right]\mathbf{x} + \mathbf{M}\boldsymbol{\delta_x} + \mathbf{L}\boldsymbol{\delta_y} + (I + \mathbf{M}B_2)\boldsymbol{\delta_u} + \mathbf{M}(I - \frac{1}{z}A)\boldsymbol{\delta_\beta} \\ &= \mathbf{M}\boldsymbol{\delta_x} + \mathbf{L}\boldsymbol{\delta_y} + (I + \mathbf{M}B_2)\boldsymbol{\delta_u} + \mathbf{M}(I - \frac{1}{z}A)\boldsymbol{\delta_\beta}, \qquad (3.40)\end{aligned}$$

where the last equality comes from (3.15b). We note that $\mathbf{M}(I - \frac{1}{z}A) = \frac{1}{z}\mathbf{L}C_2$, and thus (3.40) can be written as

$$\mathbf{u} = \mathbf{M}\boldsymbol{\delta_x} + \mathbf{L}\boldsymbol{\delta_y} + (I + \mathbf{M}B_2)\boldsymbol{\delta_u} + \frac{1}{z}\mathbf{L}C_2\boldsymbol{\delta_\beta}, \tag{3.41}$$

which is the second row of Table 3.1. Substituting (3.41) into (3.32) yields the fourth row of Table 3.1, which completes the proof. □

Chapter 4

SYSTEM LEVEL SYNTHESIS PROBLEMS

An advantage of the system level parameterizations (SLPs) described in the previous chapter is that they allow us to impose additional constraints and objectives on the system response, e.g., the set constraint $\mathcal{S}$ and the objective functional g in (2.16). These constraints and objectives may be in the form of structural constraints on the response (and the corresponding controller implementation), or may capture a suitable measure of system performance. In this chapter, we introduce the System Level Objective (SLO) (e.g., the objective functional $g(\cdot)$ in (2.16)) and the System Level Constraint (SLC) (e.g., the set constraint $\mathcal{S}$ in (2.16)) for a system response. We then formulate the System Level Synthesis (SLS) problem (e.g., problem (2.16)) by combining SLO, SLP, and SLC into an optimization problem. We show that the SLS formulation characterizes the broadest known class of convex problems in constrained linear optimal control. As a special case, we recover all possible structured optimal control problems of the form (2.8) that admit a convex representation in the Youla domain.

This chapter begins with the general formulation of the SLS problem. We then provide a catalog of useful SLCs that can be naturally incorporated into the SLPs described in the previous chapter. Finally, we give some examples of convex SLS problems.

4.1 General Formulation

Let $g(\cdot)$ be a functional capturing a desired measure of the performance of the system response, and let $\mathcal{S}$ be a set. We pose the System Level Synthesis (SLS) problem as

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & g(\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}) && \text{(4.1a)} \\ \text{subject to} \quad & \text{(3.15a)} - \text{(3.15c)} && \text{(4.1b)} \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{S}. && \text{(4.1c)} \end{aligned}$$

The functional $g(\cdot)$ in (4.1a) is called a System Level Objective (SLO), and the set constraint $\mathcal{S}$ in (4.1c) is called a System Level Constraint (SLC). The affine subspace described by (4.1b) is called a System Level Parameterization (SLP), as introduced in the previous chapter. For $g(\cdot)$ a convex functional and $\mathcal{S}$ a convex set, the resulting SLS problem is a convex optimization problem. More generally, as long as the intersection of the SLC (4.1c) and SLP (4.1b) is convex, and the restriction of the SLO (4.1a) to the constraint (4.1b) - (4.1c) is convex, the resulting SLS problem is a convex optimization problem.

Remark 3. *Recall that the SLP for a state feedback problem is given by* (3.4a) - (3.4b). *Therefore, for a state feedback problem, the SLS problem can be simplified to*

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\textit{minimize}} \quad & g(\mathbf{R},\mathbf{M}) \\ \textit{subject to} \quad & (3.4\text{a}) - (3.4\text{b}) \\ & \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{S}. \end{aligned} \tag{4.2}$$

For a state estimation problem, the SLS problem can be simplified to

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{N}\}}{\textit{minimize}} \quad & g(\mathbf{R},\mathbf{N}) \\ \textit{subject to} \quad & (3.9\text{a}) - (3.9\text{b}) \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \end{bmatrix} \in \mathcal{S}. \end{aligned} \tag{4.3}$$

4.2 Convex System Level Constraints

We focus on the SLC (4.1c) in this section. We provide a catalog of useful convex SLCs that can be naturally incorporated into the SLPs described in the previous chapter. In particular, we show that QI subspace constraints are a special case of SLCs, and as such, we provide here a description of the largest known class of constrained stabilizing controllers that admit a convex parameterization.

4.2.1 Constraints on the Youla Parameter

We show that any constraint imposed on the Youla parameter can be translated into a SLC, and vice versa. In particular, if this constraint is convex, then so is the corresponding SLC.

Consider the following modification of the standard Youla parameterization, which characterizes a set of constrained internally stabilizing controllers $\mathbf{K}$ for a plant (3.11):

$$\mathbf{K} = \mathfrak{Y}(\mathbf{Q})[\mathfrak{X}(\mathbf{Q})]^{-1}, \ \mathbf{Q} \in \mathcal{Q} \cap \mathcal{RH}_\infty. \tag{4.4}$$

Here $\mathfrak{Y}(\mathbf{Q}) = \mathbf{Y}_r - \mathbf{U}_r\mathbf{Q}$ and $\mathfrak{X}(\mathbf{Q}) = \mathbf{X}_r - \mathbf{V}_r\mathbf{Q}$ are affine maps defined in terms of a doubly co-prime factorization of the plant (3.11) (c.f. Section 2.2 and §5.4 of [81]), and $\mathcal{Q}$ is an arbitrary set — if we take $\mathcal{Q} = \mathcal{RH}_\infty$, we recover the standard Youla parameterization. By appropriately varying the set $\mathcal{Q}$, one can then characterize all possible constrained internally stabilizing controllers,[1] and hence this formulation is as general as possible. We now show that an equivalent parameterization can be given in terms of a SLC.

Theorem 3. *The set of constrained internally stabilizing controllers described by* (4.4) *can be equivalently expressed as* $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$*, where the system response* $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ *lies in the set*

$$\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L} \,\big|\, \text{(3.15a) - (3.15c) } \textit{hold}, \ \mathbf{L} \in \mathfrak{M}(\mathcal{Q})\}, \tag{4.5}$$

for $\mathfrak{M}$ *an invertible affine map as defined in Section 2.2. Further, this parameterization is convex if and only if* $\mathcal{Q}$ *is convex.*

We note that when the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ is constrained to lie within a convex set, the transformation $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$ implies that the corresponding set of parameterized controllers $\mathbf{K}$ can be non-convex. Therefore it is possible to parameterize the set of internally stabilizing controllers $\mathbf{K}$ lying in certain non-convex sets $\mathcal{C}$ using the parameterization described in Theorem 3 and appropriately selected additional convex constraints on the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$.

In order to prove this result, we first need to understand the relationship between the controller $\mathbf{K}$, the Youla parameter $\mathbf{Q}$, and the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$.

Lemma 9. *Let* $\mathbf{L}$ *be defined as in* (3.14)*, and the invertible affine map* $\mathfrak{M}$ *be defined as in Section 2.2. We then have that*

$$\mathbf{L} = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1} = \mathfrak{M}(\mathbf{Q}). \tag{4.6}$$

Proof. Here we prove the identity from the definition of the system response $\mathbf{L}$. We provide an alternative algebraic proof in Appendix 4.A in the end of this chapter. From the equations $\mathbf{u} = \mathbf{K}\mathbf{y}$ and $\mathbf{y} = \mathbf{P}_{21}\mathbf{w} + \mathbf{P}_{22}\mathbf{u}$, we can eliminate $\mathbf{u}$ and express $\mathbf{y}$ as $\mathbf{y} = (I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\mathbf{w}$. We then have that

$$\mathbf{u} = \mathbf{K}\mathbf{y} = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\mathbf{w}. \tag{4.7}$$

Recall that we define $\boldsymbol{\delta}_x = B_1\mathbf{w}$ and $\boldsymbol{\delta}_y = D_{21}\mathbf{w}$. As a result, we have $\mathbf{P}_{21}\mathbf{w} = C_2(zI - A)^{-1}\boldsymbol{\delta}_x + \boldsymbol{\delta}_y$. Substituting this identity into (4.7) yields

$$\mathbf{u} = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}[C_2(zI - A)^{-1}\boldsymbol{\delta}_x + \boldsymbol{\delta}_y]. \tag{4.8}$$

[1] In particular, to ensure that $\mathbf{K} \in \mathcal{C}$, it suffices to enforce that $\mathfrak{Y}(\mathbf{Q})[\mathfrak{X}(\mathbf{Q})]^{-1} \in \mathcal{C}$.

By definition, $\mathbf{L}$ is the closed loop mapping from $\boldsymbol{\delta}_y$ to $\mathbf{u}$. Equation (4.8) then implies that $\mathbf{L} = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}$. From [54] and [28] (c.f. Section 2.2 - 2.3), we have $\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1} = \mathfrak{M}(\mathbf{Q})$, which completes the proof. □

Proof of Theorem 3. The equivalence between the parameterizations (4.4) and (4.5) is readily obtained from Lemma 9. As $\mathfrak{M}$ is an invertible affine mapping between $\mathbf{L}$ and $\mathbf{Q}$, any convex constraint imposed on the Youla parameter $\mathbf{Q}$ can be equivalently translated into a convex SLC imposed on $\mathbf{L}$, and vice versa. □

From Theorem 3, we note that any convex constraint imposed on the Youla parameter can be translated into a convex SLC imposed on the system response $\mathbf{L}$. Note that the system response $\mathbf{R}, \mathbf{M}$, and $\mathbf{N}$ are uniquely determined by the system response $\mathbf{L}$, as suggested by the SLP. Therefore, all the convex SLC can also be translated into a convex constraint imposed on the Youla parameter. The primary advantage of the SLC-based characterization (4.5) over the Youla-based characterization (4.4) is that we give a clear physical interpretation of the constraint imposed on the system response. Specifically, a set constraint imposed on the Youla parameter as in (4.4) in general does not have any specific physical meaning. On the contrary, if we lift the optimization variables from Youla parameter $\mathbf{Q}$ to four system response transfer matrices ($\mathbf{R}$, $\mathbf{M}$, $\mathbf{N}$, $\mathbf{L}$), then the constraint set described in (4.5) has a very clear physical meaning — these are constraints that shape the desired closed loop response from disturbances to state and control action. In addition, using the controller structure given in Figure 3.2, the SLC can also be interpreted as a structured constraint for controller implementation. This feature is the key to allow certain structured controller synthesis problems to be done in a convex manner, as will be shown in Section 4.2.8 and Chapters 5 - 7.

4.2.2 Quadratically Invariant Subspace Constraints

Recall that for a subspace C that is quadratically invariant with respect to a plant $\mathbf{P}_{22}$, the set of internally stabilizing controllers $\mathbf{K}$ that lie within the subspace C can be expressed as the set of stable transfer matrices $\mathbf{Q} \in \mathcal{RH}_\infty$ satisfying $\mathfrak{M}(\mathbf{Q}) \in C$, for $\mathfrak{M}$ and invertible affine map (c.f. Section 2.2). We therefore have the following corollary to Theorem 3.

Corollary 4. *Let C be a subspace constraint that is quadratically invariant with respect to $\mathbf{P}_{22}$. Then the set of internally stabilizing controllers satisfying $\mathbf{K} \in C$ can be parameterized as in Theorem 3 with $\mathbf{L} \in \mathfrak{M}(Q) = C$.*

Proof. From Lemma 9, we have $\mathbf{L} = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}$. Invoking Theorem 14 of [53], we have that $\mathbf{K} \in \mathcal{C}$ if and only if $\mathbf{L} = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1} \in \mathcal{C}$. The claim then follows immediately from Theorem 3. □

Corollary 4 gives a reinterpretation of the main result of the QI framework [27, 31, 33, 53, 54, 57]: a QI subspace constraint $\mathcal{C}$ for controller $\mathbf{K}$ can be equivalently translated into a subspace SLC given by $\mathbf{L} \in \mathcal{C}$. Thus we see that QI subspace constraints are a special case of SLCs.

Note that Corollary 4 holds true for stable and unstable plants $\mathbf{P}$. Therefore, in order to parameterize the set of internally stabilizing controllers lying in $\mathcal{C}$, we do not need to assume the existence of an initial strongly stabilizing controller as in [53] nor do we need to perform a doubly co-prime factorization as in [54].

Finally, we note that in [33] and [31], the authors show that quadratic invariance is necessary for a subspace constraint $\mathcal{C}$ on the controller $\mathbf{K}$ to be enforceable via a convex constraint on the Youla parameter $\mathbf{Q}$. However, when $\mathcal{C}$ is not a subspace constraint, no general methods exist to determine whether the set of internally stabilizing controllers lying in $\mathcal{C}$ admits a convex representation. In contrast, determining the convexity of a SLC is trivial.

4.2.3 System Performance Constraints

Let $g(\cdot)$ be a functional of the system response — it then follows that all internally stabilizing controllers satisfying a performance level, as specified by a scalar γ, are given by transfer matrices $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ satisfying the conditions of Theorem 2 and the SLC

$$g(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}) \le \gamma. \tag{4.9}$$

Further, recall that the sublevel set of a convex functional is a convex set, and hence if g is convex, then so is the SLC (4.9). A particularly useful choice of convex functional is

$$g(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}) = \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} + D_{11} \right\|, \tag{4.10}$$

for a system norm $\|\cdot\|$, which is equivalent to the objective function of the decentralized optimal control problem (2.8). Thus by imposing several performance SLCs (4.10) with different choices of norm, one can naturally formulate multi-objective optimal control problems, e.g., mixed $\mathcal{H}_2/\mathcal{H}_\infty$ optimal control and mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control.

Remark 4. *For a continuous time system with the system norm* $\|\cdot\|$ *in* (4.10) *chosen to be the* $\mathcal{H}_2$ *norm, the closed loop transfer matrix in* (4.10) *needs to be strictly proper, i.e.,* $D_{12}L[0]D_{21} + D_{11} = 0$*.*

4.2.4 Controller Robustness Constraints

Suppose that the controller is to be implemented using limited hardware, thus introducing non-negligible quantization (or other errors) to the internally computed signals: this can be modeled via an internal additive noise $\boldsymbol{\delta_\beta}$ in the controller structure (c.f., Figure 3.2). In this case, we may wish to design a controller that further limits the effects of these perturbations on the system: to do so, we can impose a performance SLC on the closed loop transfer matrices specified in the rightmost column of Table 3.1. Note that it is not clear how to impose such controller robustness constraints using the Youla parameterization.

For another example, suppose that the sub-controllers u_i and u_j in different locations need to exchange their information through some noisy channel. If we model the communication noise as additive noise on the transmitted signal, then we can analyze the system responses of these perturbations using the structure given in Figure 3.2, and incorporate their effects in the controller design stage. This shows the usefulness of our *system level* approach for controller design with imperfect hardware.

4.2.5 Controller Architecture Constraints

The controller implementation (3.16) also allows us to naturally control the number of actuators and sensors used by a controller — this can be useful when designing controllers for large-scale systems that use a limited number of hardware resources (c.f., Section 4.3.3). In particular, assume that implementation (3.16) parameterizing stabilizing controllers that use all possible actuators and sensors. It then suffices to constrain the number of non-zero rows of the transfer matrix $[\mathbf{M} \quad \mathbf{L}]$ to limit the number of actuators used by the controller, and similarly, the number of non-zero columns of the transfer matrix $[\mathbf{N}^\top \quad \mathbf{L}^\top]^\top$ to limit the number of sensors used by the controller. As stated, these constraints are non-convex, but recently proposed convex relaxations [40, 41] can be used in their stead to impose convex SLCs on the controller architecture. We will explore this topic in more details in Section 4.3.3 and Chapter 7.

4.2.6 Positivity Constraints and Desired System Response

It has recently been observed that (internally) positive systems are amenable to efficient analysis and synthesis techniques (c.f., [50] and the references therein). Therefore it may be desirable to synthesize a controller that either preserves or enforces positivity of the resulting closed loop system. We can enforce this condition via the SLC that the elements

$$\left\{ \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} R[t] & N[t] \\ M[t] & L[t] \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \right\}_{t=1}^{\infty}$$

and the matrix $(D_{12}L[0]D_{21} + D_{11})$ are all element-wise nonnegative matrices. This SLC is easily seen to be convex.

For a similar example, suppose that we want to find a controller to reject a step disturbance $\mathbf{w}$ on the regulated output $\bar{\mathbf{z}}$. In this case, we enforce a convex SLC as

$$D_{11} + D_{12}L[0]D_{21} + \sum_{t=1}^{\infty} \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} R[t] & N[t] \\ M[t] & L[t] \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} = 0.$$

4.2.7 FIR Constraints

Given the parameterization of stabilizing controllers of Theorem 2, it is trivial to enforce that a system response be FIR with horizon T via the following SLC

$$\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L} \in \mathcal{F}_T. \tag{4.11}$$

We argue that imposing a FIR SLC is beneficial in the following ways:

(a) The closed loop response to an impulse disturbance is FIR of horizon T, where T can be set by the control designer. As such, the settling time of the system can be accurately tuned.

(b) The controller achieving the desired system response can be implemented using the FIR filter banks $\tilde{\mathbf{R}}^+, \tilde{\mathbf{M}}, \tilde{\mathbf{N}}, \mathbf{L} \in \mathcal{F}_T$, as illustrated in Figure 3.2. This simplicity of implementation is extremely helpful when applying these methods in practice.

(c) When a FIR SLC is imposed, the resulting set of stable achievable system responses and corresponding controllers admit a finite dimensional representation — specifically, the constraints specified in Theorem 2 only need to be applied to the impulse response elements $\{R[t], M[t], N[t], L[t]\}_{t=0}^{T}$.

Remark 5. *It should be noted that the computational benefits claimed above hold only for discrete time systems. For continuous time systems, a FIR transfer matrix is still an infinite dimensional object, and hence the resulting parameterizations and constraints are in general infinite dimensional as well.*

We defer discussing how to select the FIR horizon T in the context of an optimal control problem in Chapters 5 - 6, and present here instead connections between the feasibility of the SLC (4.11) and the controllability, reachability, and observability of a system (3.11).

Recall that the feasibility of condition (3.4) provides an alternative characterization of the stabilizability of the system (A, B_2). We now give an alternative characterization of the controllability of (A, B_2), in terms of condition (3.4) and a FIR SLC.

Before proceeding, we recall the notions of *controllability* and *reachability* in the sense of Kalman [25]. Given a positive integer T, we say that a system (A, B_2) with dynamics $x[t+1] = Ax[t] + B_2u[t]$ is T-step controllable if we can select a sequence of control actions $\{u[t]\}_{t=0}^{T-1}$ to drive the state $x[T]$ to 0 from any initial condition $x[0]$. If we can drive the state $x[T]$ to an arbitrary value from any initial condition $x[0]$, then the system is said to be T-step reachable. A system (A, B_2) is said to be controllable (reachable) if it is T-step controllable (reachable) for some finite T. Therefore, to check whether a system (A, B_2) is T-step controllable, it suffices to verify that the impulse responses from $\boldsymbol{\delta_x}$ to $(\mathbf{x}, \mathbf{u})$ in (3.2) are FIR of horizon T. It is therefore clear that a pair (A, B_2) is T-step controllable if and only if the following equations are feasible for some finite T:

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \tag{4.12a}$$

$$\mathbf{R}, \mathbf{M} \in \mathcal{F}_T \cap \frac{1}{z}\mathcal{RH}_\infty. \tag{4.12b}$$

The conditions for reachability are slightly more restrictive than those for controllability.

Lemma 10. *The pair* (A, B_2) *is reachable if and only if* $[A \quad B_2]$ *is full rank and* (4.12) *is feasible for a finite* T.

Proof. Assume that (4.12) is feasible. As $\mathbf{R}$ and $\mathbf{M}$ are FIRs, the poles of $\mathbf{R}$ and $\mathbf{M}$ must be located at the origin $z = 0$. From (4.12a), we observe that $\begin{bmatrix} zI - A & -B_2 \end{bmatrix}$

is right invertible in the region where $\mathbf{R}(z)$ and $\mathbf{M}(z)$ do not have poles, with $\begin{bmatrix}\mathbf{R}^\top & \mathbf{M}^\top\end{bmatrix}^\top$ being its right inverse. This means that $\begin{bmatrix}zI - A & -B_2\end{bmatrix}$ has full row rank for all $|z| > 0$. Combining with the assumption that $[A \quad B_2]$ has full row rank, we can use the PBH test [15] to show that (A, B_2) is reachable.

For the opposite, we apply the PBH test to show that $[A \quad B_2]$ must have full row rank. From the definition of reachability, it is straightforward to construct a feasible solution to (4.12). □

A similar argument leads to an alternative characterization of the observability of a pair (A, C_2): it suffices to check the feasibility of conditions (3.9) with the added FIR SLC that $\mathbf{R}, \mathbf{N} \in \mathcal{F}_T$ for some T.

Finally we note that when the triple (A, B_2, C_2) is controllable and observable, we can use equation (3.26) in Appendix 3.A to construct a FIR system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ that satisfies the conditions (3.15) of Theorem 2 — hence, the controllability and observability of a triple (A, B_2, C_2) is a necessary and sufficient condition for the existence of a FIR system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ for some horizon T.

4.2.8 Subspace and Sparsity Constraints

Let $\mathcal{L}$ be a subspace of $\mathcal{RH}_\infty$. We can then parameterize all stable achievable system responses that lie in this subspace by adding the following SLC to the parameterization of Theorem 2:

$$\begin{bmatrix}\mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L}\end{bmatrix} \in \mathcal{L}. \tag{4.13}$$

Of particular interest are subspaces $\mathcal{L}$ that define transfer matrices of sparse support. An immediate benefit of enforcing such sparsity constraints on the system response is that implementing the resulting controller (3.16) can be done in a *localized way*, i.e., each controller state β_i and control action u_i can be computed using a local subset (as defined by the support of the system response) of the global controller state β and sensor measurements y. For this reason, we refer to the constraint (4.13) as a *localized* SLC when it defines a subspace with sparse support. Further, as we discuss in details in Chapters 5 - 7, such localized constraints also allow for the resulting system response to be computed in a localized way if the SLS problem (4.1) satisfies certain technical condition. In this case, the global computation decomposes naturally into decoupled subproblems that depend only on local submatrices of the state-space representation (2.1). Clearly, both of these features are extremely desirable when computing controllers for large-scale systems.

Selecting an appropriate (feasible) localized SLC, as defined by the subspace $\mathcal{L}$, is a subtle task: it depends on an interplay between actuator and sensor density, information exchange delay and disturbance propagation delay. For instance, in our recent paper [70], we present a method that allows for the joint design of an actuator architecture and corresponding feasible localized SLC. We generalize these methods to the output feedback setting in [72], where we show that actuation and sensing architectures, as well as feasible localized SLCs, can be co-designed using convex programming. This class of problem will be discussed in details in Section 5.3 and in Chapter 7.

We informally illustrate some of these concepts in the example below, which builds on the motivating example Example 2 of Section 2.4.

Example 3. *Consider the optimal control problem* (2.10) *in Example 1, and assume that there is unit measurement delay, i.e., the control action at time t can be expressed as $u[t] = f(x[0 : t-1])$ for some function f. The optimal system response for this delayed centralized optimal control problem is given by*

$$\mathbf{R} = \frac{1}{z}I + \frac{1}{z^2}A, \quad \mathbf{M} = -\frac{1}{z^2}A^2, \tag{4.14}$$

with the optimal state-feedback controller given by $\mathbf{K} = -A^2(zI + A)^{-1}$. Note that the support of the system response elements $\mathbf{R}$ and $\mathbf{M}$ is defined by the support of A and A^2, respectively. As a concrete example, let the state matrix A be tridiagonal (hence its support defines the adjacency matrix of a chain): it then follows that the transfer matrix $\mathbf{R}$ is tridiagonal and $\mathbf{M}$ is pentadiagonal.

Using Lemma 5, the controller achieving the desired system response can be implemented using the FIR transfer matrices $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{M}}$ (which have the same support as $\mathbf{R}$ and $\mathbf{M}$), via the controller structure in Figure 3.1. This implementation is localized as each node i needs only collect its first and second neighbors' estimated disturbances $\hat{w}_j$ to compute its control action u_i and reference state value $\hat{x}_i$. In contrast, notice that implementing the control policy $\mathbf{u} = \mathbf{K}\mathbf{y}$, with controller $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1} = -A^2(zI + A)^{-1}$, requires each node to collect measurements y from every other node in the system, as $\mathbf{K}$ is dense.

As we highlighted in Example 1 and the discussion of Section 2.4, imposing sparsity constraints on the controller $\mathbf{K}$ violates the conditions of quadratic invariance when *the underlying system is strongly connected, and hence cannot be done using convex constraints on the Youla parameter. In contrast, imposing a localized SLC on the system response elements* $\mathbf{R}$ *and* $\mathbf{M}$ *is always convex, and in this case allows us to recover the centralized optimal solution.*

4.2.9 Intersections of SLCs and Spatiotemporal Constraints

Another major benefit of SLCs is that several such constraints can be imposed on the system response at once. Further, as convex sets are closed under intersection, convex SLCs are also closed under intersection. To illustrate the usefulness of this property, consider the intersection of a QI subspace SLC, a FIR SLC, and a localized SLC. The resulting SLC can be interpreted as enforcing a spatiotemporal constraint on the system response and its corresponding controller, as we explain using the chain example previously described.

Figure 4.1 shows a diagram of the system response to a particular disturbance $(\boldsymbol{\delta_x})_i$. In this figure, the vertical axis denotes the spatial coordinate of a state in the chain, and the horizontal axis denotes time: hence we refer to this figure as a space-time diagram. Depicted are the three components of the spatiotemporal constraint, namely the communication delay imposed on the controller via the QI subspace SLC, the deadbeat response of the system to the disturbance imposed by the FIR SLC, and the localized region affected by the disturbance $(\boldsymbol{\delta_x})_i$ imposed by the localized SLC.

When the effect of each disturbance $(\boldsymbol{\delta_x})_i$ can be localized within such a spatiotemporal SLC, the system is said to be *localizable* (c.f., [68, 71] and Section 5.3). Recall that the controllability and the observability of a system is determined by the existence of an FIR system response. Similarly, the localizability of a system is determined by the existence of a system response satisfying a spatiotemporal SLC — in this sense, localizability can be viewed as a natural generalization of controllability and observability to the spatiotemporal domain.

4.3 Convex System Level Synthesis Problems

In this section, we give some examples of optimal control problems that can be cast as a SLS problem.

4.3.1 Distributed Optimal Control

Here we show that by combining an appropriate SLC with the SLP described in Theorem 2, we recover the distributed optimal control formulation (2.8) as a special

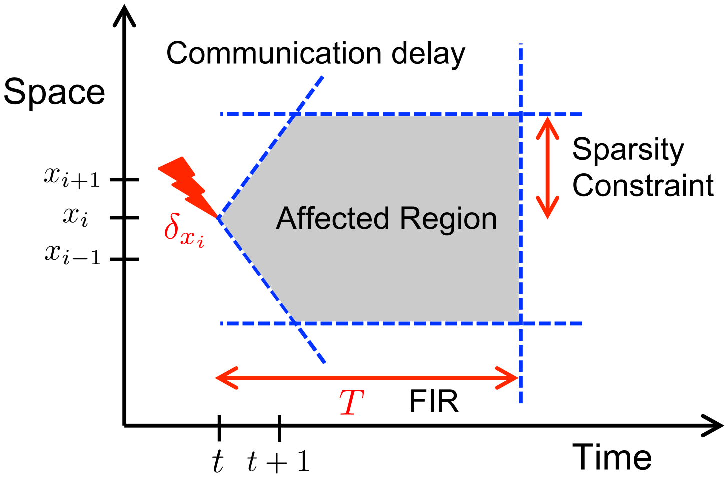


Figure 4.1: Space time diagram for a single disturbance striking the chain described in Example 3.

case of the SLS problem (4.1). Recall that the objective function in (2.8) is specified by a suitably chosen system norm measuring the size of the closed loop transfer matrix from the external disturbance $\mathbf{w}$ to the regulated output $\bar{\mathbf{z}}$. Therefore it suffices to select the SLO to be as described in equation (4.10), and to select the SLC constraint set $\mathcal{S}$ as described in Corollary 4. The resulting SLS problem

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} + D_{11} \right\| \\ \text{subject to} \quad & (3.15\text{a}) - (3.15\text{c}) \\ & \mathbf{L} \in \mathcal{C} \end{aligned} \tag{4.15}$$

is then equivalent to the distributed optimal control problem (2.8) when the subspace $\mathcal{C}$ is QI with respect to the plant $\mathbf{P}_{22}$. Note that the optimization problem (4.15) is convex as long as the set $\mathcal{C}$ imposed on the system response $\mathbf{L}$ is convex. Therefore, we can generalize the set $\mathcal{C}$ from a QI subspace constraint to arbitrary convex constraint. In particular, all decentralized optimal control problems that can be formulated as convex optimization problems in the Youla domain are special cases of the SLS problem (4.1).

4.3.2 Localized LQG Control

In [68, 71] we posed and solved a localized LQG (LLQG) optimal control problem. It can be recovered as a special case of the SLS problem (4.1) by selecting the SLO to be of the form (4.10) (with the system norm $\| \cdot \|$ chosen to be the $\mathcal{H}_2$ norm), and selecting the constraint set $\mathcal{S}$ to be a spatiotemporal SLC. In the case of a state feedback problem, which is known as the localized LQR (LLQR) problem [71], the resulting SLS formulation is of the form

$$\begin{array}{ll} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} & \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ \text{subject to} & (3.4a) - (3.4b) \\ & \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T, \end{array} \tag{4.16}$$

for $\mathcal{C}$ a QI subspace SLC, $\mathcal{L}$ a localized SLC, and $\mathcal{F}_T$ a FIR SLC. In the case of an output feedback LLQG, the resulting SLS formulation is given by

$$\begin{array}{ll} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} & \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} + D_{11} \right\| \\ \text{subject to} & (3.15a) - (3.15c) \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T. \end{array} \tag{4.17}$$

The observation that we make in [71] (and extend to the output feedback setting in [68]), is that the LLQR problem (4.16) can be decomposed into a set of independent subproblems solving for the columns $\mathbf{R}_i$ and $\mathbf{M}_i$ of the transfer matrices $\mathbf{R}$ and $\mathbf{M}$ — as these problems are independent, they can be solved in parallel. Further, the localized constraint $\mathcal{L}$ restricts each sub-problem to a local subset of the system model and states, as specified by the nonzero components of the corresponding column of the transfer matrices $\mathbf{R}$ and $\mathbf{M}$ (e.g., as was described in Example 1), allowing each of these subproblems to be expressed in terms of optimization variables (and corresponding sub-matrices of the state-space realization (3.4)) that are of significantly smaller dimension than the global system response $\{\mathbf{R}, \mathbf{M}\}$. Thus for a given feasible spatiotemporal SLC, the LLQR problem (4.16) can be solved for arbitrarily large-scale systems, assuming that each sub-controller can solve its corresponding subproblem in parallel. This argument extends to output feedback LLQG problem as well. We will discuss the LLQR and the LLQG problem in details in Chapter 5 and 6, respectively. We also show how to co-design an actuation architecture and feasible corresponding spatiotemporal constraint in Section 5.3, and so the assumption of a feasible spatiotemporal constraint is a reasonable one.

Finally in Chapter 7, we generalize all of these concepts to the class of convex localized separable SLS (CLS-SLS) problems. Specifically, we show that appropriate notions of separability for SLOs and SLCs can be defined which allow for optimal controllers to be synthesized and implemented with order constant complexity (assuming parallel computation is available for each subproblem) relative to the global system size.

4.3.3 Regularization for Design and SLS

At this point our motivation has been to extend controller synthesis methods to large-scale systems. An equally important task when designing control systems in such settings is the design of the controller architecture itself, i.e., the placement of actuators, sensors and communication links between them. As will be discussed in Section 5.3, sufficiently dense actuation, sensing and communication architectures are necessary for a localized optimal control problem to be feasible. More generally, there is a tradeoff between closed loop performance and architectural cost, as denser controller architectures lead to better closed loop performance.

The regularization for design (RFD) framework was formulated to explore this tradeoff using convex programming by augmenting the objective function with a suitable convex regularizer that penalizes the use of actuators, sensors, and communication links. The original RFD formulation allowed for controller architecture co-design in the Youla domain by exploiting QI properties of desirable architectures [38–41], but was later ported to the LLQR framework [70]. Thus to integrate RFD with the system level approach, it suffices to add a suitable regularizer, as mentioned in Section 4.2.5 and described in [40, 70], to the objective function of the SLS problem (4.1).

We will formulate the sensor actuator regularized LLQG problem in Chapter 7 to co-design a localized optimal controller and its sensing and actuation architecture. Further, we will show that the sensor actuator regularized LLQG problem belong to the class of CLS-SLS problems, and thus can be solved with constant parallel complexity for arbitrary large-scale systems.

4.3.4 Computational Complexity and Non-convex Optimization

A final advantage of the SLS problem (4.1) is that it is transparent to determine the computational complexity of the optimization problem. Specifically, the complexity of solving (4.1) is determined by the type of the SLO and the characterization of the intersection of the SLC and SLP. Further, when the SLS problem is non-convex, the direct nature of the formulation makes it straightforward to determine suitable convex relaxations or non-convex optimization techniques for the problem. In contrast, as discussed in [33], no general method exists to determine the computational complexity of the decentralized optimal control problem (2.8) for a general constraint set C.

APPENDIX

4.A Alternative Proof of Lemma 9

Algebraic Proof of Lemma 9. From the definition of $\mathbf{R}$ in (3.14), we have

$$(zI - A - B_2\mathbf{K}C_2)\mathbf{R} = I.$$

Moving $B_2\mathbf{K}C_2\mathbf{R}$ to the right-hand-side and multiplying $(zI - A)^{-1}$ on both sides, it is straightforward to derive that

$$\mathbf{R} = (zI - A)^{-1}(I + B_2\mathbf{K}C_2\mathbf{R}). \tag{4.18}$$

Then, we have

$$\begin{aligned} C_2\mathbf{R} &= C_2(zI - A)^{-1} + C_2(zI - A)^{-1}B_2\mathbf{K}C_2\mathbf{R} \\ &= C_2(zI - A)^{-1} + \mathbf{P_{22}}\mathbf{K}C_2\mathbf{R}. \end{aligned}$$

Moving $\mathbf{P_{22}}\mathbf{K}C_2\mathbf{R}$ to the left-hand-side and multiplying $(I - \mathbf{P_{22}}\mathbf{K})^{-1}$ on both sides, we have

$$C_2\mathbf{R} = (I - \mathbf{P_{22}}\mathbf{K})^{-1}C_2(zI - A)^{-1}. \tag{4.19}$$

Finally,

$$\begin{aligned} \mathbf{L} &= \mathbf{K} + \mathbf{K}C_2\mathbf{R}B_2\mathbf{K} \\ &= \mathbf{K} + \mathbf{K}(I - \mathbf{P_{22}}\mathbf{K})^{-1}C_2(zI - A)^{-1}B_2\mathbf{K} \\ &= \mathbf{K} + \mathbf{K}(I - \mathbf{P_{22}}\mathbf{K})^{-1}\mathbf{P_{22}}\mathbf{K} \\ &= \mathbf{K}(I - \mathbf{P_{22}}\mathbf{K})^{-1}, \end{aligned}$$

which completes the proof. □

Chapter 5

LOCALIZED LINEAR QUADRATIC REGULATOR

In the rest of this dissertation, we move our focus to the class of convex localized separable system level synthesis (CLS-SLS) problems. The class of CLS-SLS problems can not only be solved in a convex manner, but also be *solved* in a localized and scalable way, with $O(1)$ parallel computational complexity compared to the size of the global network. In theory, we can solve a CLS-SLS problem to synthesize localized optimal controller for systems with arbitrary large-scale if parallel computation is available. In addition to the scalability of controller synthesis, the controller achieving the desired system response can also be implemented in a localized and scalable way — each sub-controller in the network only needs to exchange information with $O(1)$ numbers of sub-controllers to compute its control action. The CLS-SLS framework is therefore extremely favorable for large-scale applications such as power grid, transportation network, and computer networking.

In this chapter, we introduce the localized linear quadratic regulator (LLQR) problem, which is a simple example of a CLS-SLS problem. We show that the LLQR controller can achieve similar transient optimal performance compared to the traditional centralized LQR one, but far more superior in terms of the scalability of controller synthesis and implementation. In particular, we demonstrate the LLQR controller on a randomized heterogeneous power network example with 51200 states in Section 5.7. We show that the LLQR controller can be synthesized in 23 minutes using a personal computer. In contrast, the theoretical computation time for the centralized LQR on the same computer is more than 200 days, and the distributed LQR is intractable.

The purpose of this chapter is to illustrate the intuition of our method, so we will keep our system model as simple as possible. We will generalize the discussion to output feedback localized linear quadratic Gaussian (LLQG) in Chapter 6, and to the general class of CLS-SLS problems in Chapter 7.

The rest of this chapter is organized as follows. We introduce the interconnected system model, explain the challenges of controller design for large-scale systems, and give the LLQR problem formulation in Section 5.1. In Section 5.2, we show that the LLQR controller can be synthesized *and* implemented in a localized and scalable way for a *localizable* system. We discuss the relations between the localizability of the system, the delay pattern of the communication network, and the actuation architecture of the controllers in Section 5.3. We also propose a regularization-based method [70] to co-design a LLQR controller and its actuation architecture in the same section. In Section 5.4, we introduce the class of *nearly localizable systems*, and discuss the robustness of applying a localized controller on a nearly localizable system. In Section 5.5, we quantify the sub-optimality of the LLQR controller and propose an adaptive constraint update algorithm to design a LLQR controller with performance guarantee [66]. In Section 5.6, we consider the localized distributed Kalman filter (LDKF) problems for large-scale state estimation, and show that LDKF problems can be solved using the LLQR algorithm [74]. Finally, we perform some numerical simulation to illustrate the effectiveness of the LLQR method in Section 5.7.

5.1 Problem Statement

We begin by introducing the interconnected system model that we consider in this chapter. We then explain why the traditional centralized and distributed LQR methods do not scale to large systems. We end this section by formulating the LLQR problem as a localized SLS problem.

5.1.1 Interconnected System Model

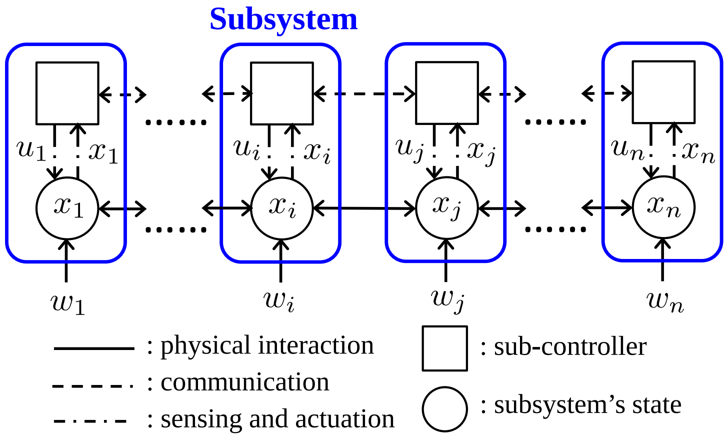


Figure 5.1: An Example of Interconnected System

Consider n dynamically coupled discrete time linear time invariant (LTI) subsystems that interact with each other according to an interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V} = \{1, \dots, n\}$ denotes the set of subsystems. We denote x_i and u_i the state vector and control action of subsystem i. The set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes the physical interaction between these subsystems — an edge (i, j) is in $\mathcal{E}$ if and only if the state x_j of subsystem j directly affects the state x_i of subsystem i. The dynamics of subsystem i is assumed to be given by

$$x_i[t+1] = A_{ii}x_i[t] + \sum_{j\in\mathcal{N}_i} A_{ij}x_j[t] + B_{ii}u_i[t] + w_i[t], \tag{5.1}$$

where $\mathcal{N}_i = \{j|(i, j) \in \mathcal{E}\}$ is the (incoming) neighbor set of subsystem i, A_{ii}, A_{ij}, B_{ii} some matrices with compatible dimension, and w_i the process disturbance of subsystem i. Figure 5.1 shows an example of such interconnected distributed system — each subsystem i has a sub-controller that takes the locally available state measurement x_i, exchanges information with some other sub-controllers through a communication network, and generates the control action u_i to control the state x_i of the physical system.

Define $x = [x_1 \ldots x_n]^\top$, $u = [u_1 \ldots u_n]^\top$, and $w = [w_1 \ldots w_n]^\top$ the stacked vectors of the subsystem states, controls, and process disturbances, respectively. The n interconnected system models (5.1) can be combined into a global system model given by

$$x[t+1] = Ax[t] + B_2u[t] + w[t], \tag{5.2}$$

with

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix} \quad \text{and} \quad B_2 = \begin{bmatrix} B_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{nn} \end{bmatrix}.$$

Note that the topology of the graph $\mathcal{G}$ is encoded in the sparsity pattern of the global system matrices A. In particular, the block sparsity pattern of A is equal to the adjacency matrix of its underlying graph $\mathcal{G}$.

Remark 6. *When each x_i and u_i in* (5.1) *are scalar variables, we call the model* (5.1) - (5.2) *a* scalar subsystem model.

Remark 7. *Note that the matrix B_2 in* (5.2) *is diagonal. If the ith subsystem does not have an actuator that can directly alter its state, we simply assign $B_{ii} = 0$. We will lift these assumptions in Chapter 7 to consider the most general output feedback interconnected system model with arbitrary sparsity pattern.*

We assume that the pair (A, B_2) in (5.2) is controllable. In addition, the disturbances w are drawn i.i.d. from a zero mean unit covariance Gaussian distribution, i.e., we have $\mathbb{E}(w[k]) = 0$ and $\mathbb{E}(w[i]w[j]^\top) = \delta_{ij}I$ for all i, j, k, where δ_{ij} is the Kronecker delta and $\mathbb{E}(\cdot)$ is the expectation operator. The objective is to find a control strategy, which is a mapping from state measurement $\mathbf{x}$ to control action $\mathbf{u}$, to minimize the expected value of the average quadratic cost

$$\mathbb{E}\Big(\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix}^\top \begin{bmatrix} C_1^\top C_1 & C_1^\top D_{12} \\ D_{12}^\top C_1 & D_{12}^\top D_{12} \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} \Big) \tag{5.3}$$

for some cost matrices (C_1, D_{12}). The traditional infinite horizon stochastic LQR problem can then be formulated as

$$\begin{aligned} \underset{\{x[k],u[k]\}_{k=1}^{\infty}}{\text{minimize}} \quad & \mathbb{E}\Big(\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix}^\top \begin{bmatrix} C_1^\top C_1 & C_1^\top D_{12} \\ D_{12}^\top C_1 & D_{12}^\top D_{12} \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} \Big) \\ \text{subject to} \quad & x[k+1] = Ax[k] + B_2 u[k] + w[k] \\ & x[0] = 0, \quad u[0] = 0 \\ & \mathbb{E}(w[k]) = 0, \quad \mathbb{E}(w[i]w[j]^\top) = \delta_{ij} I. \end{aligned} \tag{5.4}$$

5.1.2 Challenges on Scalability

Traditionally, the solution of the stochastic LQR problem (5.4) is obtained by solving a discrete time algebraic Riccati equation (DARE). The optimal solution is given by a static feedback $u[k] = Kx[k]$, where the gain matrix K can be found by solving the DARE. The solution to the LQR problem is one of the most important and elegant result in modern optimal control theory [81] due to the following reasons: (i) the optimal solution is proven to be *linear* (the control action u is a linear function of the state measurement x), (ii) the optimal solution is *static* (the control action $u[k]$ at time k depends only on the measurement $x[k]$ at time k), and (iii) the controller K can be obtained by solving a *convex* program in polynomial time. In addition to its simplicity, the LQR method has proven to be useful in extremely diverse applications [81].

However, there are some limitations of the LQR method for large-scale systems:

1. **Communication delays:** The LQR gain matrix K is generally dense even when the system matrices (A, B_2) that specify the system dynamics (5.2) are sparse. This means that the measurements from all states need to be shared *instantaneously*, which requires infinite (or impractically fast) communication speed.

2. **Scalability of controller implementation:** A dense LQR gain also implies that the measurements from all sensor need to be collected by *every sub-controller in the network*, which is not scalable to implement.

3. **Scalability of controller synthesis:** To compute the LQR gain, one need to solve a large-scale DARE. The complexity of solving the DARE is $O(n^3)$, where n is the dimension of the matrix A. Even though the DARE can be solved in polynomial time, the complexity blows up quickly for a large n.

4. **Scalability of controller re-synthesis subject to model change:** When a few entries of the global plant model (A, B_2) change, one needs to recompute the solution of (5.4) to resynthesize the global LQR optimal controller. This is not scalable for incremental design when the physical system expands.

To solve the above limitations, a common approach is to incorporate structured constraint on the controller, which leads to a constrained optimal control problem as in (2.8). In particular, the stochastic LQR problem (5.4) is a special case of a $\mathcal{H}_2$ optimal control problem, with plant model given by

$$\mathbf{P} = \left[\begin{array}{c|cc} A & I & B_2 \\ \hline C_1 & 0 & D_{12} \\ I & 0 & 0 \end{array}\right] = \begin{bmatrix} \mathbf{P_{11}} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}.$$

A structured $\mathcal{H}_2$ optimal control is given by (cf., Section 2.3)

$$\begin{aligned} \underset{\mathbf{K}}{\text{minimize}} \quad & ||\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}||_{\mathcal{H}_2}^2 && (5.5a) \\ \text{subject to} \quad & \mathbf{K} \text{ internally stabilizes } \mathbf{P} && (5.5b) \\ & \mathbf{K} \in C, && (5.5c) \end{aligned}$$

where the subspace constraint $\mathbf{K} \in C$ enforces information sharing constraints between the sub-controllers.[1] As mentioned in Chapter 2, it was shown in [53] that if the subspace constraint set C is quadratically invariant (QI) with respect to $\mathbf{P}_{22}$ then the optimal control problem admits a convex reformulation. Loosely speaking, this condition requires that sub-controllers be able to share information with each other at least as quickly as their control actions propagate through the plant [52]. The QI framework therefore provides a tractable way to address the challenge of communication delays mentioned above.

However, the distributed optimal control framework based on QI has less emphasis on the challenges of scalability mentioned above. We note that the implementation complexity of controller $\mathbf{K}$ is determined by the densest row $\mathbf{K}_i$ of $\mathbf{K}$. Specifically, if

[1] Note that the controller $\mathbf{K}$ here does not need to be static. Therefore, we use a boldface Latin letter to emphasize that $\mathbf{K}$ is a transfer matrix.

we consider the control action taken at subsystem i, which is specified by $\mathbf{u}_i = \mathbf{K}_i\mathbf{x}$, then if $\mathbf{K}_i$ is completely dense then subsystem i must collect measurements from every other subsystem $j \in \mathcal{V}$. In order to design a controller that is scalable to implement, a natural solution is to impose sparsity constraints on the controller $\mathbf{K}$ such that each row only has a small number of nonzero terms — in this way each subsystem i only needs to collect a small number of measurements to compute its control action. Unfortunately, this naive approach fails if the underlying topology of $\mathcal{G}$ of the networked system is strongly connected. Specifically, when the topology of $\mathcal{G}$ is strongly connected, any sparse constraint set $\mathcal{C}$ is not QI (cf. Example 1 in Section 2.4), which means that the constrained optimal control problem is always non-convex. Although recent methods based on convex relaxations [19] can be used to solve certain cases of the non-convex optimal control problem (5.5) with sparse constraint set $\mathcal{C}$, the underlying synthesis optimization problem is itself still large-scale and does not admit a scalable reformulation. The need to address scalability, both in the synthesis and implementation of a controller, is the driving motivation of the LLQR framework.

5.1.3 Localized LQR as a SLS Problem

We first reformulate the stochastic LQR problem (5.4) into the form of a SLS problem. For the system model (5.2), we define $\mathbf{R}$ to be the system response from $\mathbf{w}$ to $\mathbf{x}$, and $\mathbf{M}$ to be the system response from $\mathbf{w}$ to $\mathbf{u}$. The stochastic LQR problem can then be reformulated using the system response as

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} ||_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \\ & \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \frac{1}{z}\mathcal{RH}_\infty. \end{aligned} \tag{5.6}$$

Note that the constraints in (5.6) is just the system level parameterization (SLP) for state feedback systems (cf. Theorem 1 in Chapter 3). As there is no additional system level constraint (SLC) imposed on the system response, we call (5.6) a centralized (unconstrained) LQR problem.

Although (5.6) is a convex SLS, it is still a large-scale optimization problem that depends on the knowledge of the global plant model (A, B_2). To enhance the scalability of controller synthesis and implementation, we impose an additional d-

localized SLC $\mathcal{L}_d$ (this will be formally defined later) and a FIR SLC $\mathcal{F}_T$ on the system response in (5.6), which leads to the LLQR problem given by

$$\underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} ||_{\mathcal{H}_2}^2 \tag{5.7a}$$

$$\text{subject to} \quad \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \tag{5.7b}$$

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{L}_d \cap \mathcal{F}_T \cap \frac{1}{z}\mathcal{RH}_\infty. \tag{5.7c}$$

The goal of this chapter is to show that (i) the LLQR problem (5.7) can be *solved* in a localized and scalable way if the d-localized SLC and the FIR SLC are properly specified, and (ii) the LLQR controller achieving the desired localized system response can be *implemented* in a localized and scalable way. At this point, we assume that the LLQR problem (5.7) is feasible. The feasibility of (5.7), which is called the (d, T) state feedback localizability of the system (A, B_2), will be formally defined in the end of this section.

d-localized SLC

The notion of d-localized SLC $\mathcal{L}_d$ is defined based on the interaction graph $\mathcal{G}$ of the interconnected system. We first recall some standard terminology from graph theory.

Definition 6. *In an unweighted graph $\mathcal{G}$, the* length *of a path is the number of edges it uses.*

Definition 7. *For an interconnected system with an unweighted interaction graph $\mathcal{G}$, the* distance *from subsystem j to subsystem i is defined by the length of the shortest path from node j to node i in the graph $\mathcal{G}$, and is denoted by* $\text{dist}(j \to i)$.

With the definition of the distance function on the interaction graph, we define the d-incoming and outgoing sets of subsystem j as follows.

Definition 8. *The d-outgoing set of subsystem j is defined as* $\text{Out}_j(d) := \{i | dist(j \to i) \leq d\}$*, and the d-incoming set of subsystem j is defined as* $\text{In}_j(d) := \{i | dist(i \to j) \leq d\}$.

Example 4. *For a system* (5.1) *with interaction graph illustrated in Figure 5.2, the* 2*-incoming and* 2*-outgoing sets of subsystem* 5 *are given by* $\text{In}_5(2) = \{2, 3, 4, 5\}$ *and* $\text{Out}_5(2) = \{5, 6, 7, 8, 9, 10\}$*, respectively.*

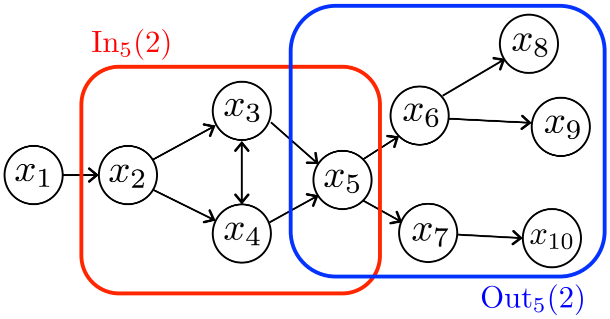


Figure 5.2: Illustration of the 2-incoming and 2-outgoing sets of subsystem 5.

Our approach to making the controller synthesis task specified in optimization problem (5.7) scalable is to confine, or *localize* the effects of each process disturbance w_j to a d-outgoing set at each subsystem j, for a d much smaller than the radius of the interaction graph $\mathcal{G}$. As we make precise in the sequel, this implies that each sub-controller j can be synthesized using the localized plant model contained within its $(d+1)$-outgoing set $\text{Out}_j(d+1)$, and implemented by collecting data from subsystems contained within its $(d+1)$-incoming set $\text{In}_j(d+1)$.

With this approach in mind, we say that the system response $\mathbf{R}$ mapping the state disturbance $\mathbf{w}$ to the state $\mathbf{x}$ is d-localized if its impulse response can be appropriately covered by d-outgoing sets. The formal definition of a d-localized system response is described as follows.

Definition 9. *For the system model* (5.2)*, let* $\mathbf{R}_{ij}$ *denote the transfer function from the perturbation* $\mathbf{w}_j$ *at sub-system* j *to the state* $\mathbf{x}_i$ *at sub-system* i*. The map* $\mathbf{R}$ *is said to be* d*-localized if and only if for every subsystem* j*,* $\mathbf{R}_{ij} = 0$ *for all* $i \notin \text{Out}_j(d)$*. Similarly, let* $\mathbf{M}_{ij}$ *denote the transfer function from the perturbation* $\mathbf{w}_j$ *at sub-system* j *to the control* $\mathbf{u}_i$ *at sub-system* i*. The map* $\mathbf{M}$ *is said to be* d*-localized if and only if for every subsystem* j*,* $\mathbf{M}_{ij} = 0$ *for all* $i \notin \text{Out}_j(d)$*.*

Remark 8. *Alternatively, we can also say that* $\mathbf{R}$ *is* d*-localized if and only if for every subsystem* i*,* $\mathbf{R}_{ij} = 0$ *for all* $j \notin \text{In}_i(d)$*.*

Remark 9. *Let* $\text{supp}\,(\cdot) : \mathbb{R}^{m\times n} \to \{0,1\}^{m\times n}$ *be the support operator, where* $\{\text{supp}\,(A)\}_{ij} = 1$ *if* $A_{ij} \neq 0$ *and* $\{\text{supp}\,(A)\}_{ij} = 0$ *otherwise. Let* $\mathcal{A} = \text{supp}\,(A) \cup \text{supp}\,(I)$*, where* $\cup$ *is the OR operator on binary matrices. For the scalar subsystem model, a convenient way to impose* d*-localized SLC on the system response* $\mathbf{R}$ *is given by*

$$\text{supp}\,(\mathbf{R}) \subseteq \text{supp}\left(\mathcal{A}^d\right). \tag{5.8}$$

In words, this says that the transfer matrix $\mathbf{R}$ is d-localized if and only if the effect of each disturbance is contained, or localized, to within a region of radius d from the source of the disturbance. In general, we can specify a localized constraint with different values of d for the system response $\mathbf{R}$ and $\mathbf{M}$, respectively. As will be shown later in this chapter, a common approach is to enforce a d-localized constraint on $\mathbf{R}$, and a $(d+1)$-localized constraint on $\mathbf{M}$. We call this special type of localized constraint a d-localized SLC, and is denoted by $\mathcal{L}_d$ (cf. (5.7)).

Definition 10. *The subspace $\mathcal{L}_d$ is called a d-localized SLC if it constrains the system response $\mathbf{R}$ to be d-localized, and $\mathbf{M}$ to be $(d+1)$-localized.*

(d, T) state feedback localizability

With the definition of the d-localized SLC, we formally define the notion of (d, T) state feedback localizability of a system (A, B_2) as follows.[2]

Definition 11. *The system (5.2) with system matrices (A, B_2) and an underlying graph $\mathcal{G}$ is said to be (d, T) state feedback localizable if (5.7) is feasible.*

Recall that a system is said to be controllable if there exists a FIR closed loop response. Here, we say that a system is (d, T) localizable if there exists a localized FIR closed loop response (cf., Section 4.2.9). In this sense, localizability can be considered as a stricter notion of the controllability of a system.

The relations between the localizability of the system, the delay pattern of the communication network, and the actuation architecture of the controllers will be discussed in details in Section 5.3.

5.2 Localized Linear Quadratic Regulator

In this section, we propose a scalable algorithm to solve the LLQR problem (5.7) in a localized and scalable way. Specifically, we show that each sub-controller j can be synthesized using the localized plant model contained within the set $\text{Out}_j(d+1)$. Therefore, if the cardinality of the set $\text{Out}_j(d+1)$ for each j is significantly smaller than the size of the global network, then the LLQR controller can be synthesized with $O(1)$ parallel computational complexity. In addition, when the system is (d, T) localizable, the LLQR controller can also be implemented in a localized way, in the sense that each sub-controller j in the network only needs to exchange

[2] Since we only discuss state feedback in this chapter, we sometimes use (d, T) localizability as an abbreviation of (d, T) state feedback localizability in this chapter.

information within the set $\text{In}_j(d+1)$. Finally, we show that the LLQR method naturally supports localized controller re-synthesis subject to model change. This solves all the challenges of scalability mentioned in Section 5.1.2.

For the ease of presentation, we only consider scalar subsystem model (cf. Remark 6) in this section. The method extends to the non-scalar subsystem model in a natural way.

5.2.1 Localized Synthesis

The localized synthesis procedure consists of two main steps. First, we perform a column-wise separation to decompose the LLQR problem (5.7) into parallel column-wise LLQR subproblems. Second, we exploit the d-localized constraint to reduce the dimension of each LLQR subproblem from global scale to local scale. These two steps are explained in details as follows.

Column-wise Separation

We first notice that (5.7) admits a *column-wise separation* — we can solve for the transfer matrices $\mathbf{R}$ and $\mathbf{M}$ column at a time. Denote $\mathbf{G}_j$ the jth column of any transfer matrix $\mathbf{G}$. From the definition of the $\mathcal{H}_2$ norm, we can rewrite the objective in (5.7) as

$$|| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} ||_{\mathcal{H}_2}^2 = \sum_{j=1}^{n} || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix}_j ||_{\mathcal{H}_2}^2.$$

This allows us to decompose (5.7) into n parallel subproblems as

$$\underset{\{\mathbf{R}_j, \mathbf{M}_j\}}{\text{minimize}} \quad || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix}_j ||_{\mathcal{H}_2}^2 \tag{5.9a}$$

$$\text{subject to} \quad \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix}_j = e_j \tag{5.9b}$$

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix}_j \in (\mathcal{L}_d)_j \cap \mathcal{F}_T \cap \frac{1}{z}\mathcal{RH}_\infty \tag{5.9c}$$

for $j = 1, \ldots, n$, with e_j a column vector with 1 on its jth entry and 0 elsewhere. By definition, the jth column of $\mathbf{R}$ and $\mathbf{M}$ are the system response of disturbance $\mathbf{w}_j$ to $\mathbf{x}$ and $\mathbf{u}$, respectively. The column-wise separation property therefore suggests that we can analyze the system response of each local disturbance $\mathbf{w}_j$ in an independent and parallel way. This separation property is implied by two different characteristics of the LLQR problem. First, the system is linear time invariant, so the superposition principle holds. This allows us to verify the system dynamics constraint in (5.7b) column at a time. Second, the disturbance w_i and w_j are assumed to be uncorrelated to each other, i.e., $w \sim \mathcal{N}(0, I)$. In addition, the cost function in (5.7a) is chosen to be the LQR cost. In this case, the objective function (5.7a) can be separated into column-wise sum as above.

Dimension Reduction

We now focus on a specific j in (5.9). Note that the localized constraint $(\mathcal{L}_d)_j$ by definition forces the non-zero entries of $\mathbf{R}$ and $\mathbf{M}$ to be contained within the sets $\text{Out}_j(d)$ and $\text{Out}_j(d+1)$, respectively. Therefore, we can interpret (5.9c) as a constraint to *localize* the effects of disturbance $\mathbf{w}_j$ within the set $\text{Out}_j(d)$, by actuating the control signal within the set $\text{Out}_j(d+1)$. In order to get further insight of (5.9), we rewrite (5.9) from the frequency domain formulation to the time domain formulation as

$$\begin{aligned} \underset{\{x[k],u[k]\}_{k=1}^{T}}{\text{minimize}} \quad & \sum_{k=1}^{T} \|C_1 x[k] + D_{12} u[k]\|_2^2 && \text{(5.10a)} \\ \text{subject to} \quad & x[1] = e_j && \text{(5.10b)} \\ & x[k+1] = Ax[k] + B_2 u[k],\ k = 1, \dots, T && \text{(5.10c)} \\ & x[T+1] = 0 && \text{(5.10d)} \\ & x_i[k] = 0,\ \text{ for } i \notin \text{Out}_j(d),\ k = 1, \dots, T && \text{(5.10e)} \\ & u_i[k] = 0,\ \text{ for } i \notin \text{Out}_j(d+1),\ k = 1, \dots, T, && \text{(5.10f)} \end{aligned}$$

where (5.10b) - (5.10c) comes from the system equation (5.9b), (5.10d) from the FIR constraint $\mathcal{F}_T$, and (5.10e) - (5.10f) from the d-localized constraint $(\mathcal{L}_d)_j$.

When the nonzero entries of the state vector $x[k]$ are contained within the set $\text{Out}_j(d)$, we know by definition that the nonzero entries of $Ax[k]$ will be contained within the set $\text{Out}_j(d+1)$. In addition, as B_2 is diagonal in our scalar subsystem model, the nonzero entries of $B_2 u[k]$ are also contained within the set $\text{Out}_j(d+1)$. Therefore, the nonzero entries of the state vector $x[k+1] = Ax[k] + B_2 u[k]$ at time step $(k+1)$ will automatically be contained within the set $\text{Out}_j(d+1)$. In other words, we do not need to check the condition $x_i[k+1] = 0$ for $i \in \{i | \text{dist}(j \to i) \geq d+2\}$. Applying this argument for each time step k, we note that we only need to check the condition $x_i[k] = 0$ on the boundary set $\{i | \text{dist}(j \to i) = d+1\}$ at each time step k. As long as the states in the boundary set are all zero in all the time horizon, it is guaranteed that the effect of initial condition $x[1] = e_j$ will not *escape* from the localized region $\mathrm{Out}_j(d+1)$, and the condition $x_i[k] = 0$ for $i \in \{i|\mathrm{dist}(j \to i) \geq d+2\}$ for all k will be automatically satisfied. Therefore, we can replace the constraint (5.10e) by the condition

$$x_i[k] = 0, \text{ for } i \in \{i|\mathrm{dist}(j \to i) = d+1\} \tag{5.11}$$

without changing the solution of (5.10).

This motivates the definition of the (j,d)-reduced state and control vectors as follows [71].

Definition 12. *The (j,d)-reduced state vector of x consists of all local state x_i with $i \in \mathrm{Out}_j(d+1)$ and is denoted by $x_{(j,d)}$. Similarly, the (j,d)-reduced control vector of u consists of all local control u_i with $i \in \mathrm{Out}_j(d+1)$ and is denoted by $u_{(j,d)}$.*

We can then define the (j,d)-reduced plant model $A_{(j,d)}$ by selecting submatrix of A consisting of the rows and columns associated with $x_{(j,d)}$, and $B_{(j,d)}$ by selecting submatrix of B_2 consisting of the rows and columns associated with $x_{(j,d)}$ and $u_{(j,d)}$, respectively. The cost matrices $C_{(j,d)}$ and $D_{(j,d)}$ can be defined by selecting submatrices of C_1 and D_{12} consisting of the columns associated with $x_{(j,d)}$ and $u_{(j,d)}$, respectively. This allows us to simplify the optimization problem (5.10) as

$$\begin{aligned}
\underset{\{x_{(j,d)}[k],u_{(j,d)}[k]\}_{k=1}^{T}}{\text{minimize}} \quad & \sum_{k=1}^{T} \|C_{(j,d)}x_{(j,d)}[k] + D_{(j,d)}u_{(j,d)}[k]\|_2^2 && (5.12a)\\
\text{subject to} \quad & x_{(j,d)}[1] = e_{r(j,d)} && (5.12b)\\
& x_{(j,d)}[k+1] = A_{(j,d)}x_{(j,d)}[k] + B_{(j,d)}u_{(j,d)}[k],\ k = 1,\dots,T && (5.12c)\\
& x_{(j,d)}[T+1] = 0 && (5.12d)\\
& x_i[k] = 0, \text{ for } i \in \{i|\mathrm{dist}(j \to i) = d+1\}, && (5.12e)
\end{aligned}$$

where $e_{r(j,d)}$ in (5.12b) is the initial condition e_j in the reduced dimension. The complexity of solving (5.12) is determined by the dimension of the vector $x_{(j,d)}$ and $u_{(j,d)}$, which is equal to the cardinality of the set $\mathrm{Out}_j(d+1)$. When we choose a parameter d that is significantly smaller than the radius of the interaction graph, the dimension of the state $x_{(j,d)}$ (or equivalently, $u_{(j,d)}$) can be much smaller than the dimension of the original state x. In this case, the complexity of solving (5.12) is independent to the size of the global network, i.e., with $O(1)$ computational complexity. In summary, we reduce the global optimization problem (5.10) into

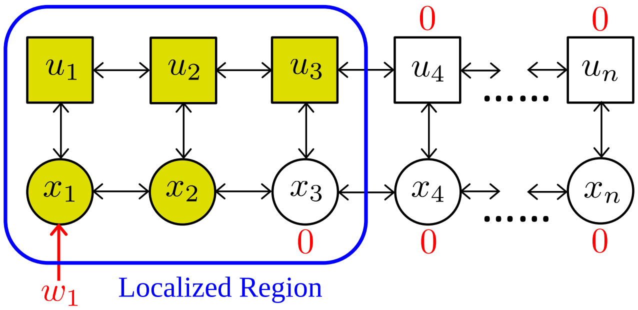


Figure 5.3: The localized region for w_1

a local optimization problem (5.12) that only depends on the local plant model $(A_{(j,d)}, B_{(j,d)})$.

We illustrate the above idea through a simple example.

Example 5. *Consider a chain of n LTI subsystems with a single disturbance* $\mathbf{w}_1$ *at the first subsystem. We solve the LLQR subproblem* (5.10) *with initial condition* $x[1] = e_1$*. Assume that the localized constraint in* (5.10) *is specified by a parameter* $d = 1$*, i.e., we have* $x_3 = \cdots = x_n = 0$ *and* $u_4 = \cdots = u_n = 0$*. In this case, we can solve* (5.10) *using only the information contained within the localized region* $\text{Out}_1(2)$ *shown in Figure 5.3. Intuitively, as long as the boundary constraint* $x_3 = 0$ *is enforced, the system response of* w_1 *can never escape from the pre-specified localized region. In other words, the condition* $x_i = 0$ *for* $i = 4, \dots n$ *is implicitly implied by the boundary constraint* $x_3 = 0$*. In order to make the state* x_3 *at the third subsystem unaffected by the disturbance* w_1 *from the first subsystem, the actuator* u_3 *at the third subsystem must actuate properly to cancel the coupling from the state* x_2 *at the second subsystem. The complexity of analyzing the system response of* w_1 *is completely independent to the size of the global network* n*, and thus our method can be scaled to systems with arbitrary large-scale.*

Summary of Localized Synthesis

In summary, we propose a scalable algorithm to solve the LLQR problem in (5.7) with two steps. First, as the system is LTI and the cost is quadratic, we perform a column-wise separation to decompose the LLQR problem (5.7) into column-wise LLQR subproblems in (5.9). We then analyze the system response for each individual disturbance w_j in an independent and parallel way. Second, we exploit the d-localized constraint imposed on the system response to reduce the dimension of each column-wise LLQR subproblem from global scale to the localized region specified by the set $\text{Out}_j(d+1)$. We then analyze the system response for each individual disturbance w_j within their own localized regions in a parallel way. This argument holds even when the localized regions for each disturbance overlap. As discussed before, if the parameter d is chosen in the way that the cardinality of the set $\text{Out}_j(d+1)$ for each j is significantly smaller (i.e., $O(1)$) than the size of the global network, then the parallel computation complexity of our LLQR synthesis algorithm is $O(1)$.

5.2.2 Localized Implementation

Once we solve the LLQR problem (5.7), we get a localized system response $(\mathbf{R}, \mathbf{M})$. Applying Theorem 1 in Chapter 3, the controller achieving the desired system response can be implemented by

$$\hat{w}[t] = x[t] - \hat{x}[t] \tag{5.13a}$$

$$u[t] = \sum_{\tau=0}^{T-1} M[\tau+1]\hat{w}[t-\tau] \tag{5.13b}$$

$$\hat{x}[t+1] = \sum_{\tau=0}^{T-2} R[\tau+2]\hat{w}[t-\tau]. \tag{5.13c}$$

The internal states of the controller (5.13) should be interpreted as follows: $\hat{w}$ is the controller estimate of the process disturbance, and $\hat{x}$ is a desired or reference state trajectory. The estimated disturbance $\hat{w}[t]$ is computed by taking the difference between the current state measurement $x[t]$ and the current reference state value $\hat{x}[t]$. The control action $u[t]$ and the next reference state value $\hat{x}[t+1]$ are then computed using past estimated disturbances $\hat{w}[t-T+1], \dots, \hat{w}[t]$. As shown in Section 3.1, the estimated disturbance $\hat{w}[t]$ indeed reconstructs the true disturbance $w[t-1]$ that perturbed the plant at time $t-1$; hence $\hat{\mathbf{w}} = z^{-1}\mathbf{w}$. It is then straightforward to show that the desired system response $\{\mathbf{R}, \mathbf{M}\}$ satisfying $\mathbf{x} = \mathbf{R}\mathbf{w}$ and $\mathbf{u} = \mathbf{M}\mathbf{w}$ is achieved.

The controller implementation (5.13) is localized in the following sense: each sub-controller u_i only needs to collect the estimated disturbance from its $(d+1)$-incoming set to compute its control action. Specifically, as the system response $\mathbf{R}$ is d-localized, the nonzero entries of the ith row of $\mathbf{R}$ are contained within the set $\text{In}_i(d)$, which means that we only need to collect $\hat{w}_j$ for $j \in \text{In}_i(d)$ to compute $\hat{x}_i$. Similarly, we can compute u_i by collecting $\hat{w}_j$ for $j \in \text{In}_i(d+1)$. We summarize the LLQR controller synthesis and implementation in Algorithm 1.

Algorithm 1: LLQR Synthesis and Implementation

```
Given the LLQR problem (5.7) ;
for each disturbance $w_j$ do
    Identify its localized region $\text{Out}_j(d+1)$ and define local plant model
      $(A_{(j,d)}, B_{(j,d)})$ within the localized region;
    Solve (5.12) to compute $x_{(j,d)}$ and $u_{(j,d)}$;
if all LLQR subproblems are feasible then
    for each LLQR subproblem do
        Distribute the solution $x_{(j,d)}$ and $u_{(j,d)}$ to the local controllers contained
          within its localized region $\text{Out}_j(d+1)$;
    for each controller $u_i$ do
        Collect the estimated disturbance $\hat{w}_j$ from the region $\text{In}_i(d+1)$ to compute
          $u_i$, $\hat{x}_i$, and $\hat{w}_i$ during controller implementation;
```

It should be noted that all the for loops in Algorithm 1 can be computed in a parallel way.

5.2.3 Localized Re-synthesis

We note that each LLQR subproblem j can be solved using the plant model information contained within its localized region $\text{Out}_j(d+1)$. Therefore, when the plant model changes locally, we only need to re-compute some of the LLQR subproblems to maintain global optimality, and then distribute the solution of these LLQR subproblems within their localized regions.

For instance, suppose that the coupling strength from state x_k to state x_j changes, i.e., the plant parameter A_{jk} changes from some nonzero value to another nonzero value. In this case, if the state x_j is covered by the $(d+1)$-outgoing region of state x_i, then we need to re-compute the ith LLQR subproblem. This is equivalent saying that we need to re-compute the ith LLQR subproblem for $i \in \text{In}_j(d+1)$. Once these LLQR subproblems are re-computed, we distribute each of these solution to its own $(d+1)$-outgoing region. In this case, the controller being updated is covered by the set

$$\bigcup_{i \in \text{In}_j(d+1)} \text{Out}_i(d+1). \tag{5.14}$$

When the sparsity pattern of the system matrix A is symmetric (i.e., the underlying interaction graph $\mathcal{G}$ is undirected), the d-incoming set of state x_j is equal to the d-outgoing set of state x_j, for all d and j. In this case, the set (5.14) can be simplified into $\text{Out}_j(2d+2)$ (or equivalently, $\text{In}_j(2d+2)$).

From the above discussion, we note that LLQR supports real-time localized controller re-synthesis subject to model change. This feature allows incremental design of the optimal controller when the physical system expands.

5.2.4 Summary of LLQR

In this section, we show that the LLQR controller can be synthesized and implemented in a localized and scalable way, if the localized constraint $\mathcal{L}_d$ and the FIR constraint $\mathcal{F}_T$ are properly specified and the system is (d, T) localizable. In addition, we can re-synthesize the optimal controller in a localized way when the model changes locally. All these properties are extremely favorable for large-scale applications.

5.3 State Feedback Localizability

We note that a key assumption of the LLQR algorithm is that the system (A, B_2) is localizable, i.e., the LLQR problem admits a localized feasible solution. In this section, we show that the (d, T) localizability of the system (5.2) depends on two different factors: (i) the delays pattern of the communication network between subcontrollers, and (ii) the locations of the actuators in the network (as specified by the matrix B_2).

We begin by formulating the LLQR problem with a spatiotemporal SLC. The spatiotemporal SLC is described by the intersection of three component: a d-localized SLC, a FIR SLC, and a communication delay SLC (cf. Section 4.2.9). Then, we assume full actuation (i.e., $B_2 = I$), and design the d-localized SLC $\mathcal{L}_d$ and the FIR SLC $\mathcal{F}_T$ for a given communication delay SLC such that the LLQR synthesis problem (5.7) is feasible. Finally, we propose a regularization-based method to *co-design* a feasible LLQR controller and locations of the actuators in the network.

5.3.1 LLQR with Spatiotemporal Constraints

Consider a LLQR problem with a spatiotemporal SLC (cf. Section 4.2.9) given by

$$\underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} ||_{\mathcal{H}_2}^2 \tag{5.15a}$$

$$\text{subject to} \quad \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \tag{5.15b}$$

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{C}_{(t_s,t_c,t_a)} \cap \mathcal{L}_d \cap \mathcal{F}_T \cap \frac{1}{z}\mathcal{RH}_\infty, \tag{5.15c}$$

where $\mathcal{C}_{(t_s,t_c,t_a)}$ is a parameterized communication delay SLC, $\mathcal{L}_d$ a d-localized SLC, and $\mathcal{F}_T$ a FIR SLC. We already introduce the localized SLC and the FIR SLC in the previous section. Here we explain the details of the communication delay SLC as follows.

We assume that the communication network has the same topology as the physical network, and the communication delay SLC $\mathcal{C}_{(t_s,t_c,t_a)}$ is characterized by three parameters: the sensing delay t_s, the communication delay t_c, and the actuation delay t_a. Specifically, it takes time t_s for a sub-controller u_i to access its state measurement x_i, time t_c for a sub-controller to transmit information to its direct neighbors, and time t_a for a control action u_i to alter the state x_i of its subsystem. The delays (t_s, t_c, t_a) are normalized with respect to the sampling time of the discrete time system (5.2), and hence they may be non-integers in general. We adopt the following convention to handle fractional delays: if information is received by a sub-controller between two sampling times t and $t + 1$, then it may be used by the sub-controller to compute its control action at time $t + 1$. The actuation delay t_a is typically set to 1 in order to be consistent to the discrete time system equation (5.2).

Remark 10. *The LLQR problem with spatiotemporal constraint* (5.15) *can be solved in a localized and scalable way using the algorithm proposed in the previous section. Therefore, we solve all the challenges for large-scale systems mentioned in Section 5.1.2.*

5.3.2 Constraint Setup Procedure

For the LLQR problem with a spatiotemporal constraint (5.15), we propose a serial procedure that sequentially designs the localized subspace $\mathcal{L}_d$, the FIR subspace $\mathcal{F}_T$, and the actuation scheme B_2 that ensure the (d, T) localizability of the system. In particular, we treat the actuation scheme B_2 as design variables in this section.

Initialization: Given a distributed plant (5.2), assume a completely dense actuation architecture, i.e., $B_2 = I$.

Subspaces $\mathcal{L}_d$ and $\mathcal{F}_T$: Given the communication delay SLC $\mathcal{C}_{(t_s,t_c,t_a)}$ imposed on the distributed controller, determine the sparsest localized constraint $\mathcal{L}_d$, i.e., with smallest d, and the minimal length T of $\mathcal{F}_T$ such that the LLQR synthesis problem (5.15) is feasible. This task can be accomplished by beginning with d and T set to 1 and incrementing these values until feasibility is achieved.

Actuator Regularization: Given the LLQR synthesis problem (5.15) constrained by the communication delay SLC $\mathcal{C}_{(t_s,t_c,t_a)}$ and the designed SLCs $\mathcal{L}_d$ and $\mathcal{F}_T$, use regularization for design (RFD) [40, 41] to explore the tradeoff between closed loop performance and actuation density, as specified by B_2. If no acceptably sparse actuation architecture can be found, return to the previous step and increase either d or T.

This design procedure is certainly not unique, but is simple and intuitive: we assume that the system has full actuation and determine the "simplest" constraints such that a localized controller can still be synthesized. Then we attempt to remove actuators such that locality and closed loop performance are preserved: if this latter step cannot be satisfactorily completed, increase the complexity of the localized controller (by increasing either d or T) and repeat.

The rest of this section focusses on the subtleties of designing the subspaces $\mathcal{L}_d$ and $\mathcal{F}_T$ and the actuator regularization task. Once the constraint setup procedure is complete, we can use the LLQR method described in Section 5.2 to synthesize a LLQR controller that uses the designed actuation architecture and that respects the localized and FIR SLCs $\mathcal{L}_d$ and $\mathcal{F}_T$.

5.3.3 Designing $\mathcal{L}_d$ and $\mathcal{F}_T$

From the discussion of Section 5.2, the subspace $(\mathcal{L}_d)_j \cap \mathcal{F}_T$ is the spatiotemporal constraint imposed on the system response from disturbance $\mathbf{w}_j$ to the global state $\mathbf{x}$ and control action $\mathbf{u}$. For a d-localized constraint $(\mathcal{L}_d)_j$, the state $\mathbf{x}_i$ is nonzero only if $i \in \text{Out}_j(d)$. The size d of the subset of subsystems that can be perturbed by a disturbance $\mathbf{w}_j$ is primarily determined by the communication delay SLC $\mathcal{C}_{(t_s,t_c,t_a)}$ that are imposed on the controller — if information can be exchanged between sub-controllers very quickly, then they can coordinate their actions to contain the effect of a disturbance in a small d-outgoing set. Conversely, if information is exchanged slowly then it may not be possible to localize a disturbance at all: as an extreme case, if communication between sub-controllers is slower than the propagation of disturbances through the plant, then there is no way to localize the effect of a local disturbance.

It should be clear that the feasibility of a d-localized SLC $\mathcal{L}_d$ depends on both the topology of the interaction graph $\mathcal{G}$ underlying the distributed system (5.2) and the information exchange SLC $\mathcal{C}_{(t_s,t_c,t_a)}$ underlying the distributed controller. For a fully actuated system ($B_2 = I$), we can characterize a relationship between the delay parameters and the minimal size d such that a d-localized constraint set $\mathcal{L}_d$ leads to a feasible LLQR problem (5.15).

Lemma 11. *Assume that the system* (5.2) *is a scalar subsystem model with full actuation, i.e., $B_2 = I$. For a communication delay SLC $\mathcal{C}_{(t_s,t_c,t_a)}$ with $t_c < 1$, the system* (5.2) *is d-localizable if $\frac{t_s+t_a}{1-t_c} - 1 \leq d$.*

Proof. Consider the disturbance $\mathbf{w}_i$ that affects subsystem i. If the LLQR problem with communication delays (5.15) is feasible for a d-localized constraint $\mathcal{L}_d$, only the states in the d-outgoing set $\text{Out}_i(d)$ may be perturbed in closed loop by disturbance $\mathbf{w}_i$. As each state can be directly actuated, we only need to ensure that the state of the "boundary" subsystems k satisfying $\text{dist}\,(i \longrightarrow k) = d + 1$ are not affected by disturbance $\mathbf{w}_i$. By definition, it takes the disturbance $(d + 1)$ time steps to affect these "boundary" states via the dynamics (5.2). As the communication network topology mimics the topology of the plant, measurements of the state deviation $x_i[t]$ taken by subsystem i is transmitted to these boundary subsystems with a delay of $t_s + (d+1)t_c$. Thus if $t_s + (d+1)t_c + t_a \leq d+1$, then the boundary subsystems are given advanced warning of the disturbance that is propagating towards them: as actuation is assumed to be dense, the corresponding boundary sub-controllers can suppress disturbance such that the states of the boundary subsystems are not perturbed. As the initial disturbance was arbitrary, this shows that the delay condition of the lemma is sufficient to ensure the feasibility of a d-localized constraint $\mathcal{L}_d$. □

Remark 11. *This condition is reminiscent of the delay characterization of QI developed in [52]. Using our notation and setup, the delays (t_s, t_c, t_a) define a QI subspace constraint if $p \cdot t_c \leq p + t_a + t_s$, where p is the distance between* any *pair of states. Thus we see that the delay condition stated in Lemma 11 is more restrictive than the QI condition: for example, if $t_c = 1$, then the delays define a QI subspace constraint, but the system is not d-localizable for a d smaller than the diameter of* *the interaction graph $\mathcal{G}$. However, because the LLQR controller is by definition d-localized, communication between subsystems is limited to within a subset of radius d if $t_c < 1$ — in contrast, the QI controller requires that local information be shared globally if the plant* (5.2) *has a strongly connected topology.*

We illustrate the meaning of Lemma 11 in Figure 5.4.

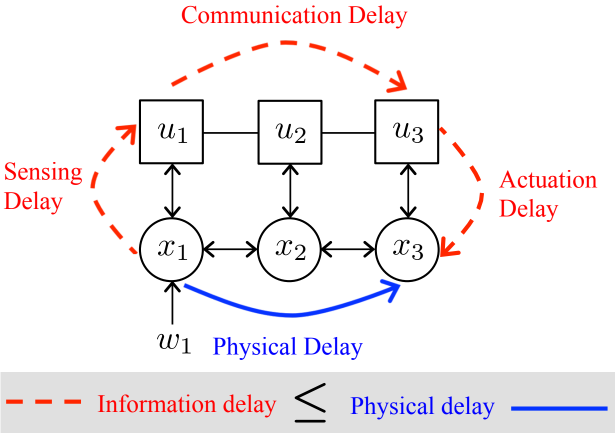


Figure 5.4: Information delay and physical delay

Example 6. *Consider a single disturbance w_1 in Figure 5.4 with a* 1*-localized SLC. Using Lemma 11, the information delay (the sum of the sensing delay from x_1 to u_1, the communication delay from u_1 to u_3, and the actuation delay from u_3 to x_3) needs to be less than or equal to the physical delay (the delay from x_1 to x_3 through dynamics coupling), so that the LLQR problem is feasible.*

We further specialize the delay condition of Lemma 11 by assuming that $t_s = 0$ and $t_a = 1$. The condition of Lemma 11 then reduces to $\frac{t_c}{1-t_c} \leq d$ — as we assume full actuation, every d-localized subset of the system is trivially controllable, and thus we can set the FIR horizon to be $T \geq d+1$. Thus using Lemma 11 and the previous discussion, given a communication delay SLC $\mathcal{C}_{(t_s,t_c,t_a)}$, we can identify small values d and T to ensure that the spatiotemporal SLC $\mathcal{S} := \mathcal{C}_{(t_s,t_c,t_a)} \cap \mathcal{L}_d \cap \mathcal{F}_T$ leads to a feasible LLQR problem (5.15).

We now give an explicit construction for $\mathcal{S}$. Assume that at time step k, we can sense the state deviation, transmit the information to j-hop away, and actuate the state. Using the delay parameters (t_s, t_c, t_a), we must have the inequality $t_s + j \cdot t_c + t_a \leq k$, or alternatively, $j \leq \frac{k-t_s-t_a}{t_c}$. The largest integer j that satisfies this inequality is given by $j = \lfloor \frac{k-t_s-t_a}{t_c} \rfloor$. Since the communication network has the same topology as the physical network, this is equivalently saying that the kth spectral component of the system response $R[k]$ is j-localized, i.e., the sparsity pattern of $R[k]$ is covered by the support of $\mathcal{A}^j$, with $\mathcal{A} = \text{supp}\,(A) \cup \text{supp}\,(I)$. The similar argument holds for the system response $\mathbf{M}$ as well. Therefore, the communication delay SLC $C_{(t_s,t_c,t_a)}$ imposes the following constraint on the system response $(\mathbf{R}, \mathbf{M})$:

$$\text{supp}\,(R[k]), \text{supp}\,(M[k]) \subseteq \text{supp}\left(\mathcal{A}^{\lfloor \frac{k-t_s-t_a}{t_c} \rfloor}\right). \tag{5.16}$$

Combining this constraint with the d-localized constraint $\mathcal{L}_d$ and the FIR constraint $\mathcal{F}_T$, the spatiotemporal SLC in (5.15c) is specified by

$$\begin{aligned} \text{supp}\,(R[k]) &\subseteq \text{supp}\left(\mathcal{A}^{\min(d, \lfloor \frac{k-t_s-t_a}{t_c} \rfloor)}\right) \\ \text{supp}\,(M[k]) &\subseteq \text{supp}\left(\mathcal{A}^{\min(d+1, \lfloor \frac{k-t_s-t_a}{t_c} \rfloor)}\right) \end{aligned} \tag{5.17}$$

for $k = 1, \cdots, T$, and $R[\tau] = 0$ and $M[\tau] = 0$ for all $\tau > T$. According to Lemma 11, the LLQR problem (5.15) is feasible for a fully actuated system if $t_c < 1$, $d \geq \frac{t_s+t_a}{1-t_c} - 1$, and $T \geq d + 1$.

Usually, the parameters of communication delay (t_s, t_c, t_a) are imposed by the pre-specified hardware constraint. In contrast, the parameters T and d can be chosen by the controller designer. The parameter T dominates the trade-off between settling time and transient performance (including $\mathcal{H}_2$ norm or maximum overshoot), and it also affects FIR horizon of the LLQR controller implementation (5.13) — thus T should chosen as small as possible while still leading to acceptable transient performance. The parameter d should also be kept as small as possible as it controls the complexity of the controller synthesis and implementation procedures. However, if sparse actuation is desired, then it is desirable to increase d from its minimum to allow some flexibility in the actuation architecture. Although these parameters typically need to be identified by trial and error, they are integer quantities that can be explored fairly easily. For instance, in the simulation of Chapter 7, we show that the localized size d often only needs to be increased by one or two to be able to design sparse actuation schemes that nonetheless achieve good closed loop performance.

5.3.4 LLQR with Actuator Regularization

Now we show how to explore the tradeoff between closed loop performance and actuation density using the RFD framework (cf., Section 4.3.3), given a feasible spatiotemporal constraint $\mathcal{S} = C_{(t_s,t_c,t_a)} \cap \mathcal{L}_d \cap \mathcal{F}_T$.

Recall that the LLQR controller is implemented using the equations (5.13) — it follows that if the ith row of the map $\mathbf{M}$ is zero, i.e., if $e_i^\top \mathbf{M} = 0$, then the control action u_i is identically zero, and the corresponding actuator at subsystem i can be discarded. If we want to construct a controller that uses at most r actuators, we can impose an additional row-cardinality constraint $N_r(\mathbf{M}) \leq r$ on the transfer matrix $\mathbf{M}$, where $N_r(\mathbf{M})$ represents the number of nonzero rows of $\mathbf{M}$. If we incorporate this constraint into the LLQR problem (5.15), we can formulate the actuator constrained LLQR control problem as

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} ||_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \text{(5.15b) and (5.15c)} \\ & N_r(\mathbf{M}) \leq r. \end{aligned} \tag{5.18}$$

Problem (5.18) is a combinatorial optimization problem due to the last constraint, and is generally computationally hard to solve. The RFD framework allows such combinatorial controller architecture design problems to be solved in a tractable way based on convex relaxations of the combinatorial penalties. We use this technique to relax (5.18) and formulate the LLQR problem with actuator regularization as

$$\underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} ||_{\mathcal{H}_2}^2 + ||\mu \mathbf{M}||_{\mathcal{U}} \tag{5.19a}$$

$$\text{subject to} \quad \text{(5.15b) and (5.15c),} \tag{5.19b}$$

where $||.||_{\mathcal{U}}$ is the actuator norm introduced in [40, 41, 70, 72]. One possible actuator norm is given by

$$||\mu \mathbf{M}||_{\mathcal{U}} = \sum_{i=1}^{n} \mu_i ||e_i^\top \mathbf{M}||_{\mathcal{H}_2}, \tag{5.20}$$

where μ_i is the relative price of each actuator, and μ is a diagonal matrix with μ_i being its ith diagonal entry. When $\mu_i = 1$ for all i, (5.20) is equivalent to the ℓ_1/ℓ_2 norm (group lasso [78] in the statistical learning literature). The parameters $\{\mu_i\}$ can be used to explore the tradeoff between closed loop performance (the square of $\mathcal{H}_2$ norm in (5.19a)) and the actuation architecture complexity (as measured by the actuator norm in (5.19a)), given a fixed controller spatiotemporal complexity (as specified by $\mathcal{L}_d$ and $\mathcal{F}_T$). Further, once an actuation architecture has been identified, a traditional LLQR optimization problem (5.15) can then be solved restricted to the designed actuation architecture, thus removing any bias or conservatism introduced by the actuator norm regularizer (cf. [40, 41] for why this two step approach is appropriate).

To further enhance the sparsity of the solution, we can use the reweighted ℓ_1 algorithm proposed in [6] to iteratively set the weights $\{\mu_i\}$ and solve (5.19) multiple times. Let $\mu_i^{(0)} = \mu_0$ for $i = 1, \dots, n$. Let $\mathbf{M}^{(k)}$ be the optimal solution of (5.19) when the weights are given by $\{\mu_i^{(k)}\}_{i=1}^n$. We update the weights at iteration $(k+1)$ by

$$\mu_i^{(k+1)} = (\,||e_i^\top \mathbf{M}^{(k)}||_{\mathcal{H}_2} + \epsilon)^{-1} \tag{5.21}$$

for some small ϵ. It is shown in [6] that this reweighted scheme usually results in sparser solution.

Equation (5.19) is a convex optimization problem, and hence is tractable to solve. However, the regularization term in (5.19) does not decompose in column at a time, so we cannot apply the original LLQR decomposition technique to solve the actuator regularized LLQR in an efficient and scalable way. Fortunately, we will show in Chapter 7 that the actuator regularized LLQR problem (5.19) belongs to the class of CLS-SLS problems, which can still be solved in a localized and scalable way by exploiting the *partial separability* of the quadratic cost and actuator regularizer using distributed optimization algorithm such as alternating direction method of multipliers (ADMM).

5.4 Nearly Localizable Systems

In the previous two subsections, we assume that the system (A, B_2) is (d, T) localizable. That is to say, there exists a (d, T) localized system response $(\mathbf{R}, \mathbf{M})$ that satisfies the constraints (5.7b) - (5.7c). We then show that the controller achieving the desired localized system response can be implemented in a localized way using (5.13). In practice, the system may not be exactly localizable. An interesting question is: if the system response $(\mathbf{R}, \mathbf{M})$ is only approximately (d, T) localizable, can we still implement a localized stabilizing controller using (5.13)? How much *non-localizability* is allowed? In this section, we give a simple necessary and sufficient condition on the robustness of the localized implementation (5.13) for *nearly* localizable systems.

We first consider a pair of (d, T) localized transfer matrices $(\mathbf{R}_c, \mathbf{M}_c)$ that satisfies

$$\begin{bmatrix} \mathbf{R}_c \\ \mathbf{M}_c \end{bmatrix} \in \mathcal{L}_d \cap \mathcal{F}_T \cap \frac{1}{z}\mathcal{RH}_\infty. \tag{5.22}$$

We assume that the controller is implemented using the following equations:

$$\hat{\mathbf{w}} = \mathbf{x} - \hat{\mathbf{x}} \tag{5.23a}$$

$$\mathbf{u} = z\mathbf{M}_c\hat{\mathbf{w}} \tag{5.23b}$$

$$\hat{\mathbf{x}} = (z\mathbf{R}_c - I)\hat{\mathbf{w}}. \tag{5.23c}$$

As $(\mathbf{R}_c, \mathbf{M}_c)$ are localized transfer matrices, (5.23) is a localized controller implementation. However, the actual system response may not be localized because the transfer matrices $(\mathbf{R}_c, \mathbf{M}_c)$ may not satisfy the equality constraint (5.7b). We use a strictly proper stable transfer matrix $\boldsymbol{\Delta_{RM}} \in \frac{1}{z}\mathcal{RH}_\infty$ to quantify the infeasibility of (5.7b) as follows:

$$\boldsymbol{\Delta_{RM}} = \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_c \\ \mathbf{M}_c \end{bmatrix} - I. \tag{5.24}$$

Note that (5.24) becomes (5.7b) when $\boldsymbol{\Delta_{RM}} = 0$. In this case, $(\mathbf{R}_c, \mathbf{M}_c)$ becomes the actual system response.

Equation (5.24) should be interpreted in the following way: (A, B_2) in (5.24) represents the *actual* system model, and $(\mathbf{R}_c, \mathbf{M}_c)$ in (5.22) - (5.24) represents the *actual* controller implementation. The infeasibility measure $\boldsymbol{\Delta_{RM}}$ may come from various sources, including model uncertainty, model bias, computation error, or localized truncation. For instance, when the system has an approximated localized system response, we can truncate the system response to get a pair of localized transfer matrices $(\mathbf{R}_c, \mathbf{M}_c)$, then implement the controller in a localized way. This localized truncation will induce a non-zero $\boldsymbol{\Delta_{RM}}$ in (5.24). The transfer matrix $\boldsymbol{\Delta_{RM}}$ characterizes all kinds of mismatch between the actual system model and the actual controller implementation.

The following theorem gives a necessary and sufficient condition for internal stability in the presence of $\boldsymbol{\Delta_{RM}}$.

Theorem 4. *Let* $(\mathbf{R}_c, \mathbf{M}_c, \boldsymbol{\Delta_{RM}})$ *be a solution of* (5.22) *and* (5.24). *Then, the controller implementation* (5.23) *internally stabilizes the system* (A, B_2) *if and only if* $(I + \boldsymbol{\Delta_{RM}})^{-1}$ *is stable.*

Proof. The proof is similar to the one in Lemma 5. We first adapt (3.24a) - (3.24c) into our problem setting:

$$(zI - A)\mathbf{x} = B_2\mathbf{u} + \boldsymbol{\delta_x} \tag{5.25a}$$

$$\mathbf{u} = z\mathbf{M}_c\hat{\mathbf{w}} + \boldsymbol{\delta_u} \tag{5.25b}$$

$$\mathbf{x} = z\mathbf{R}_c\hat{\mathbf{w}} - \boldsymbol{\delta_y}. \tag{5.25c}$$

Substituting (5.25b) and (5.25c) into (5.25a), we have

$$z(zI - A)\mathbf{R}_c\hat{\mathbf{w}} - (zI - A)\boldsymbol{\delta_y} = zB_2\mathbf{M}_c\hat{\mathbf{w}} + B_2\boldsymbol{\delta_u} + \boldsymbol{\delta_x}.$$

Moving $\hat{\mathbf{w}}$ to the left-hand-side and using the relation (5.24) from the assumption, we have

$$z(I + \boldsymbol{\Delta_{RM}})\hat{\mathbf{w}} = (zI - A)\boldsymbol{\delta_y} + B_2\boldsymbol{\delta_u} + \boldsymbol{\delta_x},$$

Denote $\boldsymbol{I_\Delta} = (I + \boldsymbol{\Delta_{RM}})^{-1}$. We then have the closed loop transfer matrices from ($\boldsymbol{\delta_x}$, $\boldsymbol{\delta_y}$, $\boldsymbol{\delta_u}$) to $\hat{\mathbf{w}}$ given by

$$\hat{\mathbf{w}} = \frac{1}{z}\boldsymbol{I_\Delta}\boldsymbol{\delta_x} + \boldsymbol{I_\Delta}(I - \frac{1}{z}A)\boldsymbol{\delta_y} + \frac{1}{z}\boldsymbol{I_\Delta}B_2\boldsymbol{\delta_u}. \tag{5.26}$$

Substituting (5.26) into (5.25b) and (5.25c), we have the closed loop transfer matrices from ($\boldsymbol{\delta_x}$, $\boldsymbol{\delta_y}$, $\boldsymbol{\delta_u}$) to ($\mathbf{x}$, $\mathbf{u}$, $\hat{\mathbf{w}}$) summarized in Table 5.1. Clearly, if $\boldsymbol{I_\Delta}$ is stable, then all the transfer matrices in Table 5.1 are stable. If $\boldsymbol{I_\Delta}$ is unstable, then the closed loop maps from $\boldsymbol{\delta_x}$ to $\hat{\mathbf{w}}$ will be unstable, and the controller does not internally stabilize the system. Therefore, the stability of $\boldsymbol{I_\Delta} = (I + \boldsymbol{\Delta_{RM}})^{-1}$ is necessary and sufficient condition for the controller implementation (5.23) to internally stabilize the system (A, B_2). □

Table 5.1: Closed Loop Maps With Non-localizability

	$\boldsymbol{\delta_x}$	$\boldsymbol{\delta_y}$	$\boldsymbol{\delta_u}$
$\mathbf{x}$	$\mathbf{R}_c\boldsymbol{I_\Delta}$	$\mathbf{R}_c\boldsymbol{I_\Delta}(zI - A) - I$	$\mathbf{R}_c\boldsymbol{I_\Delta}B_2$
$\mathbf{u}$	$\mathbf{M}_c\boldsymbol{I_\Delta}$	$\mathbf{M}_c\boldsymbol{I_\Delta}(zI - A)$	$I + \mathbf{M}_c\boldsymbol{I_\Delta}B_2$
$\hat{\mathbf{w}}$	$\frac{1}{z}\boldsymbol{I_\Delta}$	$\boldsymbol{I_\Delta}(I - \frac{1}{z}A)$	$\frac{1}{z}\boldsymbol{I_\Delta}B_2$

Note that in the present of $\boldsymbol{\Delta_{RM}}$, the actual closed loop system response is no longer exactly localizable. However, when $\boldsymbol{\Delta_{RM}}$ is small enough, $\boldsymbol{I_\Delta} = (I + \boldsymbol{\Delta_{RM}})^{-1}$ will be stable, and the closed loop system achieves global stability. The robustness of the controller implementation (5.23) allows us to use approximated solution to implement the localized controller even when the system is not exactly (d, T) localizable.

Now we specialize our discussion to the case where the non-localizability measure $\boldsymbol{\Delta_{RM}}$ comes from the bias or the uncertainty of the plant model. Specifically, we assume that the localized controller is designed based on the system model (A, B_2), but is applied on a different system model $(A + \Delta_A, B_2 + \Delta_B)$. In this case, we have

$$\boldsymbol{\Delta_{RM}} = -\Delta_A \mathbf{R}_c - \Delta_B \mathbf{M}_c. \tag{5.27}$$

From (5.27), when the plant has a higher uncertainty on its A matrix, we should penalize more on the system response $\mathbf{R}_c$ to make the non-localizability measure $\boldsymbol{\Delta_{RM}}$ small. On the other hand, if the system has a higher uncertainty on the gain matrix B_2, we should penalize more on the system response $\mathbf{M}_c$.

5.5 Adaptive Constraint Update with Performance Guarantee

Although the LLQR controller with a fixed spatiotemporal constraint can be designed and implemented in a scalable way, the additional constraint also makes the LLQR controller sub-optimal to the centralized one. In this section, we quantify the sub-optimality of a LLQR controller with a given spatiotemporal constraint. The key idea is a scalable algorithm to compute a non-trivial lower bound of the cost achieved by an idealized centralized controller. In particular, both the lower bound and the LLQR controller can be computed using local plant model information. We then use the ratio of the upper bound (the cost achieved by the LLQR controller) and the lower bound to adaptively update the spatiotemporal constraint imposed on the LLQR controller until a sub-optimality guarantee is satisfied. Using this constraint update algorithm, we are able to design the tradeoff between the complexity of a LLQR controller and its closed loop performance in a localized and automated way.

5.5.1 Lower Bound of Centralized LQR Cost

We first assume that the cost matrix $\begin{bmatrix} C_1 & D_{12} \end{bmatrix}$ in (5.10) is given by

$$\begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \begin{bmatrix} (D_x)^{\frac{1}{2}} & 0 \\ 0 & (D_u)^{\frac{1}{2}} \end{bmatrix}$$

for some positive semidefinite diagonal matrices D_x and D_u. Consider a centralized LQR problem with a given initial condition $x[1] = e_j$:

$$\underset{\{x[k],u[k]\}_{k=1}^{\infty}}{\text{minimize}} \quad \sum_{k=1}^{\infty} x[k]^\top D_x x[k] + u[k]^\top D_u u[k] \tag{5.28a}$$

$$\text{subject to} \quad x[1] = e_j \tag{5.28b}$$

$$x[k+1] = Ax[k] + B_2 u[k], \; k = 1, \dots, \infty. \tag{5.28c}$$

The LLQR problem can be formulated by incorporating additional constraints (5.10d) - (5.10f) into (5.28), as shown in (5.10). Let Ψ_j^* be the optimal value of problem (5.28), and $\Psi_j^{upper} \geq \Psi_j^*$ be the LLQR cost, i.e., the optimal value of (5.28) when additional constraints (5.10d) - (5.10f) are included. For a large-scale system, the value Ψ_j^* cannot be computed in a scalable manner. In order to quantify the degradation of Ψ_j^{upper}, we propose a scalable algorithm to provide a non-trivial lower bound $\Psi_j^{lower} \leq \Psi_j^*$ of the centralized optimal cost. In this way, the degradation Ψ_j^{upper}/Ψ_j^* of the LLQR controller is upper bounded by the ratio $\Psi_j^{upper}/\Psi_j^{lower}$. In other words, we can compute a sub-optimality guarantee $\Psi_j^{lower}/\Psi_j^{upper}$ for the LLQR controller with a given initial condition $x[1] = e_j$ using only local plant model information.

From the discussion of Section 5.2, we can analyze the solution of (5.10), which is an upper bound of the optimal value of (5.28), within the localized region $\text{Out}_j(d+1)$ for each disturbance w_j. Similar to the upper bound problem, here we would like to compute a lower bound of the optimal value of (5.28) based on the network information contained within the set $\text{Out}_j(d+1)$. Consider the following example.

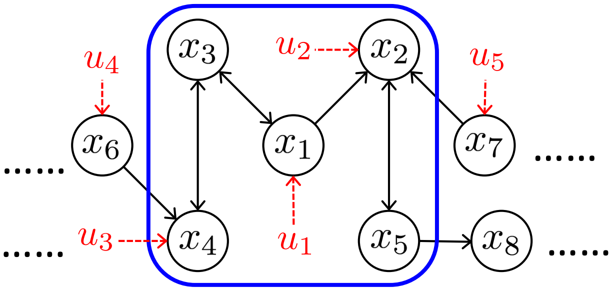


Figure 5.5: Localized region for the initial condition $x[1] = e_1$ in a large network

Example 7. *Figure 5.5 shows the interconnection of subsystem x_1 to other subsystems in a large network. The global network in this example can be arbitrary large. The goal is to compute a lower bound of* (5.28) *for a given initial condition $x[1] = e_1$ within a given localized region* $\text{Out}_1(2) = \{x_1, x_2, x_3, x_4, x_5\}$. *In particular, we have no information about the system model outside the localized region* $\text{Out}_1(2)$.

The intuition of our approach is as follows. First, note that the matrices (D_x, D_u) are diagonal and positive semidefinite. Therefore, a lower bound of the objective function in (5.28) can be computed by setting $x_i = 0$ and $u_i = 0$ for all x_i and u_i outside the localized region. Then, we restrict the system dynamics (5.28c) within the localized region, and match the inconsistent part by introducing additional zero penalty control action at the boundary of the localized region. We formalize this idea by the partition of the state and control vectors as follows.

Given a localized region $\text{Out}_j(d+1)$, we partition the global state vector x into three components: the internal state vector x_{in}, the boundary state vector x_b, and the external state vector x_{out}. We partition the global control vector u into two components: the internal control vector u_{in} and the external control vector u_{out}. The state vector is partitioned in the following way.

Definition 13. *The external state vector x_{out} of a localized region $\text{Out}_j(d+1)$ is defined by the set*

$$x_{out} = \{x_i | x_i \notin \text{Out}_j(d+1)\}.$$

The boundary state vector x_b of $\text{Out}_j(d+1)$ is a subset of $\text{Out}_j(d+1)$ that contains the states coupled from the external state vector x_{out}. Specifically, we have

$$x_b = \{x_i | x_i \in \text{Out}_j(d+1), \exists x_k \notin \text{Out}_j(d+1), A_{ik}x_k \neq 0\}.$$

Finally, the internal state vector is given by

$$x_{in} = \{x_i | x_i \in \text{Out}_j(d+1), \forall x_k \notin \text{Out}_j(d+1), A_{ik}x_k = 0\}.$$

Example 8. *Consider the example in Figure 5.5 with localized region $\text{Out}_1(2)$ $= \{x_1, x_2, x_3, x_4, x_5\}$. The external state vector of this localized region contains all the states except those in $\text{Out}_1(2)$, i.e., x_{out} includes $x_6, \ldots, x_8$ and all the states not shown in the figure. The boundary state vector contains the states that are coupled from the external region, which is $x_b = \{x_2, x_4\}$. The internal state vector is given by $x_{in} = \{x_1, x_3, x_5\}$. Note that x_5 is coupled to, but not coupled from the external region, and thus is a component of the internal state vector.*

Remark 12. *It should be noted that the boundary region defined here for the lower bound problem is* different *from the boundary region defined for the LLQR problem* (5.10)*. For the LLQR problem (an upper bound of* (5.28)*), the boundary states are those that couple* to *the external region (e.g., $\{x_5\}$ in Figure 5.5). For the lower bound of* (5.28)*, the boundary states are those that couple* from *the external region (e.g., $\{x_2, x_4\}$ in Figure 5.5).*

The external control vector u_{out} for a localized region $\text{Out}_j(d+1)$ is defined by the set of control actions that only directly affect the states in x_{out}. Let b_i be the ith column of matrix B_2. For the scalar subsystem model, the external control vector u_{out} is the set of u_i where the nonzero locations of $b_i u_i$ are contained within the set x_{out}. The internal control vector u_{in} is the complement of the external control vector u_{out}.

Example 9. *For Figure 5.5, we have $u_{in} = \{u_1, u_2, u_3\}$. The external control vector u_{out} contains u_4 and u_5 and all the control actions not shown in the figure.*

With the partition of the state and control vectors, we can rewrite the objective function in (5.28a) by

$$\sum_{k=1}^{\infty} \begin{bmatrix} x_{in}[k] \\ x_b[k] \\ x_{out}[k] \end{bmatrix}^\top \begin{bmatrix} D_{x,in} & 0 & 0 \\ 0 & D_{x,b} & 0 \\ 0 & 0 & D_{x,out} \end{bmatrix} \begin{bmatrix} x_{in}[k] \\ x_b[k] \\ x_{out}[k] \end{bmatrix} + \begin{bmatrix} u_{in}[k] \\ u_{out}[k] \end{bmatrix}^\top \begin{bmatrix} D_{u,in} & 0 \\ 0 & D_{u,out} \end{bmatrix} \begin{bmatrix} u_{in}[k] \\ u_{out}[k] \end{bmatrix}. \tag{5.29}$$

As D_x and D_u are diagonal and positive semidefinite, a lower bound for (5.29) is given by

$$\sum_{k=1}^{\infty} \begin{bmatrix} x_{in}[k] \\ x_b[k] \end{bmatrix}^\top \begin{bmatrix} D_{x,in} & 0 \\ 0 & D_{x,b} \end{bmatrix} \begin{bmatrix} x_{in}[k] \\ x_b[k] \end{bmatrix} + u_{in}[k]^\top D_{u,in} u_{in}[k]. \tag{5.30}$$

Note that (5.30) can be computed using the information contained within the localized region $\text{Out}_j(d+1)$.

The system dynamics in (5.28c) can be rewritten as

$$\begin{bmatrix} x_{in}[k+1] \\ x_b[k+1] \\ x_{out}[k+1] \end{bmatrix} = \begin{bmatrix} A_{in,in} & A_{in,b} & 0 \\ A_{b,in} & A_{b,b} & A_{b,out} \\ A_{out,in} & A_{out,b} & A_{out,out} \end{bmatrix} \begin{bmatrix} x_{in}[k] \\ x_b[k] \\ x_{out}[k] \end{bmatrix} + \begin{bmatrix} B_{in,in} & 0 \\ B_{b,in} & 0 \\ B_{out,in} & B_{out,out} \end{bmatrix} \begin{bmatrix} u_{in}[k] \\ u_{out}[k] \end{bmatrix}. \tag{5.31}$$

The zero in equation (5.31) is from the definition — the external state vector x_{out} does not couple to the internal state vector x_{in} directly, and the external control vector u_{out} cannot adjust the internal state vector x_{in} and the boundary state vector x_b directly. The only possibility that connects the external region to the localized region is through the term $A_{b,out}$. This coupling can be captured by introducing an additional control action $u_{add}[k] = A_{b,out} x_{out}[k]$ into the system. In this case, the system dynamics within the localized region is given by

$$\begin{bmatrix} x_{in}[k+1] \\ x_b[k+1] \end{bmatrix} = \begin{bmatrix} A_{in,in} & A_{in,b} \\ A_{b,in} & A_{b,b} \end{bmatrix} \begin{bmatrix} x_{in}[k] \\ x_b[k] \end{bmatrix} + \begin{bmatrix} B_{in,in} & 0 \\ B_{b,in} & I \end{bmatrix} \begin{bmatrix} u_{in}[k] \\ u_{add}[k] \end{bmatrix} \tag{5.32}$$

$$u_{add}[k] = A_{b,out} x_{out}[k]. \tag{5.33}$$

Since we are only interested in the lower bound of (5.28), we can discard the constraint (5.33) to increase the feasible set of the optimization problem. Combining with the objective function (5.30), we can compute a lower bound of (5.28) by solving the following optimization problem

$$\text{minimize} \quad (5.30) \tag{5.34a}$$

$$\text{subject to} \quad \begin{bmatrix} x_{in}[1] \\ x_b[1] \end{bmatrix} = e_{r(j)} \text{ and } (5.32), \tag{5.34b}$$

where $e_{r(j)}$ is the initial condition e_j in the reduced dimension. Note that (5.34) is a smaller centralized LQR problem that can be computed using only the plant model information contained within the localized region $\text{Out}_j(d+1)$. Similar to the LLQR problem, the computational complexity of solving (5.34) depends on the number of nodes in the localized region. If the size of the localized region is significantly smaller than the size of the global network, then (5.34) can be solved efficiently.

Remark 13. *To make the LQR problem* (5.34) *well-posed, we can add a term* $\sum_{k=1}^{\infty} \epsilon u_{add}^\top[k]\, u_{add}[k]$ *in the objective function for some small* ϵ*. As long as* ϵ *is small enough, problem* (5.34) *still gives a lower bound for the optimal value in* (5.28).

Example 10. *Consider the example in Figure 5.5 with localized region* $\text{Out}_1(2) = \{x_1, x_2, x_3, x_4, x_5\}$*. To compute a lower bound of the optimal cost, we introduce additional control actions on the boundary states* x_2 *and* x_4*. The lower bound can then be solved by the LQR problem shown in Figure 5.6.*

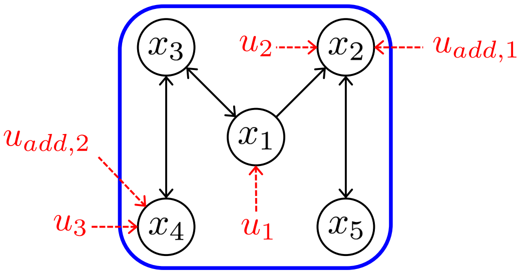


Figure 5.6: Lower bound problem for Figure 5.5

Remark 14. *For comparison, we include the upper bound problem (LLQR problem) for Figure 5.5 in Figure 5.7. For the upper bound problem, we introduce additional constraint $x_5 = 0$ into the system. As discussed before, the boundary states for the upper bound problem is different from that for the lower bound problem.*

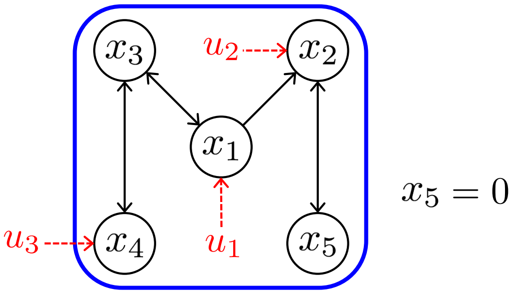


Figure 5.7: Upper bound problem for Figure 5.5

Remark 15. *The upper and lower bounds for distributed optimal control problem* (2.8) *satisfying a QI subspace constraint can be computed in a similar fashion. First, we reformulate the QI subspace constraint imposed on the distributed controller into a subspace SLC, as shown in* (4.15)*. With a given localized region, the lower bound of the distributed optimal control problem is then given by* (5.34) *with an additional subspace constraint. This becomes another distributed optimal control problem, but only requires the plant model information contained within the pre-specified localized region. The distribute optimal control problem with reduced dimension can then be solved by the algorithm presented in [29].*

5.5.2 Adaptive Constraint Update Algorithm

As described in the previous subsection, the LLQR controller for a given initial condition $x[1] = e_j$ achieves at least $\Psi_j^{lower}/\Psi_j^{upper}$ degree of optimality. This ratio can be used as a signal to update the spatiotemporal constraint imposed on the LLQR controller. For instance, we can increase the radius d of the d-localized SLC or the length T of the FIR SLC to synthesize a LLQR controller with tighter upper bound. Likewise, increasing the size of the localized region also gives a tighter lower bound. The high-level idea of this adaptive constraint update algorithm is presented in Algorithm 2.

Note that the for loop in Algorithm 2 can be computed in a parallel way. This means that all the subsystems can compute their bounds within their localized regions in parallel. There are multiple ways to update the spatiotemporal constraint

Algorithm 2: Localized LQR with Adaptive Constraint Update

```
Given the global plant model $(A, B_2)$, diagonal cost matrices $(D_x, D_u)$, an initial
  spatiotemporal SLC $\mathcal{C} \cap \mathcal{L}_d \cap \mathcal{F}_T$, and an optimality requirement $\mu \in [0, 1]$;
for each subsystem $x_j$ do
    Initialize $\Psi_j^{lower} = 0$, $\Psi_j^{upper} = \infty$;
    while $\Psi_j^{lower} / \Psi_j^{upper} < \mu$ do
        Solve the LLQR subproblem in Algorithm 1 to obtain $\Psi_j^{upper}$;
        Solve (5.34) in the localized region to obtain $\Psi_j^{lower}$;
        if $\Psi_j^{lower} / \Psi_j^{upper} < \mu$ then
            Relax the constraint by increasing the length $T_j$ of the FIR constraint or
              the size of the localized region $\text{Out}_j(d + 1)$;
```

in Algorithm 2. In practice, the LLQR algorithm is effective only when all the localized regions are significantly smaller than the global network. Therefore, we should try to keep the localized regions as small as possible. It should be noted that Algorithm 2 may not be feasible with arbitrary μ when there is a hard constraint imposed on the maximum size of the localized region. However, as will be shown in Section 5.7, it is possible for the LLQR controller to achieve 99% global optimality even with some extremely sparse constraints. Assume that Algorithm 2 found a solution $\Psi_j^{lower} / \Psi_j^{upper} \geq \mu$ for all j for a given μ. Then, the LLQR controller achieves at least $\sum_j \Psi_j^{lower} / \sum_j \Psi_j^{upper} \geq \mu$ degree of optimality compared to the idealized centralized controller

5.6 Localized Distributed Kalman Filter

In this section, we present the localized distributed Kalman filter (LDKF) [74] architecture for a class of large-scale state estimation problems. Mathematically, the LDKF problem is dual to the LLQR problem described in the previous sections in this chapter. Therefore, the LDKF has the following desirable properties: (1) each local estimator only needs to collect the information within a localized region to estimate its local state, and (2) each local estimator can be designed by solving a local optimization problem using local plant model information. The decomposition of the global problem into local subproblems thus allows for the method to scale to arbitrarily large heterogeneous systems — this is clearly an extremely favorable property for large-scale estimation problems.

We begin by reviewing some literature in the community of distributed Kalman filtering, and show that these methods do not scale gracefully for large-scale estimation problems. We then formulate the LDKF problem for state estimation, and apply the results from the previous sections to solve the LDKF problem in a scalable way.

5.6.1 Motivation of LDKF

The celebrated Kalman filter achieves minimum mean square error for linear state estimation problems via an elegant and easily interpretable recursive method. Unfortunately, the Kalman filter is an inherently centralized method, and is neither scalable to compute nor physically implementable for large-scale systems. Specifically, the computation of the traditional Kalman filter involves solving an Algebraic Riccati Equation (ARE) and computing a matrix inverse, which is complicated when the size of the problem goes large. Even if a centralized estimator can be computed, large-scale estimation problems are nonetheless subject to practical communication delays between sensors and estimators which can degrade the performance of a centralized scheme substantially. These limitations make centralized estimation unappealing in large-scale applications such as weather forecasting [18], ocean data assimilation [20], biological signal analysis [36], and state estimation in the power grid [24].

Various methods have been proposed in the field of distributed Kalman filtering, but many still suffer from scalability issues that limit their application to large-scale systems. For instance, both the consensus-based algorithm of [47] and the diffusion-based algorithm of [7] require each local sensor to store and use the *global plant model*, and to estimate the *global state* during implementation. This introduces a huge computational burden, and is prohibitive for large-scale applications. An exception is the work of [26], in which the authors use spatial decomposition, observation fusion, and approximated algorithms on matrix inversion to design a scalable Kalman-like filter. However, the algorithms involve multiple iterations, and the transient behavior of the algorithm is hard to analyze.

The motivation of the LDKF framework is to facilitate the scalability of estimator synthesis and implementation for large-scale systems. Our main technical tool is to cast the LDKF problem as a dual problem to LLQR, then solve the LDKF problem using the algorithm proposed in the previous sections.

5.6.2 Traditional Kalman Filter

We adopt to the common setting for Kalman filter (cf. [1]). Consider a discrete time LTI system with dynamics given by

$$\begin{aligned} x[k+1] &= Ax[k] + B_2 u[k] + \delta_x[k] \\ y[k] &= C_2 x[k] + \delta_y[k], \end{aligned} \tag{5.35}$$

where x is the state, u the control input, y the sensor measurements, δ_x the process noise, and δ_y the sensor noise. Our goal is to design a state estimator $\hat{x}$ based on the measurement y and a pre-specified control input u. Similar to the setting of LLQR, we are interested in the case when the system matrices (A, B_2, C_2) are high-dimensional yet suitably sparse. Our approach is to exploit the sparsity of (A, B_2, C_2) to derive a scalable algorithm for state estimator design.

We assume that the process noise δ_x and sensor noise δ_y are independent zero mean AWGN, with covariance matrix given by

$$\mathbb{E}(\begin{bmatrix} \delta_x[i] \\ \delta_y[i] \end{bmatrix} \begin{bmatrix} \delta_x[j] \\ \delta_y[j] \end{bmatrix}^\top) = \begin{cases} \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} & \text{if } i = j \\ 0 & i \neq j. \end{cases} \tag{5.36}$$

We assume that δ_x and δ_y are uncorrelated to keep the formulas simple, while the method described in this section still works when δ_x and δ_y are correlated. The initial condition $x[0]$ is also assumed to be a Gaussian random vector with mean x_0 and variance Σ_0, and $x[0]$ is uncorrelated with $\delta_x[k]$ and $\delta_y[k]$ for all k.

Let $\hat{x}[k|\tau]$ denote the estimate of the state $x[k]$ given the collected information up to time τ, i.e. the measurements $y[t]$ and control inputs $u[t]$ from $t = 1, \ldots, \tau$. The Kalman filter for the LTI system (5.35) is specified by

$$\hat{x}[k|k] = \hat{x}[k|k-1] + K(y[k] - C_2 \hat{x}[k|k-1]) \tag{5.37}$$

$$\hat{x}[k+1|k] = A\hat{x}[k|k] + B_2 u[k] \tag{5.38}$$

with initial condition given by $\hat{x}[0|-1] = x_0$. The matrix K in (5.37) is known as the Kalman gain, which can be found by solving an ARE. Let Σ be the solution to the discrete time ARE

$$\Sigma = A\Sigma A^\top + W - A\Sigma C_2^\top (C_2 \Sigma C_2^\top + V)^{-1} C_2 \Sigma A^\top. \tag{5.39}$$

The Kalman gain in (5.37) can then be computed as

$$K = \Sigma C_2^\top (C_2 \Sigma C_2^\top + V)^{-1}. \tag{5.40}$$

The Kalman filter is optimal in the sense of minimum mean square error. Let $\tilde{x}[k|k-1] = x[k] - \hat{x}[k|k-1]$ be the estimation error before $y[k]$ is measured. The Kalman filter algorithm in (5.37) - (5.38) minimizes the mean square error

$$\mathbb{E}\left(\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \tilde{x}[k|k-1]^\top \tilde{x}[k|k-1]\right). \tag{5.41}$$

Similarly, let $\tilde{x}[k|k] = x[k] - \hat{x}[k|k]$ be the estimation error after $y[k]$ is measured. The mean square error of $\tilde{x}[k|k]$ is also minimized.

Equations (5.37) and (5.38) can be combined into a single equation as

$$\hat{x}[k+1|k] = A\hat{x}[k|k-1] + Bu[k] + H(y[k] - C_2\hat{x}[k|k-1]) \tag{5.42}$$

with $H = AK$ is a gain matrix. We refer to (5.42) as the *delayed form* of state estimation. We can also combine equations (5.37) and (5.38) to obtain

$$\hat{x}[k+1|k+1] = (I - KC_2)(A\hat{x}[k|k] + B_2u[k]) + Ky[k+1]. \tag{5.43}$$

We refer to (5.43) as the *current form* of state estimation.

Limitations

Here we point out some limitations of the traditional Kalman filter for large-scale systems:

1. **Communication delays:** The Kalman gain given by (5.40) is generally dense even when the system matrices (A, B_2, C_2) that specify the system dynamics (5.35) are sparse. This means that the measurements from all sensors need to be shared *instantaneously*, which requires infinite (or impractically fast) communication speed.

2. **Scalability of estimator implementation:** A dense Kalman gain (5.40) also implies that the measurements from all sensor need to be collected by *every estimator in the network*, which is not scalable to implement.

3. **Scalability of estimator synthesis:** To compute the Kalman gain (5.40), one need to solve a large-scale ARE (5.39). The complexity of solving (5.39) is $O(n^3)$, where n is the dimension of the matrix A. This can be prohibitive for a large n.

4. **Scalability of estimator re-synthesis subject to model change:** When the global plant model (A, B_2, C_2) changes locally, one needs to recompute the solution to (5.39) to resynthesize the global Kalman filter. This is not scalable for incremental design when the physical system expands.

To design a state estimation algorithm for large-scale systems, one must overcome the aforementioned limitations of the traditional Kalman filter. Distributed Kalman filter architectures in [47] or [7] may resolve the first two limitations, but not the latter two. This motivates our development of the LDKF architecture, in which the estimator can be both implemented *and* designed in a localized and scalable way.

5.6.3 LDKF Formulation

We now use the system response to analyze the estimation error dynamics of the Kalman filter.

Delayed Form LDKF

Consider first the delayed form (5.42) of the Kalman filter. Taking the z-transform of equation (5.42), we get

$$(zI - A)\hat{\mathbf{x}} = B_2\mathbf{u} + H(\mathbf{y} - C_2\hat{\mathbf{x}}). \tag{5.44}$$

In the following, we assume that the estimator structure (5.44) is fixed, but the gain matrix H is unknown and needs to be designed. Although the Kalman filter can be implemented via a static gain H, this is not necessary. We relax the static gain H to be a proper transfer matrix $\mathbf{H}$ and rewrite (5.44) as

$$(zI - A)\hat{\mathbf{x}} = B_2\mathbf{u} + \mathbf{H}(\mathbf{y} - C_2\hat{\mathbf{x}}) \tag{5.45}$$

in the sequel. This extra freedom will be key in allowing us to incorporate spatiotemporal constraints on the transfer matrices that define the estimator. Combining (5.45) and (5.35), we have the estimation error dynamics

$$(zI - A + \mathbf{H}C_2)\tilde{\mathbf{x}} = \boldsymbol{\delta}_x - \mathbf{H}\boldsymbol{\delta}_y. \tag{5.46}$$

Define $\mathbf{R} := (zI - A + \mathbf{H}C_2)^{-1}$ to be the system response from process noise $\boldsymbol{\delta}_x$ to estimation error $\tilde{\mathbf{x}}$, and likewise let $\mathbf{N} := -\mathbf{R}\mathbf{H}$ be the system response from sensor noise $\boldsymbol{\delta}_y$ to estimation error $\tilde{\mathbf{x}}$. Equation (5.46) can then be rewritten as

$$\tilde{\mathbf{x}} = \mathbf{R}\boldsymbol{\delta}_x + \mathbf{N}\boldsymbol{\delta}_y. \tag{5.47}$$

Rather than finding a suitable gain matrix $\mathbf{H}$ to optimize the system response $(\mathbf{R}, \mathbf{N})$ indirectly, we instead characterize the set of stable achievable system response $(\mathbf{R}, \mathbf{N})$ and optimize *directly* over those sets. Note that the characterization of the set $(\mathbf{R}, \mathbf{N})$ can be obtained by applying the dual of Theorem 1 in Section 3.1, which gives the following lemma.

Lemma 12. *The system response* $(\mathbf{R}, \mathbf{N})$ *with finite mean square error can be induced by an estimator with structure (5.45) if and only if the following two affine constraints hold:*

$$\begin{bmatrix}\mathbf{R} & \mathbf{N}\end{bmatrix}\begin{bmatrix}zI - A \\ -C_2\end{bmatrix} = I \tag{5.48}$$

$$\begin{bmatrix}\mathbf{R} & \mathbf{N}\end{bmatrix} \in \frac{1}{z}\mathcal{RH}_\infty. \tag{5.49}$$

Proof. To prove the necessary direction, we show that (5.48) and (5.49) must hold for an estimator with structure (5.45) and suitable gain matrix $\mathbf{H}$. Equation (5.48) can be verified using the identity $\mathbf{R}(zI - A + \mathbf{H}C_2) = I$ directly. For (5.49), note that the transfer matrices $\mathbf{R}$ and $\mathbf{N}$ must be stable so that the mean square error of the estimator is finite. Besides, as $\mathbf{R} = (zI - A + \mathbf{H}C_2)^{-1}$ and $\mathbf{N} = -\mathbf{RH}$, the transfer matrices $\mathbf{R}$ and $\mathbf{N}$ must be strictly proper, i.e., $R[0] = 0$ and $N[0] = 0$. Therefore, (5.49) must holds.

To prove the sufficient direction, we show that the desired system response $(\mathbf{R}, \mathbf{N})$ can be induced by an estimator with structure (5.45) if (5.48) and (5.49) hold. For any solution of (5.48) - (5.49), we construct a gain matrix $\mathbf{H}_0 = -\mathbf{R}^{-1}\mathbf{N}$ for (5.45). In this case, the estimation error dynamics (5.46) become

$$(zI - A - \mathbf{R}^{-1}\mathbf{N}C_2)\tilde{\mathbf{x}} = \boldsymbol{\delta}_x + \mathbf{R}^{-1}\mathbf{N}\boldsymbol{\delta}_y. \tag{5.50}$$

Multiplying $\mathbf{R}$ to both sides of (5.50) and substituting (5.48) into the equation, we can show that the desired system response $\tilde{\mathbf{x}} = \mathbf{R}\boldsymbol{\delta}_x + \mathbf{N}\boldsymbol{\delta}_y$ is achieved. □

Lemma 12 suggests that we can implement the estimator using the gain matrix $\mathbf{H} = -\mathbf{R}^{-1}\mathbf{N}$ to achieve the desired system response. Substituting this identity back to (5.45) and multiplying $\mathbf{R}$ to both sides of the equation, we get

$$\hat{\mathbf{x}} = \mathbf{R}B_2\mathbf{u} - \mathbf{N}\mathbf{y}. \tag{5.51}$$

This gives a simpler estimator implementation.

Given this characterization of valid system response, we now aim to find an expression for the Kalman filter objective function in terms of **R** and **N**. Using the error dynamics (5.47) and the AWGN assumptions on the noise dynamics, it is straightforward to show that the Kalman filter is the optimal solution to the following SLS problem

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{N}\}}{\text{minimize}} \quad & \| \begin{bmatrix} \mathbf{R} & \mathbf{N} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & (5.48) - (5.49) \end{aligned} \tag{5.52}$$

with

$$\begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} = \begin{bmatrix} W^{\frac{1}{2}} & 0 \\ 0 & V^{\frac{1}{2}} \end{bmatrix}.$$

We can view optimization problem (5.52) as an alternative formulation for the Delayed Form of the Kalman filter problem.

In order to enhance the scalability of estimator design and implementation, we impose an additional d-localized SLC $\mathcal{L}_d$, a FIR SLC $\mathcal{F}_T$, and a QI information sharing SLC C on the system response. This leads to the Delayed Form LDKF given by

$$\begin{aligned} \underset{\mathbf{R},\mathbf{N}}{\text{minimize}} \quad & \| \begin{bmatrix} \mathbf{R} & \mathbf{N} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & (5.48) - (5.49) \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \end{bmatrix} \in \mathcal{L}_d \cap \mathcal{F}_T \cap C. \end{aligned} \tag{5.53}$$

Problem (5.53) is just the transpose of a LLQR problem. Therefore, we can solve (5.53) using the method described in the previous sections in a localized and scalable way. Specifically, we first perform a row-wise separation to decompose (5.53) into row-wise LDKF subproblems. For each LDKF subproblem, the d-localized constraint $\mathcal{L}_d$ further allows us to reduce the dimension of each LDKF subproblem from *global* scale to *local* scale. Each LDKF subproblem in the reduced dimension can then be solved by a local optimization problem using local plant model information only. In particular, the complexity of solving a LDKF subproblem is independent with the size of the global system. In other words, our method can scale to systems of arbitrary size if parallel computation is available.

Remark 16. *In* (5.53)*, we try to find a localized system response for the state estimation error. It should be noted that we are* not *localizing the effect of the process/sensor noise on the state vector. Rather, we are localizing the* state estimation error *due to the noise.*

Remark 17. *It should be noted that the localized synthesis method is valid for arbitrary noise covariance matrices* (W, V)*, i.e., even when the noise is globally correlated. In addition, when the system matrices* (A, B_2, C_2) *change locally, we only need to resolve some of the row-wise LDKF subproblems and update the estimator locally. It follows that this allows for the incremental addition of new subsystems to the global system without the need for a complete redesign of the estimator.*

Current Form LDKF

Consider the current form of Kalman filter in (5.43). Taking the z-transform of (5.43) and rearranging some terms, we have

$$(zI - A)\hat{\mathbf{x}} = -KC_2A\hat{\mathbf{x}} + (I - KC_2)B_2\mathbf{u} + zK\mathbf{y}.$$

We relax the static Kalman gain K to a proper transfer matrix $\mathbf{K}$ in the sequel. We then have

$$\begin{aligned}(zI - A)\hat{\mathbf{x}} &= -\mathbf{K}C_2A\hat{\mathbf{x}} + (I - \mathbf{K}C_2)B_2\mathbf{u} + z\mathbf{K}\mathbf{y} \\ &= -\mathbf{K}C_2A\hat{\mathbf{x}} + (I - \mathbf{K}C_2)B_2\mathbf{u} + z\mathbf{K}(C_2\mathbf{x} + \boldsymbol{\delta}_y) \\ &= -\mathbf{K}C_2A\hat{\mathbf{x}} + (I - \mathbf{K}C_2)B_2\mathbf{u} + z\mathbf{K}\boldsymbol{\delta}_y + \mathbf{K}C_2(z\mathbf{x}) \\ &= -\mathbf{K}C_2A\hat{\mathbf{x}} + (I - \mathbf{K}C_2)B_2\mathbf{u} + z\mathbf{K}\boldsymbol{\delta}_y + \mathbf{K}C_2(A\mathbf{x} + B_2\mathbf{u} + \boldsymbol{\delta}_x) \\ &= \mathbf{K}C_2A\tilde{\mathbf{x}} + B_2\mathbf{u} + z\mathbf{K}\boldsymbol{\delta}_y + \mathbf{K}C_2\boldsymbol{\delta}_x. \end{aligned} \tag{5.54}$$

Combining (5.54) with (5.35), we have

$$(zI - A)\tilde{\mathbf{x}} = -\mathbf{K}C_2A\tilde{\mathbf{x}} + (I - \mathbf{K}C_2)\boldsymbol{\delta}_x - z\mathbf{K}\boldsymbol{\delta}_y.$$

The system response from noise to the estimation error are therefore given by

$$\begin{aligned}\mathbf{R} &= (zI - A + \mathbf{K}C_2A)^{-1}(I - \mathbf{K}C_2) \\ \mathbf{N} &= -(I - \frac{1}{z}(I - \mathbf{K}C_2)A)^{-1}\mathbf{K}. \end{aligned} \tag{5.55}$$

For the characterization of all valid system response, we still have the identity (5.48). The constraint in (5.49) changes slightly however, as the transfer matrix from sensor noise to estimation error **N** is only restricted to be proper in the Current Form of the Kalman filter. The estimator equation in (5.51) remains unchanged. The Current Form LDKF is then given by

$$\begin{aligned}
\underset{\mathbf{R},\mathbf{N}}{\text{minimize}} \quad & \| \begin{bmatrix}\mathbf{R} & \mathbf{N}\end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \| \\
\text{subject to} \quad & (5.48) \\
& \mathbf{R} \in \frac{1}{z}\mathcal{RH}_\infty, \ \mathbf{N} \in \mathcal{RH}_\infty \\
& \begin{bmatrix}\mathbf{R} & \mathbf{N}\end{bmatrix} \in \mathcal{L}_d \cap \mathcal{F}_T \cap \mathcal{C}. \qquad (5.56)
\end{aligned}$$

Similar to the Delayed Form LDKF, the Current Form LDKF (5.56) can also be solved in a localized way using the LLQR algorithm.

5.6.4 Localized Estimator Implementation

After solving the LDKF problem, the LDKF estimator can be implemented using the transfer matrix form (5.51) to achieve the desired system response. When a d-localized system response $(\mathbf{R}, \mathbf{N})$ exists, the implementation (5.51) is localized and thus scalable. This is indicated by the sparsity pattern of each row of **R** and **N** in (5.51) — each component of the state estimate $\hat{\mathbf{x}}$ can be computed by collecting only some components of the measurement **y** and the control action **u**.

Another scalable way to implement the LDKF estimator is given by

$$\begin{aligned}
z\hat{\mathbf{x}} &= A\hat{\mathbf{x}} + B_2\mathbf{u} - \boldsymbol{\beta} \\
\boldsymbol{\beta} &= (I - z\mathbf{R})\boldsymbol{\beta} + z\mathbf{N}(y - C_2\hat{\mathbf{x}}). \qquad (5.57)
\end{aligned}$$

Here, $(I - z\mathbf{R})$ is strictly proper and $z\mathbf{N}$ is proper, so the estimator structure is causal and well-defined. It can be shown that (5.57) is equivalent to (5.45) for $\mathbf{H} = -\mathbf{R}^{-1}\mathbf{N}$. As **R** and **N** are localized transfer matrices, the implementation in (5.57) is localized and thus scalable. The benefit of (5.57) over (5.51) is that (5.57) is compatible with the form of an extended Kalman filter, which provides a possible approach to extend our methods to nonlinear systems.

5.7 Simulation Results

In this section, we perform some numerical simulation to demonstrate the usefulness of the LLQR approach. We begin with a simple stylized chain model, and use the LLQR method to explore tradeoffs with respect to plant model parameters, closed loop performance, implementation complexity, and robustness. The simulation results suggest that we can usually find a fairly sparse localized constraint $\mathcal{L}_d$ such that the LLQR controller achieves similar transient performance compared to the centralized LQR optimal controller, but far more superior in terms of the scalability of controller synthesis and implementation.

Then, we demonstrate the scalability of LLQR controller on a randomized heterogeneous power network example with 51200 states. We show that the LLQR controller can be computed in 23 minutes using a personal computer without using parallel computation. In addition, we use Algorithm 2 to compute a LLQR controller with at least 99% optimality guarantee in 38 minutes for the same example. In contrast, the theoretical computation time for the centralized LQR is more than 200 days using the same computer, and the distributed scheme is simply intractable.

5.7.1 Chain Model

We use a 100 node bi-directional chain with scalar subsystems for this numerical study. The dynamics of each subsystem i is given by

$$x_i[t+1] = \alpha(x_i[t] + \kappa x_{i-1}[t] + \kappa x_{i+1}[t]) + b_i u_i[t] + w_i[t]$$

for $i = 1, \cdots 100$, with $x_0 = x_{101} = 0$ as the boundary value. We can vary the parameter κ to adjust the coupling strength between subsystems, and vary the parameter α to adjust the instability of the open loop system. The value b_i is given by 1 if there is an actuator at subsystem i, and 0 otherwise. We place 40 actuators in the chain network, with actuator location specified by $i = 5j - 4$ and $5j$ for $j = 1, \cdots, 20$. The objective function is given in the form of $||\mathbf{x}||_2^2 + \gamma||\mathbf{u}||_2^2$, where γ is the relative penalty between state deviation and control effort. If we choose $\gamma = 0$ and have $b_i = 1$ for all i, then this example reduces to Example 1 in Section 2.4. We first choose $\kappa = \gamma = 1$, and adjust α to make the spectral radius of A be 1.1 for the simulated example. Note that we use a plant with uniform, symmetric parameters and topology just for the convenience of illustration.

Performance vs. controller complexity

We solve the LLQR problem (5.7), and study the effects of choosing different sized localized regions d and the length of the FIR horizon T. As shown in Figure 5.8, appropriate choices of the parameters (d, T) lead to no degradation in performance with respect to a centralized optimal controller, while leading to significant improvements in synthesis and implementation complexity. If we choose

$(d, T) = (4, 15)$, then there is only 0.3% performance degradation compared to the centralized optimal one, which corresponds to $(d, T) = (99, \infty)$.

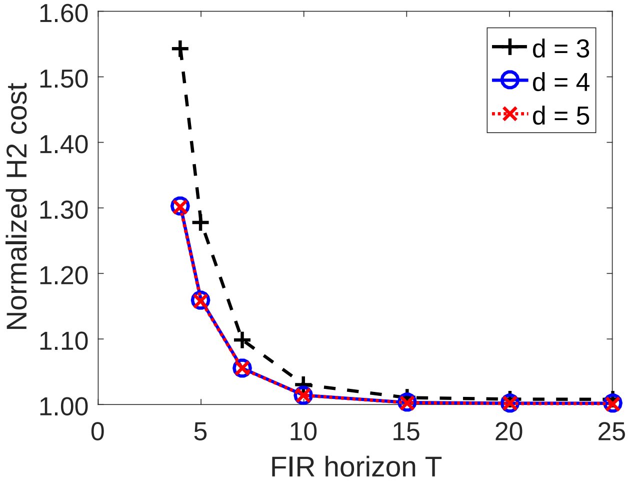


Figure 5.8: Performance vs. FIR horizon T and localized region d. The cost is normalized with respect to the optimal centralized $\mathcal{H}_2$ cost.

The result shown in Figure 5.8 holds for a wide range of the parameters (κ, α, γ) of the plant. Specifically, we vary the coupling strength κ from 0.1 to 1, adjust the instability of the system from 1.1 to 3, change the relative penalty γ from 10^{-6} to 10^{6}, and change the actuator density from 20% to 100% — all the results are qualitatively similar to the one shown in Figure 5.8.

Performance vs. communication delay

We next consider the LLQR problem with a spatiotemporal SLC (5.15), and study the effects of communication delay on the performance by varying t_c in (5.15). We choose $(d, T) = (7, 20)$ for the previous example, and study the tradeoff between communication delay t_c and the normalized $\mathcal{H}_2$ cost (i.e., the square root of the LQR cost). As shown in Figure 5.9, communication delay only leads to slight degradation in performance. Note that the degradation is mostly contributed by the delay constraint. To verify this claim, we compare our localized controller with the QI optimal controller on a 40-state chain example — the QI method cannot scale to the 100-state example due to both memory issue and long computation time. Simulation shows that the localized FIR constraint $(d, T) = (7, 20)$ only leads to 0.03% degradation compared to a QI optimal controller with the same delay constraint.

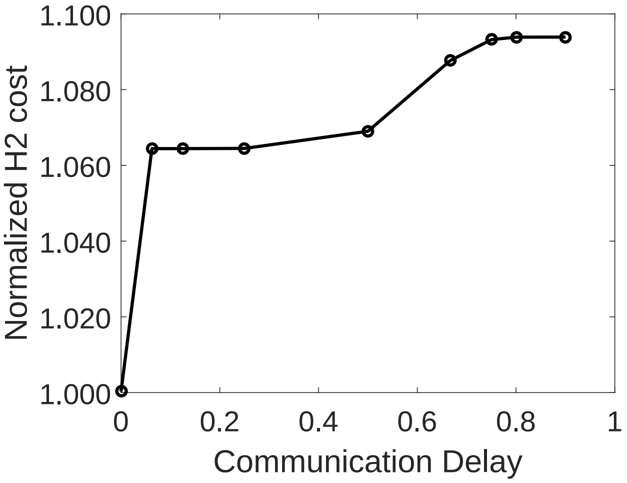


Figure 5.9: Performance vs. communication speed ($(d, T) = (12, 20)$ for $t_c = 0.9$, $(d, T) = (7, 20)$ for the rest). The cost is normalized with respect to the optimal centralized $\mathcal{H}_2$ cost.

5.7.2 Power System Model

We begin with a 20×20 mesh topology representing the interconnection between subsystems, and drop each edge with probability 0.2. The resulting interconnected topology is shown in Figure 5.10 — we assume that all edges are undirected. The dynamics of each subsystem is given by the discretized swing equation for power network. Consider the swing dynamic equation

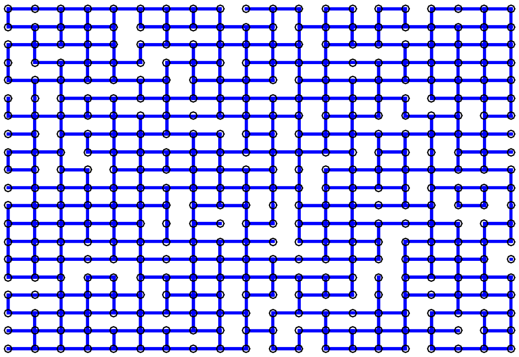


Figure 5.10: Interconnected topology for the simulation example

$$m_i \ddot{\theta}_i + d_i \dot{\theta}_i = - \sum_{j \in \mathcal{N}_i} k_{ij} (\theta_i - \theta_j) + w_i + u_i, \tag{5.58}$$

where θ_i, $\dot{\theta}_i$, m_i, d_i, w_i, u_i are the phase angle deviation, frequency deviation, inertia, damping, external disturbance, and control action of the controllable load of bus i. The coefficient k_{ij} is the coupling term between buses i and j. We let

$x_i := [\theta_i \quad \dot{\theta}_i]^\top$ be the state of bus i and use $e^{A\Delta t} \approx I + A\Delta t$ to discretize the swing dynamics. Equation (5.58) can then be expressed in the form of (5.1) with

$$A_{ii} = \begin{bmatrix} 1 & \Delta t \\ -\frac{k_i}{m_i}\Delta t & 1 - \frac{d_i}{m_i}\Delta t \end{bmatrix}, \quad A_{ij} = \begin{bmatrix} 0 & 0 \\ \frac{k_{ij}}{m_i}\Delta t & 0 \end{bmatrix},$$

and $B_{ii} = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$. The parameters k_{ij}, d_i, m_i^{-1} are randomly generated and uniformly distributed between 0.2 and 1. In addition, we set $\Delta t = 0.2$ and $k_i = \sum_{j \in \mathcal{N}_i} k_{ij}$.

We consider the LLQR problem with a spatiotemporal SLC (5.15) enforced as follows. The localized region for each subsystem j is specified by its two-hop neighborhood.[3] This means that each subsystem can only communicate up to its two-hop neighbors during implementation, and use the plant model up to its two-hop neighbors for controller synthesis. For communication delays, we assume that $u_i[t]$ can access $x_j[\tau]$ at time $\tau \leq t - k$ if subsystem (i, j) are k-hop neighbors. As the disturbance takes two steps to propagate to its neighboring subsystems, the communication speed is twice faster than the speed of disturbance propagation. We impose the FIR constraint with length $T = 15$.

Large-Scale Example

We now allow the size of the mesh network to vary and compare the computation time needed to synthesize a centralized, distributed, and localized LQR optimal controller. The distributed LQR controller is computed using the methods described in [29], in which we assume the same communication delay constraints as LLQR. The empirical relationship obtained between computation time and problem size for the different control schemes is illustrated in the log-log plot in Figure 5.11. For the LLQR controller, we plot both the total computation time and the average computation time per subsystem. As can be seen in Figure 5.11, the computation time needed for the distributed controller grows rapidly when the size of problem increases. For the centralized controller, the slope of the log-log plot in Fig. 5.11 is 3, which matches the theoretical complexity of $O(n^3)$. The slope for the LLQR (total time) is approximately 1.27, which is larger than the theoretical value 1. This overhead is likely caused by other computational issues such as memory management and data copy. For the largest example that we computed, we are able to synthesize

[3] For the mesh network, the number of subsystems contained within the two-hop neighborhood region of a subsystem is up to 13. Since each subsystem has 2 scalar states, the number of states for each localized region is up to 26.

a LLQR optimal controller for a system with 51200 states in about 23 minutes using a personal computer. If the computation were to be parallelized across the 25600 subsystems in the large-scale network (as would be done in a practical situation), the synthesis procedure can be performed in under 0.1 second. In contrast, the theoretical computation time for the centralized LQR using the same computer is more than 200 days, and the distributed LQR is intractable.

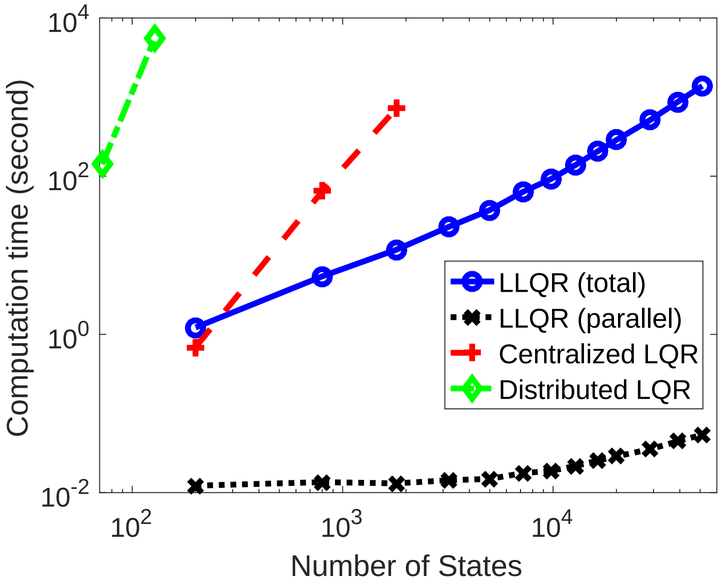


Figure 5.11: Computation time for the centralized, distributed, and localized LQR

Large-Scale Example with Sub-optimality Guarantee

In order to quantify the sub-optimality of the LLQR controller for this large-scale example, we use Algorithm 2 to adaptively update the LLQR spatiotemporal constraint according to the following rule — if the ratio of the lower bound to the upper bound is less than 98% for a particular subsystem, then the subsystem adaptively increases the FIR length T to 20 for itself.

We then compute the LLQR controller and the lower bound for the 51200-state system described above. Among the 25600 subsystems, 156 subsystems increase the FIR length from 15 to 20. The ratio of the overall lower bound to the overall upper bound is 99.01%. This means that the LLQR controller, despite the additional communication delay constraint, localized region constraint, and FIR constraint, is guaranteed to be at least 99.01% optimal compared to the unconstrained centralized LQR optimal controller. The computation of the LLQR controller and its lower bound is finished in 38 minutes. If the computation is parallelized into 25600 subsystems and each subsystem is capable of solving its subproblem using a personal computer, then the LLQR controller with performance guarantee can be computed within 0.1 seconds. In Table 5.2, we summarize the comparison between centralized, distributed and localized LQR for this 51200-state example. It is clear that LLQR has superior performance over centralized and distributed LQR in terms of the scalability of controller synthesis and implementation.

Table 5.2: Comparison Between Centralized, Distributed and Localized LQR on a 51200-State Randomized Example

		LQR	Distributed	LLQR
Closed Loop	Affected region	Global	Global	2-hop
	Affected time	Long	Long	20 steps
	Quadratic Cost	1	1.01	1.01
Synthesis	Comp. complexity	$O(n^3)$	$> O(n^3)$	$O(n)$
	Parallel complexity	$O(n^3)$	$> O(n^3)$	$O(1)$
	Comp. time	200 days	Inf	38 mins
	Parallel time	200 days	Inf	0.1 second
	Plant model	Global	Global	2-hop
	Redesign	Offline	Offline	Real-time
Implementation	Comm. Speed	Inf	2 times	2 times
	Comm. Range	Global	Global	2-hop

Chapter 6

LOCALIZED LINEAR QUADRATIC GAUSSIAN

In this chapter, we discuss the localized linear quadratic Gaussian (LLQG) problem, which is the output feedback version of the LLQR problem. Similar to the LLQR controller, the LLQG controller can also be synthesized and implemented in a localized and scalable way, if the localized SLC is properly specified. The primary difference between LLQR and LLQG is the additional affine constraint (3.15b) in the system level synthesis problem. This constraint breaks the column-wise separability of the LLQG problem, so the LLQR decomposition technique does not directly apply to the output feedback problem. In order to solve the LLQG problem in a scalable way, we use the distributed optimization algorithm such as alternating direction method of multipliers (ADMM) to decouple the LLQG problem into two iterative subroutines. We show that each subroutine can then be solved using the LLQR algorithm described in the previous chapter. This provides a localized yet iterative algorithm to solve the LLQG problem in a scalable way.

This chapter is organized as follows. In Section 6.1, we introduce the output feedback version of the interconnected system model and give the LLQG formulation. In Section 6.2, we apply Theorem 2 in Section 3.2 to show that LLQG controller is scalable to implement. Then, we propose the ADMM algorithm in Section 6.3 to solve the LLQG problem in a localized yet iterative manner. Finally, simulation results are shown in Section 6.4.

6.1 Problem Statement

6.1.1 Interconnected System Model

Here we consider the output feedback version of the interconnected system model. Consider n dynamically coupled discrete time LTI subsystems that interact with each other according to an interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V} = \{1, \dots, n\}$ denotes the set of subsystems. We denote x_i, u_i, and y_i the state vector, control action, and measurement of subsystem i. The set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes the physical interaction between these subsystems — an edge (i, j) is in $\mathcal{E}$ if and only if the state x_j of subsystem j directly affects the state x_i of subsystem i. The dynamics of subsystem

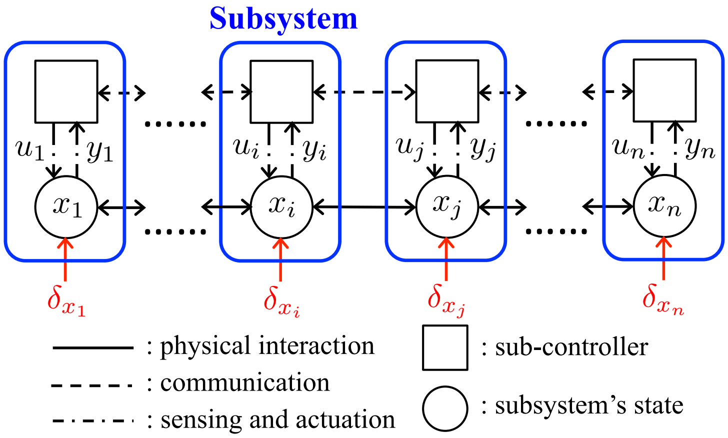


Figure 6.1: An Example of Interconnected System

i is assumed to be given by

$$x_i[t+1] = A_{ii}x_i[t] + \sum_{j\in\mathcal{N}_i} A_{ij}x_j[t] + B_{ii}u_i[t] + \delta_{x_i}[t] \tag{6.1a}$$

$$y_i[t] = C_{ii}x_i[t] + \delta_{y_i}[t], \tag{6.1b}$$

where $\mathcal{N}_i = \{j|(i,j)\in\mathcal{E}\}$ is the (incoming) neighbor set of subsystem i, (A_{ii}, A_{ij}, B_{ii}, C_{ii}) some matrices with compatible dimension, and δ_{x_i} and δ_{y_i} the process disturbance and sensor disturbance, respectively. Comparing to the state feedback problem (5.1), the output feedback problem (6.1) only has noisy and partial measurement of the state (6.1b). This makes the output feedback problem significantly harder than its state feedback counterpart. Figure 6.1 shows an example of such interconnected distributed system — each subsystem i has a sub-controller that takes the locally available measurement y_i, exchanges information with some other sub-controllers through a communication network, and generates the control action u_i to control the state x_i of the physical system.

Define $x = [x_1 \dots x_n]^\top$, $u = [u_1 \dots u_n]^\top$, $y = [y_1 \dots y_n]^\top$, $\delta_x = [\delta_{x_1} \dots \delta_{x_n}]^\top$, and $\delta_y = [\delta_{y_1} \dots \delta_{y_n}]$ the stacked vectors of the subsystem states, controls, measurements, and process and sensor disturbances, respectively. The n interconnected system models (6.1) can be written into a global system model as

$$x[t+1] = Ax[t] + B_2u[t] + \delta_x[t] \tag{6.2a}$$

$$y[t] = C_2x[t] + \delta_y[t], \tag{6.2b}$$

with

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, \quad B_2 = \begin{bmatrix} b_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_{nn} \end{bmatrix}, \quad \text{and} \quad C_2 = \begin{bmatrix} c_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & c_{nn} \end{bmatrix}.$$

Remark 18. *We call the model* (6.1) - (6.2) *a* scalar subsystem model *if x_i, u_i, y_i are scalar variables for all i.*

Remark 19. *Note that the matrices B_2 and C_2 in* (6.2) *are assumed to be diagonal. If the ith subsystem does not have a sensor for measurement, we can simply assign $c_{ii} = 0$. We will lift these assumptions in Chapter 7 to consider the most general interconnected system model with arbitrary sparsity pattern on (A, B_2, C_2).*

We assume that the process disturbance $\boldsymbol{\delta_x}$ and sensor disturbance $\boldsymbol{\delta_y}$ are zero mean i.i.d AWGNs, with covariance matrix given by

$$\mathbb{E}\left(\begin{bmatrix} \delta_x[i] \\ \delta_y[i] \end{bmatrix} \begin{bmatrix} \delta_x[j] \\ \delta_y[j] \end{bmatrix}^\top\right) = \begin{cases} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix}^\top & \text{if } i = j \\ 0 & i \neq j. \end{cases} \tag{6.3}$$

The objective is to find an output feedback control strategy, which is a mapping from measurement $\mathbf{y}$ to control action $\mathbf{u}$, to minimize the expected value of the average quadratic cost (5.3) for some cost matrices (C_1, D_{12}). The traditional infinite horizon stochastic LQG problem can then be formulated as

$$\begin{aligned} \underset{\{x[k],u[k]\}_{k=1}^{\infty}}{\text{minimize}} \quad & \mathbb{E}\Big(\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix}^\top \begin{bmatrix} C_1^\top C_1 & C_1^\top D_{12} \\ D_{12}^\top C_1 & D_{12}^\top D_{12} \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix}\Big) \\ \text{subject to} \quad & (6.2) \text{ and } (6.3). \end{aligned} \tag{6.4}$$

The equivalent SLS formulation of the LQG problem (6.4) is given by

$$\underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \right\|_{\mathcal{H}_2}^2 \tag{6.5a}$$

$$\text{subject to} \quad \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{6.5b}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{6.5c}$$

$$\mathbf{R}, \mathbf{M}, \mathbf{N} \in \frac{1}{z}\mathcal{RH}_\infty, \ \mathbf{L} \in \mathcal{RH}_\infty. \tag{6.5d}$$

Note that (6.5b) - (6.5d) is just the system level parameterization (SLP) for the output feedback system (cf., Theorem 2 in Section 3.2). Since there is no additional system level constraint (SLC) imposed on the system response, we refer to (6.5) as the centralized (unconstrained) LQG problem.

6.1.2 Localized LQG as a SLS Problem

From the discussion of Section 5.1.2, the centralized LQG controller is neither scalable to compute nor scalable to implement for large-scale interconnected systems. In order to enhance the scalability of controller synthesis and implementation, we incorporate an additional d-localized SLC $\mathcal{L}_d$ (we will define this for output feedback system later) and a FIR SLC $\mathcal{F}_T$ into the LQG problem (6.5). This leads to the LLQG problem (cf., Section 4.3.2) given by

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \right\|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & (6.5\text{b}) - (6.5\text{d}) \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in C \cap \mathcal{L}_d \cap \mathcal{F}_T \end{aligned} \tag{6.6}$$

for C a communication delay SLC.

The d-localized SLC $\mathcal{L}_d$ for output feedback system is defined as follows.

Definition 14. *The subspace $\mathcal{L}_d$ is called a d-localized SLC for the output feedback problem* (6.6) *if it constrains the system response* $\mathbf{R}$ *to be d-localized,* $\mathbf{M}$ *and* $\mathbf{N}$ *to be* $(d+1)$*-localized, and* $\mathbf{L}$ *to be* $(d+2)$*-localized.*

The (d, T) output feedback localizability of the system is defined as follows.

Definition 15. *The system* (6.2) *with system matrices* (A, B_2, C_2) *and communication delay SLC C is said to be* (d, T) *output feedback localizable if* (6.6) *is feasible.*

We also make the following assumption throughout the rest of this chapter.

Assumption 3. *At least one of the matrices*

$$\begin{bmatrix} C_1 & D_{12} \end{bmatrix} \ \textit{or} \ \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix}$$

is block-diagonal.

For the case where $\begin{bmatrix} C_1 & D_{12} \end{bmatrix}$ is block-diagonal, the global quadratic cost function (5.3) can be expressed as the sum of local quadratic cost functions. For the case where $\begin{bmatrix} B_1^\top & D_{21}^\top \end{bmatrix}^\top$ is block-diagonal, the process and sensor disturbances at different subsystems are uncorrelated to each other.

Under Assumption 3, the goal of this chapter is to show the following:

1. **Localized controller implementation:** If the system (6.2) is (d, T) output feedback localizable, then the LLQG controller achieving the desired localized system response can be *implemented* in a localized and scalable way. Specifically, the controller at subsystem i can compute its control action u_i by collecting the information (including the measurement and controller's state) within the set $\text{In}_i(d+2)$.

2. **Localized controller synthesis:** The LLQG problem (6.6) can be *solved* in a localized and scalable way if the d-localized SLC and the FIR SLC are properly specified. Specifically, the controller at subsystem i can be synthesized using the plant model contained within the set $\text{In}_i(d+2) \cup \text{Out}_i(d+2)$.

We will discuss localized controller implementation in Section 6.2, and localized controller synthesis in Section 6.3

6.2 Localized Controller Implementation

The localized implementation of the LLQG controller is a direct application of Theorem 2 in Section 3.2. From Theorem 2, the controller achieving the desired localized system response of (6.6) can be implemented by

$$\begin{aligned} z\boldsymbol{\beta} &= \tilde{\mathbf{R}}^+\boldsymbol{\beta} + \tilde{\mathbf{N}}\mathbf{y} \\ \mathbf{u} &= \tilde{\mathbf{M}}\boldsymbol{\beta} + \mathbf{L}\mathbf{y}, \end{aligned} \tag{6.7}$$

with $\tilde{\mathbf{R}}^+ = z\tilde{\mathbf{R}} = z(I - z\mathbf{R})$, $\tilde{\mathbf{M}} = z\mathbf{M}$, $\tilde{\mathbf{N}} = -z\mathbf{N} \in \mathcal{RH}_\infty$. The variable $\boldsymbol{\beta}$ in (6.7) is interpreted as controller's internal state. In (6.7), we share both the measurement $\mathbf{y}$ and controller's internal state $\boldsymbol{\beta}$ during implementation. We then compute the control action $\mathbf{u}$ and the controller's internal state at the next time step. As discussed in Section 3.2, the implementation (6.7) can be considered as a generalization of the state space realization of the controller.

For the solution of the LLQG problem (6.6), the system response $\mathbf{R}$, $\mathbf{M}$, $\mathbf{N}$, and $\mathbf{L}$ are at most $(d+2)$-localized. Note that the transfer matrices $\tilde{\mathbf{R}}^+$, $\tilde{\mathbf{M}}$, and $\tilde{\mathbf{N}}$ are at most $(d+2)$-localized as well. Using the controller implementation (6.7), each subsystem i only need to collect the information $(\mathbf{y}_j, \boldsymbol{\beta}_j)$ of subsystem j from the set $j \in \text{In}_i(d+2)$ to compute the control action $\mathbf{u}_i$ and the controller's internal state $\boldsymbol{\beta}_i$. If the parameter d is significantly smaller than the radius of the interconnected system, then each subsystem only need to collect *local* information during controller implementation — this offers a scalable way to implement the LLQG controller for large-scale systems. Note that the LLQG controller can still be implemented in a localized way even when Assumption 3 does not hold.

6.3 Localized Controller Synthesis

In this section, we propose a scalable algorithm to solve the LLQG problem (6.6) with Assumption 3 in a localized way. Note that the LLQR decomposition technique cannot be applied to optimization problem (6.6) for plants (6.2) corresponding to output feedback problems. This is because the constraints (6.5b) and (6.5c) admit incompatible decompositions: constraint (6.5b) can be decomposed column-wise, whereas constraint (6.5c) can be decomposed row-wise, introducing a coupling between all optimization variables. The ADMM has proven very useful in "breaking" such coupling between optimization variables, allowing for large-scale problems to be decomposed and solved efficiently. Our approach to developing a scalable solution to the LLQG problem (6.6) is to combine the ADMM technique with the LLQR decomposition introduced in the previous chapter.

To reduce notational clutter, we assume that $B_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $D_{21} = \begin{bmatrix} 0 & \sigma_y I \end{bmatrix}$, where σ_y is the relative magnitude between process disturbance and sensor disturbance.[1]Using these values for B_1 and D_{21}, the LLQG problem (6.6) can be written as

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \sigma_y \mathbf{N} \\ \mathbf{M} & \sigma_y \mathbf{L} \end{bmatrix} ||_{\mathcal{H}_2}^2 & \text{(6.8a)} \\ \text{subject to} \quad & \text{(6.5b)} - \text{(6.5d)} & \text{(6.8b)} \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L}_d \cap \mathcal{F}_T. & \text{(6.8c)} \end{aligned}$$

[1] The methods in this section extend in a natural way to the case where $\begin{bmatrix} B_1^\top & D_{21}^\top \end{bmatrix}^\top$ is block-diagonal. By solving the transpose of the LLQG problem, the method can also extend to the case where $\begin{bmatrix} C_1 & D_{12} \end{bmatrix}$ is block-diagonal.

6.3.1 ADMM Algorithm

We now make a series of observations that motivate the use of the ADMM algorithm to solve the LLQG problem (6.8). First, notice that if we remove constraint (6.5c) from problem (6.8), then the resulting optimization problem admits a column-wise LLQR separation, which as described in the previous chapter allows for the global problem to be decomposed into subproblems of size defined by that of the $(d + 2)$-outgoing sets of the subsystems. Through a dual argument, we can show that verifying the feasibility of constraint (6.5c) can be done row at a time, resulting in a feasibility problem that admits a row-wise LLQR separation, once again allowing for the global problem to be decomposed into easily solved subproblems. In order to exploit the decomposition properties of each of these modified problems, we leverage the standard ADMM technique of shifting the coupling from the difficult to enforce constraints (6.5b) and (6.5c) to a simple equality constraint through the introduction of a redundant variable: we make this approach precise in what follows.

We use

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix}$$

to denote the system response that we are solving for. Let $\mathbf{\Psi}$ be a duplicate of the optimization variable $\mathbf{\Phi}$. Following [5], we define the extended-real-value functionals $h^{(r)}(\mathbf{\Phi})$ and $h^{(c)}(\mathbf{\Psi})$ by

$$\begin{aligned} h^{(r)}(\mathbf{\Phi}) &= \begin{cases} 0 & \text{if (6.5c), (6.5d), (6.8c)} \\ \infty & \text{otherwise} \end{cases} \\ h^{(c)}(\mathbf{\Psi}) &= \begin{cases} \text{(6.8a)} & \text{if (6.5b), (6.5d), (6.8c)} \\ \infty & \text{otherwise.} \end{cases} \end{aligned} \tag{6.9}$$

Here, $h^{(r)}(\mathbf{\Phi})$ can be considered as a row-wise separable component of (6.8), and $h^{(c)}(\mathbf{\Psi})$ can be considered as a column-wise separable component of (6.8).

Remark 20. *Note that the constraints* (6.5d) *and* (6.8c) *are included in the definition of both $h^{(r)}(\cdot)$ and $h^{(c)}(\cdot)$. This is a key point to allow the subroutines of the ADMM algorithm to be solvable in a localized way, as shown in the following.*

Using these definitions, we can rewrite the LLQG optimization problem (6.8) as

$$\begin{aligned} \underset{\{\mathbf{\Phi},\mathbf{\Psi}\}}{\text{minimize}} \quad & h^{(r)}(\mathbf{\Phi}) + h^{(c)}(\mathbf{\Psi}) \\ \text{subject to} \quad & \mathbf{\Phi} = \mathbf{\Psi}. \end{aligned} \tag{6.10}$$

The form of optimization problem (6.10) is precisely that needed by the ADMM approach [5], and can be solved via the iteration

$$\mathbf{\Phi}^{k+1} = \underset{\mathbf{\Phi}}{\operatorname{argmin}} \left(h^{(r)}(\mathbf{\Phi}) + \frac{\rho}{2}||\mathbf{\Phi} - \mathbf{\Psi}^k + \mathbf{\Lambda}^k||^2_{\mathcal{H}_2} \right) \tag{6.11a}$$

$$\mathbf{\Psi}^{k+1} = \underset{\mathbf{\Psi}}{\operatorname{argmin}} \left(h^{(c)}(\mathbf{\Psi}) + \frac{\rho}{2}||\mathbf{\Psi} - \mathbf{\Phi}^{k+1} - \mathbf{\Lambda}^k||^2_{\mathcal{H}_2} \right) \tag{6.11b}$$

$$\mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k + \mathbf{\Phi}^{k+1} - \mathbf{\Psi}^{k+1}. \tag{6.11c}$$

Recall that the system response $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are constrained to lie in the FIR subspace $\mathcal{F}_T$, and hence is a finite dimensional variable: it follows that each of the problems specified by the ADMM algorithm (6.11) can be formulated as finite dimensional optimization problems by associating the FIR transfer matrices with their matrix representations. We now focus on the problem specifying the $\mathbf{\Psi}^{k+1}$ iterates (6.11b), which can be written as

$$\begin{aligned}
\underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \sigma_y \mathbf{N} \\ \mathbf{M} & \sigma_y \mathbf{L} \end{bmatrix} ||^2_{\mathcal{H}_2} + \frac{\rho}{2} || \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} - \mathbf{\Phi}^{k+1} - \mathbf{\Lambda}^k ||^2_{\mathcal{H}_2} \\
\text{subject to} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \\
& \mathbf{R}, \mathbf{M}, \mathbf{N} \in \frac{1}{z}\mathcal{RH}_\infty, \ \mathbf{L} \in \mathcal{RH}_\infty \\
& \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L}_d \cap \mathcal{F}_T.
\end{aligned} \tag{6.12}$$

From the form of this problem, it is apparent that an analogous argument to that presented in Chapter 5 applies — we first perform a column-wise separation of the optimization problem (6.12), then exploit the d-localized SLC $\mathcal{L}_d$ to reduce the dimension of each subproblem from global scale to the localized region defined by the $(d + 2)$-outgoing set of each disturbance. Similarly, subproblem (6.11a) admits a row-wise LLQR separation, and the Lagrange multiplier update equation (6.11c) decomposes element-wise. Thus if the ADMM weight ρ is shared between subsystems prior to the synthesis procedure, the optimization problems specifying the ADMM algorithm (6.11) decompose into subproblems specified by the $(d + 2)$-outgoing and $(d + 2)$-incoming sets of the system.

An added benefit of the LLQG framework is the ability to perform real-time re-synthesis of optimal controllers. In particular, suppose that the dynamics (6.1) describing the dynamics of a collection $\mathcal{D}$ of subsections change — in order to suitably update the LLQG optimal controller, only the components of the system response $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ corresponding to the response of subsystems j satisfying $\text{In}_j(d+2) \cap \mathcal{D} \neq \emptyset$ or $\text{Out}_j(d+2) \cap \mathcal{D} \neq \emptyset$ need to be updated.

Next we show that the problems specifying the iterates $\mathbf{\Phi}^{k+1}$ and $\mathbf{\Psi}^{k+1}$ can be solved in *closed form* allowing for the update equations (6.11a) and (6.11b) to be implemented via matrix multiplication. We end the section with a discussion of conditions guaranteeing the convergence of the iterates $\mathbf{\Phi}^{k+1}$ and $\mathbf{\Psi}^{k+1}$ to the optimal solution to the LLQG problem (6.8).

Remark 21. *The ADMM approach specified in* (6.11) *can be used with other objective functions that admit a column-wise separation and/or row-wise separation. We will generalize the application of the algorithm* (6.11) *to a broader class of problems, which is called the convex localized separable SLS (CLS-SLS) problems, in Chapter 7. An interesting special case is that we can solve problem* (6.8) *for arbitrary B_1 and D_{21} if $[C_1 \quad D_{12}]$ is block-diagonal — in particular, this means that a LLQG controller can be synthesized in a scalable way using our proposed algorithm even if the process and sensor noise are globally correlated, so long as the subsystem's performance objectives are decoupled.*

6.3.2 Analytic Solution

We now focus on optimization problem (6.12), which specifies the iterates $\mathbf{\Psi}^{k+1}$. Following the LLQR method described in Section 5.2, we perform a column-wise separation of the objective and constraints of (6.12), and exploit the d-localized SLC of the system response to reduce the dimensionality of each resulting subproblem. Specifically, for each disturbance $\boldsymbol{\delta}_{x_j}$ or $\boldsymbol{\delta}_{y_j}$ at subsystem j, we solve an optimization of the same form as (6.12) except with all decision variables, state-space parameters and constraints restricted to the $(d+2)$-outgoing set of subsystem j. The result is an optimization problem similar to (5.12). We also note that optimization problem (5.12) and the dimensionality reduced version of optimization problem (6.12) are least-squares problems subject to affine constraints. Consequently, the optimal solution is specified as an affine function of the problem data $\mathbf{\Psi}^{k+1}$ and $\mathbf{\Lambda}^k$ in the reduced dimension. The affine function for each dimensionality reduced subproblem only need to be evaluated *once*, after which the updates to the iterates (6.11b) can be performed via multiple matrix multiplication, all in the reduced dimension. We defer a detailed mathematical derivation of the analytic solution of ADMM updates in Appendix 7.C in Chapter 7. Note that the $\mathbf{\Phi}^{k+1}$ iterate (6.11a) can be carried out using matrix multiplication in a similar fashion.

Thus using this approach to solving the iterate updates (6.11a) and (6.11b), the LLQG optimization problem (6.8) can be solved nearly as quickly as the state-feedback problem, as the update equations require first solving a least-squares problem defined on the $(d+2)$-incoming and $(d+2)$-outgoing sets of the system and then using matrix multiplication.

6.3.3 Convergence and Stopping Criteria

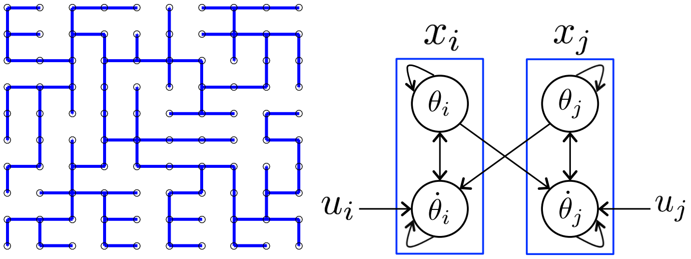
Assume that the optimization problem (6.10) is feasible, and let $\mathbf{\Psi}^*$ be an optimal solution. Further assume that the matrix $[C_1 \quad D_{12}]$ has full column rank, and $[B_1; D_{21}]$ has full row rank. In this case, the objective function is strongly convex with respect to $\mathbf{\Psi}$, and hence any optimal solution $\mathbf{\Psi}^*$ is the unique optimal solution. As the extended-real-value functions $h^{(r)}(\cdot)$ and $h^{(c)}(\cdot)$ specified in (6.9) are closed, proper, and convex, we have that strong duality holds and that optimization problem (6.10) satisfies the convergence conditions state in [5]. From [5], the objective of (6.10) converges to its optimal value. As the objective function is a continuous function of $\mathbf{\Psi}$ and the optimal solution $\mathbf{\Psi}^*$ is unique, it follows that the primal variable iterates converge to $\mathbf{\Psi}^*$, i.e., $\mathbf{\Psi}^k \rightarrow \mathbf{\Psi}^*$ and $\mathbf{\Phi}^k \rightarrow \mathbf{\Psi}^*$. Note that the rank condition on the objective function matrices is only a sufficient condition for primal variable convergence. A less restrictive conditions for the convergence of the ADMM algorithm will be discussed in Appendix 7.B in Chapter 7. The design of the stopping criteria for the ADMM algorithm (6.11) can also be found in Appendix 7.B.

6.4 Simulation Results

In this section, we demonstrate the LLQG algorithm using the power system example (5.58). After introducing the power system model, we discuss how to set the constraints $\mathcal{L}_d$ and $\mathcal{F}_T$ to enhance the scalability of controller implementation and synthesis. We show that the LLQG controller, with additional localized constraint, FIR constraint, and communication delay constraint, can achieve comparable transient performance to a centralized LQG optimal one (the $\mathcal{H}_2$ optimal controller), but far more superior than the centralized and the distributed methods in terms of the scalability of controller synthesis and implementation. Specifically, we synthesize the LLQG controller for a randomized heterogeneous networked system with 12800 states in 22 minutes, where the centralized and the distributed counterparts cannot be computed.

6.4.1 Power System Model

We begin with a randomized spanning tree embedded on a 10×10 mesh network representing the interconnection between subsystems. The resulting interconnected topology is shown in Figure 6.2a — we assume that all edges are undirected. The dynamics of each subsystem is given by the discretized swing equation for power network (5.58). Similar to the model described in Section 5.7, we let $x_i := [\theta_i \quad \dot{\theta}_i]^\top$ be the state of bus i and use $e^{A\Delta t} \approx I + A\Delta t$ to discretize the swing dynamics. Equation (5.58) can then be expressed in the form of (6.1) with



(a) Interconnected topology

(b) Interaction between neighboring subsystems

Figure 6.2: Simulation example interaction graph.

$$A_{ii} = \begin{bmatrix} 1 & \Delta t \\ -\frac{k_i}{m_i}\Delta t & 1 - \frac{d_i}{m_i}\Delta t \end{bmatrix}, \quad A_{ij} = \begin{bmatrix} 0 & 0 \\ \frac{k_{ij}}{m_i}\Delta t & 0 \end{bmatrix}, B_{ii} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ and } C_{ii} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We set $\Delta t = 0.2$ and $k_i = \sum_{j \in \mathcal{N}_i} k_{ij}$. In addition, the parameters k_{ij}, d_i, and m_i^{-1} are randomly generated and uniformly distributed between $[0.5, 1]$, $[1, 1.5]$, and $[0, 2]$, respectively. The instability of the plant is characterized by the spectral radius of the matrix A, which is 1 in the simulated example. The interactions between neighboring subsystems of the discretized model is described by Figure 6.2b. We assume that each subsystem in the power network has a phase measurement unit (PMU), a frequency sensor, and a controllable load that generates u_i.

From (5.58), the external disturbance w_i only directly affects the frequency deviation $\dot{\theta}_i$. To make the objective functional strongly convex, we introduce small artificial disturbance on the phase deviation θ_i as well. We assume that the process noise on frequency and phase are uncorrelated AWGNs with covariance matrices given by I and $10^{-4}I$, respectively. In addition, we assume that both the phase deviation and the frequency deviation are measured with some sensor noise. The sensor noise of phase and frequency measurements are uncorrelated AWGNs with covariance matrix given by $10^{-2}I$. We choose equal penalty on the state deviation and control effort, i.e., $\begin{bmatrix} C_1 & D_{12} \end{bmatrix} = I$.

Based on the above setting, we formulate a $\mathcal{H}_2$ optimal control (LQG) problem that minimizes the $\mathcal{H}_2$ norm of the transfer matrix from the process and sensor noises to the regulated output. The $\mathcal{H}_2$ norm of the closed loop is given by 13.3169 when a proper centralized $\mathcal{H}_2$ optimal controller is applied, and 16.5441 when a strictly proper centralized $\mathcal{H}_2$ optimal controller is applied. In the rest of this section, we normalized the $\mathcal{H}_2$ norm with respect to the proper centralized $\mathcal{H}_2$ optimal controller.

6.4.2 LLQG

The underlying assumption of the centralized optimal control scheme is that the measurement can be transmitted *instantaneously* with *every subsystem* in the network. To incorporate realistic communication delay constraint and facilitate the scalability of controller design, we impose additional communication delay constraint, localized constraint, and FIR constraint on the system response. We introduce these constraints in a sequential order as follows.

For the communication delay constraint $\mathcal{C}$, we assume that each subsystem takes one time step to transmit the information to its neighboring subsystems. Mathematically, the control action $u_i[t]$ of subsystem i at time t can receive $(y_j[\tau], \beta_j[\tau])$ of subsystem j for time $\tau \leq t - k$ if the distance between subsystems i and j is k. The interaction between subsystems illustrated in Figure 6.2b implies that it takes two time steps for a disturbance at subsystem j to propagate to its neighboring subsystems, and hence the communication speed is twice as fast as propagation speed of disturbances through the plant. For the given communication delay constraint $\mathcal{C}$, we use the method described in Section 5.3 to design the sparsest localized constraint $\mathcal{L}$. In this example, we can localize the effect of each process and sensor noise within its two-hop neighbors. This implies that each subsystem j only needs to exchange the information within its two-hop neighbors, and use the restricted plant model within its two-hop neighbors to synthesize its sub-controller.

Once the communication delay constraint $\mathcal{C}$ and the localized constraint $\mathcal{L}$ are specified, we run some simulation to exploit the tradeoff between the length of the FIR constraint $\mathcal{F}_T$ and the transient performance. Figure 6.3 shows the tradeoff curve between the transient performance of the LLQG controller and the length T of the FIR constraint. For the given communication delay constraint $\mathcal{C}$ and the locality constraint $\mathcal{L}$, the LLQG controller is feasible with the FIR constraint $\mathcal{F}_T$ for all $T \geq 3$.

When the length of the FIR constraint increases, the $\mathcal{H}_2$ norm of the closed loop converges quickly to the unconstrained optimal value. For instance, for FIR length $T = 7, 10$, and 20, the performance degradation compared to the unconstrained $\mathcal{H}_2$ optimal controller are given by 3.8%, 1.0%, and 0.1%, respectively. This further means that the performance degradation due to the additional communication delay constraint $\mathcal{C}$ and the localized constraint $\mathcal{L}$ is less than 0.1%. From Figure 6.3, we show that the LLQG controller, with additional communication delay constraint, localized constraint, and FIR constraint, can achieve similar transient performance to an unconstrained optimal $\mathcal{H}_2$ controller.

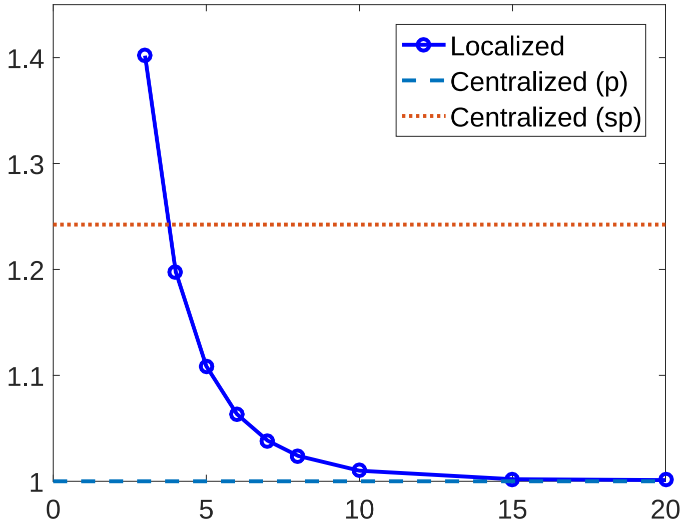


Figure 6.3: The vertical axis is the normalized $\mathcal{H}_2$ norm of the closed loop when the LLQG controller is applied. The LLQG controller is subject to the constraint $\mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T$. The horizontal axis is the horizon T of the FIR constraint $\mathcal{F}_T$, which is also the settling time of the impulse response. We plot the normalized $\mathcal{H}_2$ norm for the centralized unconstrained optimal controller (proper and strictly proper) in the same figure.

To further illustrate the advantages of the LLQG scheme, we choose $T = 20$ and compare the LLQG controller, distributed LQG optimal controller, and the centralized LQG optimal controller in terms of the closed loop performance, the complexity of controller synthesis, and the complexity of controller implementation in Table 6.1. The distributed optimal controller is computed using the method described in [28], in which we assume the same communication constraint $\mathcal{C}$ as the LLQG controller. It can be seen that the LLQG controller is vastly preferable in all aspects, except for a slight degradation in the closed-loop performance. In particular, the localized constraint $\mathcal{L}$ in this example has almost no effect on the closed loop performance.

Table 6.1: Comparison Between Centralized, Distributed, and Localized LQG Optimal Control

		Centralized	Distributed	Localized
Closed Loop	Affected region	Global	Global	2-hop
	Affected time	Long	Long	20 steps
	Normalized $\mathcal{H}_2$	1	1.001	1.001
Synthesis	Comp. complexity	$O(n^3)$	$> O(n^3)$	$O(n)$
	Parallel complexity	$O(n^3)$	$> O(n^3)$	$O(1)$
	Plant model	Global	Global	2-hop
	Redesign	Offline	Offline	Real-time
Implementation	Comm. Speed	Inf	2 times	2 times
	Comm. Range	Global	Global	2-hop

6.4.3 Large-Scale Example

We now allow the size of the problem to vary and compare the computation time needed to synthesize a centralized, distributed, and localized LQG optimal controller. We choose $T = 7$ for the LLQG controller. The empirical relationship obtained between computation time and problem size for different control schemes is illustrated in Figure 6.4. As can be seen in Figure 6.4, the computation time needed for the distributed controller grows rapidly when the size of problem increases. For the centralized one, the slope in the log-log plot in Figure 6.4 is 3, which matches the theoretical complexity $O(n^3)$. The slope for the LLQG controller is about 1.4, which is larger than the theoretical value 1. We believe this overhead is caused by other computational issue such as memory management and data structure. We note that the computational bottleneck that we faced in computing our large-scale example was that we were using a laptop to compute the controller (and hence the localized subproblems were essentially solved in serial) — in practice, if each local subsystem is capable of solving its corresponding localized subproblem, our approach scales to systems of arbitrary size as all computations can be done in parallel. For the largest example we have, we can finish the LLQG synthesis for a system with 12800 states in 22 minutes using a laptop. If the computation is parallelized into all 6400 sub-systems, the synthesis algorithm can be done within 0.2 second. In contrast, the theoretical time to compute the centralized LQG optimal controller for the same example is more than a week, and the distributed LQG optimal controller is intractable.

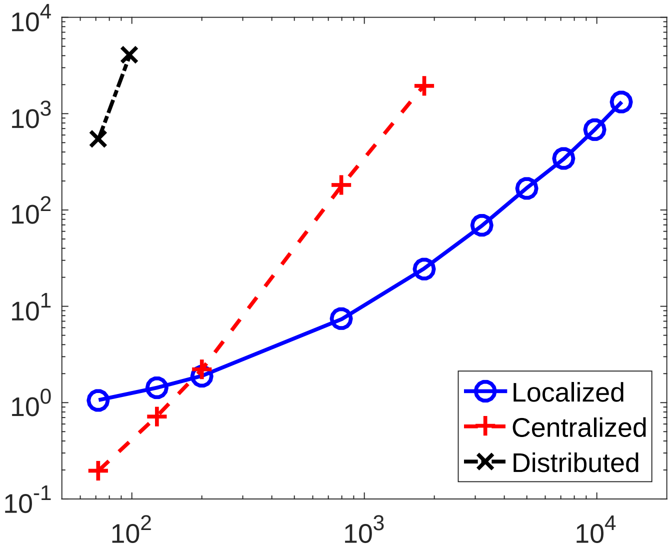


Figure 6.4: Computation time for the centralize, distributed, and localized LQG controller. The horizontal axis denotes the number of states of the system, and the vertical axis is the computation time in seconds.

Chapter 7

SYSTEM LEVEL SYNTHESIS FOR LARGE-SCALE SYSTEMS

In this chapter, we introduce the class of convex localized separable system level synthesis (CLS-SLS) problems, which include localized LQR (cf. Chapter 5) and localized LQG (cf. Chapter 6) as special cases. We propose a unified algorithm to solve all CLS-SLS problems in a localized and scalable way, with $O(1)$ parallel computational complexity compared to the size of the global network. In addition to the scalability of controller synthesis, the controller achieving the desired localized system response can also be implemented in a localized and scalable way — each sub-controller in the network can compute its control action by communicating with $O(1)$ numbers of other sub-controllers during implementation. As a result, the CLS-SLS problems can be scaled to arbitrary large-scale network if parallel computation is available. We give several examples of CLS-SLS problems, including the localized $\mathcal{H}_2$ optimal control problem with sensor actuator regularization and the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control problem.

This chapter is organized as follows. In Section 7.1, we introduce the system model, recall the system level synthesis (SLS) framework introduced in Chapters 3 - 4, and show how to use the convex localized SLS framework to design a localized optimal controller that is scalable to implement. We then point out the key technical condition for a convex localized SLS problem to be solvable in a localized and scalable way — *separability*. In Section 7.2, we introduce the class of column/row-wise separable problems, which is the extension of localized LQR (LLQR) introduced in Chapter 5. We then introduce the general CLS-SLS problems in Section 7.3, and show that CLS-SLS framework is a natural extension of localized LQG (LLQG) introduced in Chapter 6. Finally, simulation results are shown in Section 7.4 to demonstrate the generality and scalability of the CLS-SLS framework.

7.1 Problem Setup

7.1.1 Mathematical Notation

Let $\mathbb{Z}^+$ be the set of all positive integers. We use calligraphic lower case letters such as $\mathcal{r}$ and $\mathcal{c}$ to denote subsets of $\mathbb{Z}^+$. We say that $\{\mathcal{r}_1, \dots \mathcal{r}_m\}$ is a partition of a set $\mathcal{q} \subset \mathbb{Z}^+$ if and only if $\mathcal{r}_i \neq \emptyset$ for all i, $\cup_{i=1}^m \mathcal{r}_i = \mathcal{q}$, and $\mathcal{r}_i \cap \mathcal{r}_j = \emptyset$ for all $i \neq j$.

Consider a transfer matrix $\mathbf{\Phi}$ with n_r rows and n_c columns. Let r be a subset of $\{1, \dots, n_r\}$ and c a subset of $\{1, \dots, n_c\}$. We use $\mathbf{\Phi}(r, c)$ to denote the submatrix of $\mathbf{\Phi}$ by selecting the rows according to the set r and columns according to the set c. We use the symbol : to denote the set of all rows or all columns, i.e., we have $\mathbf{\Phi} = \mathbf{\Phi}(:, :)$. Let $\{c_1, \dots c_p\}$ be a partition of the set $\{1, \dots, n_c\}$. Then $\{\mathbf{\Phi}(:, c_1), \dots, \mathbf{\Phi}(:, c_p)\}$ is a column-wise partition of the transfer matrix $\mathbf{\Phi}$.

A permutation matrix is a square binary matrix that has exactly one entry of 1 in each row and each column and zeros elsewhere.

7.1.2 System Model

We consider a discrete time linear time invariant (LTI) system with dynamics given by

$$x[t+1] = Ax[t] + B_2 u[t] + \delta_x[t] \tag{7.1a}$$

$$y[t] = C_2 x[t] + \delta_y[t], \tag{7.1b}$$

where x, u, y, δ_x, and δ_y are the global state vector, control actions, measurements, and process and sensor disturbances, respectively. We use n_x, n_y, and n_u to denote the dimension of x, y, and u, respectively. All x, u, y, δ_x, and δ_y are large-scale. We assume that the system model (A, B_2, C_2) are large-scale but suitably sparse. Different from the interconnected system model introduced in Chapters 5 and 6, we do not make any assumption on the specific sparsity pattern of (A, B_2, C_2). In particular, the system matrices do not need to be block diagonal. Thus the LTI system model considered in this chapter is as general as possible.

7.1.3 Localized System Level Synthesis

Here we briefly recall the SLS framework introduced in Chapters 3 - 4. For an LTI system with dynamics given by (7.1), we define a system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ to be the maps satisfying

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_x \\ \boldsymbol{\delta}_y \end{bmatrix}. \tag{7.2}$$

From Theorem 2 in Chapter 3, the parameterization of all stable achievable system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ is described by the following set of affine equations:

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{7.3a}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{7.3b}$$

$$\mathbf{R}, \mathbf{M}, \mathbf{N} \in \frac{1}{z}\mathcal{RH}_\infty, \quad \mathbf{L} \in \mathcal{RH}_\infty. \tag{7.3c}$$

In addition, the controller achieving the desired system response (7.2) is given by

$$z\boldsymbol{\beta} = \tilde{\mathbf{R}}^+\boldsymbol{\beta} + \tilde{\mathbf{N}}\mathbf{y} \tag{7.4a}$$

$$\mathbf{u} = \tilde{\mathbf{M}}\boldsymbol{\beta} + \mathbf{L}\mathbf{y}, \tag{7.4b}$$

where $\tilde{\mathbf{R}}^+ = z(I - z\mathbf{R})$, $\tilde{\mathbf{N}} = -z\mathbf{N}$, $\tilde{\mathbf{M}} = z\mathbf{M}$, $\mathbf{L}$ are in $\mathcal{RH}_\infty$. With the parameterization of all stable achievable system response in (7.3a) - (7.3c), we now incorporate a system level objective (SLO) $g(\cdot)$ and a system level constraint (SLC) $\mathcal{S}$ to formulate the SLS problem:

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & g(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}) && (7.5a) \\ \text{subject to} \quad & (7.3a) - (7.3c) && (7.5b) \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{S}. && (7.5c) \end{aligned}$$

The SLS problem (7.5) is *convex* as long as the SLO (7.5a) is convex and the intersection of the SLC (7.5c) and the affine space (7.5b) is convex.

We now express the SLC set $\mathcal{S}$ as the intersection of three convex set components: the localized (sparsity) constraint $\mathcal{L}$, the finite impulse response (FIR) constraint $\mathcal{F}_T$, and an arbitrary convex set component $\mathcal{X}$, i.e., $\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}$. The localized constraint $\mathcal{L}$ imposed on a transfer matrix $\mathbf{G}$ is a collection of sparsity constraints with the form $\mathbf{G}_{ij} = 0$ for some i and j. Here $\mathcal{L}$ can impose *arbitrary* sparsity constraint — in particular, it does not need to be a d-localized SLC as described in Chapters 5 and 6. The constraint $\mathcal{F}_T$ restricts the optimization variables to have finite impulse responses of horizon T, which makes (7.5) a finite dimensional convex program. The constraint $\mathcal{X}$ includes any other convex constraint imposed by the system — in particular, the set $\mathcal{X}$ can be the combination of all kinds of convex SLCs introduced in Section 4.2, which includes the communication delay SLC considered in Chapters 5 and 6 as a special case. This leads to a *localized SLS problem* given by

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad & g(\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}) && \text{(7.6a)} \\ \text{subject to} \quad & \text{(7.3a)} - \text{(7.3c)} && \text{(7.6b)} \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}. && \text{(7.6c)} \end{aligned}$$

A SLS problem is called *localized* if it is in the form of (7.6). We call (7.6) a convex localized SLS if (7.6) is convex.

Remark 22. *For a state feedback problem ($C_2 = I$ and $D_{21} = 0$, cf., Section 3.1), the localized SLS problem* (7.6) *can be simplified into the form of*

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\textit{minimize}} \quad & g(\mathbf{R},\mathbf{M}) \\ \textit{subject to} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \\ & \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X} \cap \frac{1}{z}\mathcal{RH}_\infty. \end{aligned} \qquad \text{(7.7)}$$

In addition, the controller achieving the desired system response can be implemented by the transfer matrices **R** *and* **M** *directly (cf., Theorem 1 in Section 3.1).*

Remark 23. *It should be noted that although the optimization problems* (7.6) *and* (7.7) *are convex for arbitrary localized constraint $\mathcal{L}$, they are not necessarily feasible. In Section 5.3, we showed that a necessary condition for the existence of a localized (sparse) system response is that the communication speed between sub-controllers is faster than the speed of disturbance propagation in the plant. We also offer some guideline to design a feasible localized constraint $\mathcal{L}$ in the same section.*

7.1.4 Localized Implementation

Here we show how to design a controller that is scalable to implement using the localized SLS framework (7.6). Our approach is to impose sparsity constraint on the system response $(\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L})$ through the localized constraint $\mathcal{L}$. Theorem 2 shows how the sparsity of the system response $(\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L})$ translates into the implementation complexity of a controller as in (7.4). For instance, if each row of these transfer matrices are suitably sparse[1], then each sub-controller i only needs to collect a small number of measurements $\mathbf{y}_j$ and controller states $\boldsymbol{\beta}_j$ to compute its control action $\mathbf{u}_i$ and controller state $\boldsymbol{\beta}_i$ (cf. Section 6.2 and [68, 69, 71]). In this case, the complexity to implement each sub-controller, which is measured by the number of required communication links, is independent to the size of the global network, i.e., the parallel implementation complexity is $O(1)$. If the localized SLS problem is convex, then we have a *convex* way to design a controller that is scalable to implement. This holds true for arbitrary convex SLO $g(\cdot)$ and arbitrary convex SLC $\mathcal{X}$. In contrast, for a strongly connected network, any sparsity constraint imposed on the controller makes the distributed optimal control problem (2.8) non-convex (cf. Section 2.4).

7.1.5 Problem Statement: Separability

From the above discussion, we can use the convex localized SLS framework (7.6) to design a localized optimal controller that is scalable to implement. It is then natural to ask if a convex localized SLS problem can also be *solved* in a localized and scalable way, using the approach similar to LLQR and LLQG synthesis (cf., Chapters 5 - 6). It turns out that we need one more property of the SLS problem to support localized synthesis — *separability*. The aim of this chapter is to formally define the notion of separability of the SLS problem, and propose a scalable algorithm to solve all CLS-SLS problems in a localized way. Specifically, if the localized constraint $\mathcal{L}$ of a CLS-SLS problem is suitably specified, the global problem (7.6) can be decomposed into parallel local subproblems with constant complexity. As the parallel computational complexity is $O(1)$, we can solve the CLS-SLS problems for systems with arbitrary large-scale.

7.2 Column/Row-wise Separable Problems

In this section, we consider the state feedback localized SLS problem (7.7), which is a special case of (7.6). We identify the technical conditions on the SLO and the SLC in (7.7) such that the optimization problem (7.7) can be decomposed into parallel local optimization subproblems without using distributed optimization. This technique is crucial for the general algorithm proposed in Section 7.3.

This section begins with the *column-wise separable problems*, which is a natural extension of the LLQR method introduced in Chapter 5. This section ends with

[1] The number of nonzero entries are significantly smaller than the dimension of the transfer matrix.

the highlight of row-wise separable problems as the dual of column-wise separable problems.

7.2.1 Column-wise Separable Problems

The goal of this subsection is to identify the technical conditions on $g(\cdot)$ and $\mathcal{X}$ in (7.7) such that the global optimization problem (7.7) can be solved in a localized and distributed way using the column-wise separation technique similar to that of LLQR. To simplify the notation of (7.7), we use

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix}$$

to represent the system response we want to optimize for, and we denote $\mathbf{Z}_{AB}$ the transfer matrix $\begin{bmatrix} zI - A & -B_2 \end{bmatrix}$. The state feedback localized SLS problem (7.7) can then be written as

$$\begin{aligned} \underset{\mathbf{\Phi}}{\text{minimize}} \quad & g(\mathbf{\Phi}) && (7.8a) \\ \text{subject to} \quad & \mathbf{Z}_{AB}\mathbf{\Phi} = I && (7.8b) \\ & \mathbf{\Phi} \in \mathcal{S}, && (7.8c) \end{aligned}$$

with $\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X} \cap \frac{1}{z}\mathcal{RH}_\infty$.

Similar to the LLQR problem, we note that the matrix variable $\mathbf{\Phi}$ in constraint (7.8b) can be examined column at a time. It is then natural to see if the SLO (7.8a) and the SLC (7.8c) can also be decomposed in a column-wise manner. More generally, we note that it suffices to find a column-wise *partition* of the optimization variable to decompose (7.8) into parallel subproblems. Recall that we use calligraphic lower case letters such as $\mathcal{c}$ to denote subsets of positive integer. Let $\{\mathcal{c}_1, \dots, \mathcal{c}_p\}$ be a partition of the set $\{1, \dots, n_x\}$. The optimization variable $\mathbf{\Phi}$ can then be partitioned column-wisely into the set $\{\mathbf{\Phi}(:, \mathcal{c}_1), \dots, \mathbf{\Phi}(:, \mathcal{c}_p)\}$. The column-wise separability of the SLO (7.8a) is defined as follows.

Definition 16. *The system level objective* $g(\mathbf{\Phi})$ *in* (7.8a) *is said to be* column-wise separable *with respect to the column-wise partition* $\{\mathcal{c}_1, \dots, \mathcal{c}_p\}$ *if*

$$g(\mathbf{\Phi}) = \sum_{j=1}^{p} g_j(\mathbf{\Phi}(:, \mathcal{c}_j)) \tag{7.9}$$

for some functionals $g_j(\cdot)$ *for* $j = 1, \dots, p$.

We note that the objective functional of the LLQR problem (5.7) is column-wise separable with respect to arbitrary column-wise partition. Here we give some more examples of column-wise separable SLOs.

Example 11. *Consider the LLQR problem* (5.7)*, but now we assume that the covariance matrix of the disturbance is given by* $B_1^\top B_1$ *for some matrix* B_1*, i.e.,* $w \sim \mathcal{N}(0, B_1^\top B_1)$*. Suppose that there exists a permutation matrix* Π *such that the matrix* ΠB_1 *is block diagonal. This happens when the global noise vector* $\mathbf{w}$ *can be partitioned into uncorrelated subsets. The objective in this case is given by*

$$|| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \mathbf{\Phi} B_1 ||_{\mathcal{H}_2}^2 = || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \mathbf{\Phi} \Pi^\top \Pi B_1 ||_{\mathcal{H}_2}^2.$$

Note that $\mathbf{\Phi}\Pi^\top$ *is a column-wise permutation of the optimization variable. We can define a column-wise partition on* $\mathbf{\Phi}\Pi^\top$ *according to the block diagonal structure of the matrix* ΠB_1 *to decompose the objective in a column-wise manner.*

Example 12. *In the LLQR examples, we rely on the inherent separable property of the* $\mathcal{H}_2$ *norm. Specifically, the square of the* $\mathcal{H}_2$ *norm of a FIR transfer matrix* $\mathbf{G} \in \mathcal{F}_T$ *is given by*

$$||\mathbf{G}||_{\mathcal{H}_2}^2 = \sum_i \sum_j ||\mathbf{g}_{ij}||_{\mathcal{H}_2}^2 = \sum_i \sum_j \sum_{t=0}^{T} (g_{ij}[t])^2. \tag{7.10}$$

Motivated by the $\mathcal{H}_2$ *norm, we define the element-wise* ℓ_1 *norm (denoted by* e_1*) of a transfer matrix* $\mathbf{G} \in \mathcal{F}_T$ *as*

$$||\mathbf{G}||_{e_1} = \sum_i \sum_j \sum_{t=0}^{T} |g_{ij}[t]|.$$

The previous example still holds if we change the square of the $\mathcal{H}_2$ *norm to the element-wise* ℓ_1 *norm.*

The column-wise separability of the SLC (7.8c) is defined as follows.

Definition 17. *The system level constraint* $\mathcal{S}$ *in* (7.8c) *is said to be* column-wise separable *with respect to the column-wise partition* $\{c_1, \dots, c_p\}$ *if the following condition is satisfied:*

$$\mathbf{\Phi} \in \mathcal{S} \quad \textit{if and only if} \quad \mathbf{\Phi}(:, c_j) \in \mathcal{S}_j \textit{ for } j = 1, \dots, p$$

for some sets $\mathcal{S}_j$ *for* $j = 1, \dots, p$*. Mathematically, this condition is expressed as*

$$\mathbf{\Phi} \in \mathcal{S} \quad \Longleftrightarrow \quad \bigcap_{j=1}^{p} \mathbf{\Phi}(:, c_j) \in \mathcal{S}_j. \tag{7.11}$$

Equation (7.11) is interpreted as follows. Suppose that the SLC $\mathbf{\Phi} \in \mathcal{S}$ is column-wise separable with respect to the partition $\{c_1, \dots, c_p\}$. To check whether the transfer matrix $\mathbf{\Phi}$ belongs to the set $\mathcal{S}$, we can equivalently check whether the set constraint $\mathbf{\Phi}(:, c_j) \in \mathcal{S}_j$ is satisfied for each j.

Remark 24. *It can be readily shown that the localized constraint $\mathcal{L}$, the FIR constraint $\mathcal{F}_T$, and the strictly proper constraint $\frac{1}{z}\mathcal{RH}_\infty$ are column-wise separable with respect to any column-wise partition. Therefore, the column-wise separability of the SLC $\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X} \cap \frac{1}{z}\mathcal{RH}_\infty$ in* (7.8) *is determined by the column-wise separability of the constraint $\mathcal{X}$. If $\mathcal{S}$ is column-wise separable, then we can express the set constraint $\mathcal{S}_j$ in* (7.11) *as $\mathcal{S}_j = \mathcal{L}(:, c_j) \cap \mathcal{F}_T \cap \mathcal{X}_j \cap \frac{1}{z}\mathcal{RH}_\infty$ for some $\mathcal{X}_j$ for each j.*

We can now formally define column-wise separable SLS problems as follows.

Definition 18. *The state feedback system level synthesis problem* (7.7) *is said to be a* column-wise separable problem *if the SLO* (7.8a) *and the SLC* (7.8c) *are both column-wise separable with respect to some column-wise partition $\{c_1, \dots, c_p\}$.*

Remark 25. *Recall that* (7.8) *is by definition a localized SLS problem. Therefore, if* (7.8) *is column-wise separable and convex, then* (7.8) *is a CLS-SLS problem.*

For a column-wise separable SLS problem (7.8), we can partition (7.8) into p parallel subproblems as

$$\begin{aligned} \underset{\mathbf{\Phi}(:, c_j)}{\text{minimize}} \quad & g_j(\mathbf{\Phi}(:, c_j)) && \text{(7.12a)} \\ \text{subject to} \quad & \mathbf{Z}_{AB}\mathbf{\Phi}(:, c_j) = I(:, c_j) && \text{(7.12b)} \\ & \mathbf{\Phi}(:, c_j) \in \mathcal{L}(:, c_j) \cap \mathcal{F}_T \cap \mathcal{X}_j && \text{(7.12c)} \end{aligned}$$

for $j = 1, \dots, p$.

Then, similar to the LLQR method, we exploit the localized constraint $\mathcal{L}(:, c_j)$ in (7.12c) to reduce the dimension of (7.12) for each j from global scale to local scale. We defer a detailed dimension reduction algorithm for arbitrary localized constraint in Appendix 7.A. Here we just highlight the results. In Appendix 7.A, we show that the optimization subproblem (7.12) can be reduced into

$$\begin{aligned} \underset{\mathbf{\Phi}(s_j,c_j)}{\text{minimize}} \quad & \bar{g}_j(\mathbf{\Phi}(s_j, c_j)) && \text{(7.13a)} \\ \text{subject to} \quad & \mathbf{Z}_{AB}(t_j, s_j)\mathbf{\Phi}(s_j, c_j) = I(t_j, c_j) && \text{(7.13b)} \\ & \mathbf{\Phi}(s_j, c_j) \in \mathcal{L}(s_j, c_j) \cap \mathcal{F}_T \cap \bar{\mathcal{X}}_j, && \text{(7.13c)} \end{aligned}$$

where s_j and t_j are sets of positive integer defined in Appendix 7.A, and $\bar{g}_j$ and $\bar{\mathcal{X}}_j$ the SLO and SLC in the reduced dimension, respectively. Roughly speaking, the set s_j is the collection of optimization variables contained within the localized region specified by $\mathcal{L}(:, c_j)$, and the set t_j is the collection of states that are directly affected by the optimization variables in s_j. The complexity of solving (7.13) is determined by the cardinality of the sets c_j, s_j, and t_j, which are determined by the sparsity of the localized constraint and the system matrices (A, B_2, C_2). For instance, the cardinality of the set s_j is equal to the number of nonzero rows of the localized constraint $\mathcal{L}(:, c_j)$. When the localized constraint and the system matrices are suitably sparse, it is possible to make the size of these sets much smaller than the size of the global network. In this case, the global optimization subproblem (7.12) reduces to a local optimization subproblem (7.13) which depends on the local plant model $\mathbf{Z}_{AB}(t_j, s_j)$ only.

7.2.2 Row-wise Separable Problems

The technique described in the previous subsection can be readily extended to row-wise separable problems. Consider a state estimation SLS problem (the dual of the state feedback SLS) given by

$$\begin{aligned} \underset{\mathbf{\Phi}}{\text{minimize}} \quad & g(\mathbf{\Phi}) && \text{(7.14a)} \\ \text{subject to} \quad & \mathbf{\Phi}\mathbf{Z}_{AC} = I && \text{(7.14b)} \\ & \mathbf{\Phi} \in \mathcal{S}, && \text{(7.14c)} \end{aligned}$$

with $\mathbf{\Phi} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \end{bmatrix}$ and $\mathbf{Z}_{AC} = \begin{bmatrix} zI - A^\top & -C_2^\top \end{bmatrix}^\top$. Let $\{r_1, \dots, r_q\}$ be a partition of the set $\{1, \dots, n_x\}$. The row-wise separability of the SLO and SLC in (7.14) are defined as follows.

Definition 19. *The system level objective* $g(\mathbf{\Phi})$ *in* (7.14a) *is said to be* row-wise separable *with respect to the row-wise partition* $\{r_1, \dots, r_q\}$ *if*

$$g(\mathbf{\Phi}) = \sum_{j=1}^{q} g_j(\mathbf{\Phi}(r_j, :)) \tag{7.15}$$

for some functionals $g_j(\cdot)$ for $j = 1, \ldots, q$.

Definition 20. *The system level constraint $\mathcal{S}$ in* (7.14c) *is said to be* row-wise separable *with respect to the row-wise partition $\{r_1, \ldots, r_p\}$ if*

$$\mathbf{\Phi} \in \mathcal{S} \quad \Longleftrightarrow \quad \bigcap_{j=1}^{q} \mathbf{\Phi}(r_j, :) \in \mathcal{S}_j \tag{7.16}$$

for some sets $\mathcal{S}_j$ for $j = 1, \ldots, q$.

Definition 21. *The state estimation system level synthesis problem* (7.14) *is said to be a* row-wise separable problem *if the SLO* (7.14a) *and the SLC* (7.14c) *are both row-wise separable with respect to some row-wise partition $\{r_1, \ldots, r_p\}$.*

It can be readily seen that the localized distributed Kalman filter (LDKF) problems introduced in Section 5.6 is convex, localized, and row-wise separable. Therefore, LDKF is an example of a CLS-SLS problem.

7.2.3 Summary

In this section, we define the column-wise separability of the SLO and SLC in Definitions 16 and 17, respectively. We then define the column-wise separable SLS problem in Definition 18. We propose an algorithm to solve a convex, localized, and column-wise separable SLS problem in a localized and scalable way. Specifically, we first use the column-wise separability of the problem to decompose the SLS problem into parallel subproblems. We then exploit the localized constraint to reduce the dimension of each subproblem from global scale to local scale. The similar technique is applied to row-wise separable problems as well.

7.3 Convex Localized Separable System Level Synthesis Problems

In this section, we discuss the output feedback localized SLS problem (7.6) with arbitrary B_2 and C_2. Note that (7.6) is neither column-wise nor row-wise separable due to the coupling constraints (7.3a) and (7.3b), so the techniques introduced in the previous section do not apply. We introduce the class of *partially separable* SLS problems in this section, which is a substantial generalization of the column/row-wise separable SLS problems described in Section 7.2 and the LLQG problems introduced in Chapter 6. The intersection of the convex, localized, and partially separable SLS problems form the class of CLS-SLS problems. For a CLS-SLS problem (7.6), we can use ADMM algorithm to decouple (7.6) into two iterative subroutines. Each subroutine can then be solved in a localized and scalable way using the column/row-wise separation method described in the previous section. We show that many constrained optimal control problems belong to the class of CLS-SLS problems. Examples include the localized $\mathcal{H}_2$ optimal control problem with sensor actuator regularization and the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control problem. These problems can be solved for systems with arbitrary large-scale.

7.3.1 Partially Separable Problems

We begin by simplifying the notation of (7.6). We use

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix}$$

to represent the system response we want to optimize for. Denote $\mathbf{Z}_{AB}$ the transfer matrix $\begin{bmatrix} zI - A & -B_2 \end{bmatrix}$ and $\mathbf{Z}_{AC}$ the transfer matrix $\begin{bmatrix} zI - A^\top & -C_2^\top \end{bmatrix}^\top$. Let J_B be the matrix in the right-hand-side of (7.3a) and J_C be the matrix in the right-hand-side of (7.3b). For the ease of presentation, we also include the subspace constraint (7.3c) into the convex set component $\mathcal{X}$. The localized SLS problem (7.6) can be written as

$$\begin{aligned} \underset{\mathbf{\Phi}}{\text{minimize}} \quad & g(\mathbf{\Phi}) && (7.17a) \\ \text{subject to} \quad & \mathbf{Z}_{AB}\mathbf{\Phi} = J_B && (7.17b) \\ & \mathbf{\Phi}\mathbf{Z}_{AC} = J_C && (7.17c) \\ & \mathbf{\Phi} \in \mathcal{S} && (7.17d) \end{aligned}$$

with $\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}$. We assume that (7.17) is a convex problem throughout the rest of the section. The goal of this section is to identify the technical conditions on $g(\cdot)$ and $\mathcal{S}$ in (7.17) so that the convex localized SLS problem (7.17) can be solved in a localized and scalable way.

The primary difference between the output feedback SLS problem (7.17) and its state feedback simplification (7.8) is the additional constraint (7.17c). The column-wise separation technique does not apply to the output feedback problem (7.17) directly because the constraint (7.17c) does not admit a column-wise separation. However, similar to the observation we made in the LLQG problem (cf., Chapter 6), we note that the optimization variable $\mathbf{\Phi}$ in constraint (7.17c) can be solved row at a time. Our strategy is to use distributed optimization techniques such as ADMM to decouple (7.17) into a row-wise separable component and a column-wise separable component. If the SLO (7.17a) and the SLC (7.17d) can be *split* into a row-wise separable component and a column-wise separable component, then we can solve the convex localized SLS problem (7.17) in a localized yet iterative way, using the algorithm similar to the one for LLQG.

From the definition of the system response (7.2), the number of rows and columns of $\mathbf{\Phi}$ in (7.17) are given by $(n_x + n_u)$ and $(n_x + n_y)$, respectively. Let $\{r_1, \ldots, r_q\}$ be a partition of the set $\{1, \ldots, n_x + n_u\}$, and $\{c_1, \ldots, c_p\}$ be a partition of the set $\{1, \ldots, n_x + n_y\}$. We define partially separable SLO as follows.

Definition 22. *The convex system level objective $g(\mathbf{\Phi})$ in* (7.17a) *is said to be* partially separable *with respect to the row-wise partition $\{r_1, \ldots, r_q\}$ and the column-wise partition $\{c_1, \ldots, c_p\}$ if $g(\mathbf{\Phi})$ can be written as the sum of two convex objectives $g^{(r)}(\mathbf{\Phi})$ and $g^{(c)}(\mathbf{\Phi})$, where $g^{(r)}(\mathbf{\Phi})$ is row-wise separable with respect to the row-wise partition $\{r_1, \ldots, r_q\}$ and $g^{(c)}(\mathbf{\Phi})$ is column-wise separable with respect to the column-wise partition $\{c_1, \ldots, c_p\}$. Specifically, we have*

$$\begin{aligned} g(\mathbf{\Phi}) &= g^{(r)}(\mathbf{\Phi}) + g^{(c)}(\mathbf{\Phi}) \\ g^{(r)}(\mathbf{\Phi}) &= \sum_{j=1}^{q} g_j^{(r)}(\mathbf{\Phi}(r_j, :)) \\ g^{(c)}(\mathbf{\Phi}) &= \sum_{j=1}^{p} g_j^{(c)}(\mathbf{\Phi}(:, c_j)) \end{aligned} \tag{7.18}$$

for some convex functionals $g_j^{(r)}(\cdot)$ for $j = 1, \ldots, q$, and $g_j^{(c)}(\cdot)$ for $j = 1, \ldots, p$.

Note that the column/row-wise separable SLO defined in Definition 16 and 19 are special cases of partially separable SLOs. We define partially separable SLC (7.17d) as follows.

Definition 23. *The convex system level constraint $\mathcal{S}$ in* (7.17d) *is said to be* partially separable *with respect to the row-wise partition $\{r_1, \ldots, r_q\}$ and the column-wise partition $\{c_1, \ldots, c_p\}$ if $\mathcal{S}$ can be written as the intersection of two convex sets $\mathcal{S}^{(r)}$ and $\mathcal{S}^{(c)}$, where $\mathcal{S}^{(r)}$ is row-wise separable with respect to the row-wise partition $\{r_1, \ldots, r_q\}$ and $\mathcal{S}^{(c)}$ is column-wise separable with respect to the column-wise partition $\{c_1, \ldots, c_p\}$. Specifically, we have*

$$\begin{aligned} \mathcal{S} &= \mathcal{S}^{(r)} \cap \mathcal{S}^{(c)} \\ \mathbf{\Phi} \in \mathcal{S}^{(r)} &\Longleftrightarrow \bigcap_{j=1}^{q} \mathbf{\Phi}(r_j, :) \in \mathcal{S}_j^{(r)} \\ \mathbf{\Phi} \in \mathcal{S}^{(c)} &\Longleftrightarrow \bigcap_{j=1}^{p} \mathbf{\Phi}(:, c_j) \in \mathcal{S}_j^{(c)} \end{aligned} \tag{7.19}$$

for some convex sets $\mathcal{S}_j^{(r)}$ for $j = 1, \dots, q$ and $\mathcal{S}_j^{(c)}$ for $j = 1, \dots, p$.

Similarly, the column/row-wise separable SLC defined in Definition 17 and 20 are special cases of partially separable SLCs.

Remark 26. *Recall that the SLC set $\mathcal{S}$ in* (7.17) *is given by $\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}$. The localized constraint $\mathcal{L}$ and the FIR constraint $\mathcal{F}_T$ are partially separable with respect to arbitrary row-wise and column-wise partition. Therefore, the partial separability of the SLC $\mathcal{S}$ is determined by the partial separability of the set $\mathcal{X}$. If $\mathcal{X}$ is partially separable, we can express the original SLC $\mathcal{S}$ as an intersection of the sets $\mathcal{S}^{(r)} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}^{(r)}$ and $\mathcal{S}^{(c)} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}^{(c)}$, where $\mathcal{X}^{(r)}$ is a row-wise separable component of $\mathcal{X}$ and $\mathcal{X}^{(c)}$ a column-wise separable component of $\mathcal{X}$. Note that the localized constraint and the FIR constraint are included in both $\mathcal{S}^{(r)}$ and $\mathcal{S}^{(c)}$. As will be shown later, this is the key point to allow the subroutines of the ADMM algorithm to be solved in a scalable way.*

We now formally define partially separable SLS problems as follows.

Definition 24. *The output feedback system level synthesis problem* (7.17) *is said to be a* partially separable problem *if the SLO* (7.17a) *and the SLC* (7.17d) *are both partially separable with respect to a row-wise partition $\{r_1, \dots, r_q\}$ and a column-wise partition $\{c_1, \dots, c_p\}$.*

The definition of the class of CLS-SLS problems is then straightforward.

Definition 25. *A SLS problem* (7.5) *is said to be a* CLS-SLS problem *if it is convex, localized, and partially separable.*

For a CLS-SLS problem (7.17), let $\mathbf{\Psi}$ be a duplicate of the optimization variable $\mathbf{\Phi}$. We define extended-real-value functionals $h^{(r)}(\mathbf{\Phi})$ and $h^{(c)}(\mathbf{\Psi})$ by

$$
\begin{aligned}
h^{(r)}(\mathbf{\Phi}) &= \begin{cases} g^{(r)}(\mathbf{\Phi}) & \text{if (7.17c)}, \mathbf{\Phi} \in \mathcal{S}^{(r)} \\ \infty & \text{otherwise} \end{cases} \\
h^{(c)}(\mathbf{\Psi}) &= \begin{cases} g^{(c)}(\mathbf{\Psi}) & \text{if (7.17b)}, \mathbf{\Psi} \in \mathcal{S}^{(c)} \\ \infty & \text{otherwise.} \end{cases}
\end{aligned} \tag{7.20}
$$

The CLS-SLS problem (7.17) can then be reformulated as

$$
\begin{aligned}
\underset{\{\mathbf{\Phi}, \mathbf{\Psi}\}}{\text{minimize}} \quad & h^{(r)}(\mathbf{\Phi}) + h^{(c)}(\mathbf{\Psi}) \\
\text{subject to} \quad & \mathbf{\Phi} = \mathbf{\Psi}.
\end{aligned} \tag{7.21}
$$

Problem (7.21) can be solved via the standard ADMM approach [5]. Specifically, the ADMM algorithm for (7.21) is given by

$$\mathbf{\Phi}^{k+1} = \underset{\mathbf{\Phi}}{\text{argmin}} \left(h^{(r)}(\mathbf{\Phi}) + \frac{\rho}{2} ||\mathbf{\Phi} - \mathbf{\Psi}^k + \mathbf{\Lambda}^k||^2_{\mathcal{H}_2} \right) \tag{7.22a}$$

$$\mathbf{\Psi}^{k+1} = \underset{\mathbf{\Psi}}{\text{argmin}} \left(h^{(c)}(\mathbf{\Psi}) + \frac{\rho}{2} ||\mathbf{\Psi} - \mathbf{\Phi}^{k+1} - \mathbf{\Lambda}^k||^2_{\mathcal{H}_2} \right) \tag{7.22b}$$

$$\mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k + \mathbf{\Phi}^{k+1} - \mathbf{\Psi}^{k+1}, \tag{7.22c}$$

where the square of the $\mathcal{H}_2$ norm is computed by (7.10). As the FIR constraint $\mathcal{F}_T$ is imposed on $\mathbf{\Phi}$ and $\mathbf{\Psi}$ through the SLC sets $\mathcal{S}^{(r)}$ and $\mathcal{S}^{(c)}$, subroutines (7.22a) and (7.22b) are both finite dimensional optimization problems.

Subroutines (7.22a) and (7.22b) can be further decomposed row-wisely and column-wisely, respectively. For instance, the optimization problem corresponding to subroutine (7.22b) is given by

$$\begin{aligned} \underset{\mathbf{\Psi}}{\text{minimize}} \quad & g^{(c)}(\mathbf{\Psi}) + \frac{\rho}{2} ||\mathbf{\Psi} - \mathbf{\Phi}^{k+1} - \mathbf{\Lambda}^k||^2_{\mathcal{H}_2} && (7.23\text{a}) \\ \text{subject to} \quad & \mathbf{Z}_{AB}\mathbf{\Psi} = J_B && (7.23\text{b}) \\ & \mathbf{\Psi} \in \mathcal{S}^{(c)}. && (7.23\text{c}) \end{aligned}$$

This problem has the same form as (7.8) — the right-hand-side of (7.23b) does not affect the column-wise separability of the problem. The $\mathcal{H}_2$ norm regularizer in (7.23a) is column-wise separable with respect to arbitrary column-wise partition. As the objective $g^{(c)}(\cdot)$ and the constraint $\mathcal{S}^{(c)}$ are column-wise separable with respect to a given column-wise partition, we can use the column-wise separation technique described in the previous section to decompose problem (7.23) into parallel subproblems. Recall that we have $\mathcal{S}^{(c)} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}^{(c)}$. We can then exploit the localized constraint $\mathcal{L}$ and use the technique described in Section 7.2.1 and Appendix 7.A to further reduce the dimension of each subproblem from global scale to local scale. Overall, subroutine (7.22b) can be solved in a localized and scalable manner. Similarly, subroutine (7.22a) can also be solved via the row-wise separation. Equation (7.22c) can be computed element-wisely since it is a matrix addition. Therefore, (7.22a) - (7.22c) is a scalable algorithm to solve the CLS-SLS problem (7.17) by using row-wise and column-wise separation alternatively and iteratively.

We prove that the ADMM algorithm (7.22a) - (7.22c) converges to an optimal solution of (7.17) (or equivalently, (7.21)) under the following assumptions. The details of the proof, as well as the stopping criteria of the algorithm, can be found in Appendix 7.B.

Assumption 4. *Problem* (7.17) *has a feasible solution in the relative interior of the set $\mathcal{S}$.*

Assumption 5. *The functionals $g^{(r)}(\cdot)$ and $g^{(c)}(\cdot)$ are closed, proper, and convex.*

Assumption 6. *The sets $\mathcal{S}^{(r)}$ and $\mathcal{S}^{(c)}$ are closed and convex.*

7.3.2 Examples of CLS-SLS Problems

In this subsection, we give some examples of CLS-SLS problems in optimal control. All these problems can be solved using the ADMM algorithm (7.22) in a localized and scalable way.

Partially Separable Objectives

We begin by considering some examples of partially separable SLOs.

Example 13. *Consider the SLO of the distributed optimal control problem in* (4.10)*, with the norm given by either the square of the $\mathcal{H}_2$ norm or the element-wise ℓ_1 norm defined in Example 12. Suppose that there exists a permutation matrix Π such that the matrix $\begin{bmatrix} B_1^\top & D_{21}^\top \end{bmatrix} \Pi$ is block diagonal. Using a similar argument as in Example 11, we can find a column-wise partition to decompose the SLO in a column-wise manner. Likewise, suppose that there exists a permutation matrix Π such that the matrix $\begin{bmatrix} C_1 & D_{12} \end{bmatrix} \Pi$ is block diagonal. We can find a row-wise partition to decompose the SLO in a row-wise manner. In both cases, the SLO is column/row-wise separable and thus partially separable.*

Example 14. *Consider the weighted actuator norm defined in [40, 41, 70], which is given by*

$$||\mu \begin{bmatrix} \mathbf{M} & \mathbf{L} \end{bmatrix} ||_{\mathcal{U}} = \sum_{i=1}^{n_u} \mu_i || e_i^\top \begin{bmatrix} \mathbf{M} & \mathbf{L} \end{bmatrix} ||_{\mathcal{H}_2}, \tag{7.24}$$

where μ is a diagonal matrix with μ_i being its ith diagonal entry, and e_i is a unit vector with 1 on its ith entry and 0 elsewhere. When μ is an identity matrix, the actuator norm (7.24) *is equivalent to the ℓ_1/ℓ_2 norm, or the group lasso [78] in the statistical learning literature. This norm acts as a regularizer to make the transfer* *matrix* $\begin{bmatrix}\mathbf{M} & \mathbf{L}\end{bmatrix}$ *row-wise sparse, i.e., have many zero rows. Recall from Theorem 2 that the controller achieving the desired system response can be implemented by* (7.4). *If the ith row of the transfer matrix* $\begin{bmatrix}\mathbf{M} & \mathbf{L}\end{bmatrix}$ *is a zero row, then the ith component of the control action* u_i *is always equal to zero. This means that we can remove the control action* u_i *without changing the closed loop response. It is clear that the actuator norm defined in* (7.24) *is row-wise separable with respect to arbitrary row-wise partition. This still holds true when the actuator norm is defined by the* ℓ_1/ℓ_∞ *norm. Similarly, consider the weighted sensor norm given by*

$$|| \begin{bmatrix}\mathbf{N} \\ \mathbf{L}\end{bmatrix} \lambda ||_{\mathcal{Y}} = \sum_{i=1}^{n_y} \lambda_i || \begin{bmatrix}\mathbf{N} \\ \mathbf{L}\end{bmatrix} e_i ||_{\mathcal{H}_2}, \tag{7.25}$$

where λ *is a diagonal matrix with* λ_i *being its ith diagonal entry. The sensor norm in* (7.25) *is a regularizer to make the transfer matrix* $\begin{bmatrix}\mathbf{N}^\top & \mathbf{L}^\top\end{bmatrix}^\top$ *column-wise sparse. Using the controller implementation* (7.4), *the sensor norm can be treated as a regularizer on the measurement* y. *For instance, if the ith column of the transfer matrix* $\begin{bmatrix}\mathbf{N}^\top & \mathbf{L}^\top\end{bmatrix}^\top$ *is identical to zero, then we can ignore the measurement* y_i *without changing the closed loop response. The sensor norm defined in* (7.24) *is column-wise separable with respect to any column-wise partition.*

Example 15. *From Definition 22, it is straightforward to see that the class of partially separable SLOs with the same partition are closed under summation. Therefore, we can combine all the partially separable SLOs described above, and the resulting SLO is still partially separable. For instance, consider the SLO given by*

$$g(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}) = || \begin{bmatrix}C_1 & D_{12}\end{bmatrix} \begin{bmatrix}\mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L}\end{bmatrix} \begin{bmatrix}B_1 \\ D_{21}\end{bmatrix} ||_{\mathcal{H}_2}^2 + ||\mu \begin{bmatrix}\mathbf{M} & \mathbf{L}\end{bmatrix} ||_{\mathcal{U}} + || \begin{bmatrix}\mathbf{N} \\ \mathbf{L}\end{bmatrix} \lambda ||_{\mathcal{Y}}, \tag{7.26}$$

where μ *and* λ *are the relative penalty between the* $\mathcal{H}_2$ *performance, actuator and sensor regularizer, respectively. If there exists a permutation matrix* Π *such that the matrix* $\begin{bmatrix}C_1 & D_{12}\end{bmatrix}\Pi$ *is block diagonal, then the SLO* (7.26) *is partially separable. Specifically, the* $\mathcal{H}_2$ *norm and the actuator regularizer belong to the row-wise separable component, and the sensor regularizer belongs to the column-wise separable component.*

Partially Separable Constraints

We consider some examples of partially separable SLCs beside the localized constraint $\mathcal{L}$ and the FIR constraint $\mathcal{F}_T$ described in Remark 26.

Example 16. *Consider the $\mathcal{L}_1$ norm [11] of a FIR transfer matrix $\mathbf{G} \in \mathcal{F}_T$, which is given by*

$$||\mathbf{G}||_{\mathcal{L}_1} = \max_i \quad \sum_j \sum_{t=0}^{T} |g_{ij}[t]|. \tag{7.27}$$

The $\mathcal{L}_1$ norm is the induced norm of a ℓ_∞ input signal to a ℓ_∞ output signal. Suppose that the ℓ_∞ norm of the disturbance w in (7.2) *is bounded, and we want to bound the ℓ_∞ norm of the state vector x and the control action u. We can impose the constraint*

$$|| \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} ||_{\mathcal{L}_1} \leq \gamma \tag{7.28}$$

in the optimization problem (7.17) *for some γ. The solution of* (7.28) *forms a convex set because it is a sublevel set of a convex function. Therefore,* (7.28) *is a convex SLC. From the definition* (7.27)*, the SLC* (7.28) *is row-wise separable with respect to any row-wise partition.*

Example 17. *From Definition 23, the class of partially separable SLCs with the same partition are closed under intersection. Therefore, we can combine all the partially separable SLCs described above, and the resulting SLC is still partially separable. For instance, the combination of the localized constraint $\mathcal{L}$, the FIR constraint $\mathcal{F}_T$, and the $\mathcal{L}_1$ constraint in* (7.28) *is partially separable. This property is extremely useful because it provides a unified framework to deal with all kinds of partially separable constraints at once.*

Partially Separable Problems

With some examples of partially separable SLOs and SLCs, we now consider two CLS-SLS problems: (i) localized $\mathcal{H}_2$ optimal control problem with sensor actuator regularization, and (ii) localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control. These two problems are used in Section 7.4 as case study examples.

Example 18. *The localized $\mathcal{H}_2$ optimal control with sensor actuator regularization is formulated by*

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\textit{minimize}} \quad & (7.26) && (7.29a) \\ \textit{subject to} \quad & (7.3a) - (7.3c) && (7.29b) \\ & \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T, && (7.29c) \end{aligned}$$

where C is the communication delay SLC. If there exists a permutation matrix Π *such that the matrix* $\begin{bmatrix} C_1 & D_{12} \end{bmatrix} \Pi$ *or the matrix* $\begin{bmatrix} B_1^\top & D_{21}^\top \end{bmatrix} \Pi$ *is block diagonal, then* (7.29) *is partially separable.*

Remark 27. *When the penalty of the sensor and actuator norms are zero, problem* (7.29) *reduces to a LLQG problem (cf., Chapter 6). If the system is state feedback and we only have actuator regularizer, then* (7.29) *reduces to the LLQR problem with actuator regularization (cf., Section 5.3).*

Problem (7.29) can be used to co-design a localized optimal controller and its sensing and actuation interface by choosing the relative weight among the $\mathcal{H}_2$ performance, actuator and sensor norm. We emphasize the fact that (7.29) is a CLS-SLS problem, and thus can be solved by the ADMM algorithm (7.22) with $O(1)$ parallel computational complexity. In other words, we can co-design a localized optimal controller and its sensing and actuation interface in a localized and scalable way. The LLQR with actuator regularization problem described in Section 5.3 is a special case of (7.29), and thus can be solved in a localized way using the ADMM algorithm (7.22) as well.

The weights μ_i and λ_i in the regularizers (7.24) and (7.25) can be properly chosen to further enhance row/column-wise sparsity. For instance, we can use the reweighted ℓ_1 algorithm proposed in [6] to iteratively set the weights and solve (7.29) multiple times. Let $\mu_i^{(0)} = \mu_0$ for $i = 1, \dots, n_u$ and $\lambda_i^{(0)} = \lambda_0$ for $i = 1, \dots, n_y$. Let $(\mathbf{R}^{(k)}, \mathbf{M}^{(k)}, \mathbf{N}^{(k)}, \mathbf{L}^{(k)})$ be the optimal solution of (7.29) when the weights are given by $\{\mu_i^{(k)}\}_{i=1}^{n_u}$ and $\{\lambda_i^{(k)}\}_{i=1}^{n_y}$. We update the weights at iteration $(k+1)$ by

$$\begin{aligned} \mu_i^{(k+1)} &= (\,||e_i^\top \begin{bmatrix} \mathbf{M}^{(k)} & \mathbf{L}^{(k)} \end{bmatrix} ||_{\mathcal{H}_2} + \epsilon)^{-1} \\ \lambda_i^{(k+1)} &= (\,|| \begin{bmatrix} \mathbf{N}^{(k)} \\ \mathbf{L}^{(k)} \end{bmatrix} e_i ||_{\mathcal{H}_2} + \epsilon)^{-1} \end{aligned} \qquad (7.30a)$$

for some small ϵ. It is shown in [6] that this reweighted scheme usually results in sparser solution.

Next, we consider the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control given as follows:

$$\underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad || \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} ||_{\mathcal{H}_2}^2 \qquad (7.31a)$$

$$\text{subject to} \quad (7.3a) - (7.3c), (7.28), (7.29c). \qquad (7.31b)$$

The localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control problem can be used to design the tradeoff between average-case performance and worst-case performance. Specifically, the $\mathcal{H}_2$ objective in (7.31a) is the expected value of the energy of the state and control for AWGN disturbances, which measures the average-case performance of the closed loop response. The $\mathcal{L}_1$ constraint in (7.28) is the ℓ_∞-ℓ_∞ induced norm, which measures the worst-case state and control deviation for ℓ_∞ bounded disturbances.

7.3.3 Analytic Solution and Acceleration

Suppose that the Assumptions 4 - 6 in Section 7.3.1 hold. The ADMM algorithm presented in (7.22) is a special case of the proximal algorithm [5, 9, 48]. For certain type of objective functionals $h^{(r)}(\cdot)$ and $h^{(c)}(\cdot)$, the proximal operators can be evaluated analytically (see Ch. 6 of [48]). In this situation, we only need to evaluate the proximal operators *once*, and iterate (7.22a) and (7.22b) in closed form. This improves the overall computation time significantly. We explain how to express the solutions of (7.22a) and (7.22b) using proximal operators in detail in Appendix 7.C. Here we list a few examples that the proximal operators can be evaluated analytically.

Example 19. *Consider the LLQG problem in Chapter 6. When the global optimization problem is decomposed into parallel subproblems, each subproblem is a convex quadratic program restricted on an affine set. In this case, the proximal operator is an affine function [5, 48, 68]. We only need to calculate this affine function once. The iteration in* (7.22a) - (7.22c) *can then be carried out using multiple matrix multiplications in the reduced dimension, which significantly improves the overall computation time.*

Example 20. *Consider the LLQR problem with actuator regularization in Section 5.3, which is the state feedback version of* (7.29). *The column-wise separable part is identical to the LLQG example, so the update* (7.22b) *can be computed using matrix multiplicatiosn. Suppose that we use* (7.24) *as the actuator norm. We showed in [70] that the row-wise separable part can be simplified into multiple unconstrained optimization problems, with proximal operators given by vectorial soft-thresholding [48]. This offers an efficient way to update* (7.22a).

Beside ADMM, there exists other distributed algorithms that can be used to solve the optimization problem (7.21) in a localized and scalable way. For instance, if either $g^{(r)}(\cdot)$ or $g^{(c)}(\cdot)$ is strongly convex, we can use the alternating minimization algorithm (AMA) [60] to simplify the ADMM algorithm. Some other alternatives include the over-relax ADMM [5] and the accelerated version of ADMM and AMA proposed in [22].

7.4 Simulation Results

In this section, we continue with the power system example described in Section 6.4, and demonstrate the localized $\mathcal{H}_2$ optimal control with sensor actuator regularization (7.29) and the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control (7.31).

We first solve the localized $\mathcal{H}_2$ optimal control with sensor actuator regularization to co-design the LLQG controller and the locations of sensors and actuators in the power network. Then, we solve the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control problem to exploit the tradeoff between average-case performance and worst-case performance of the closed loop response.

7.4.1 Localized $\mathcal{H}_2$ with Sensor Actuator Regularization

We first consider the 10×10 mesh example shown in Figure 6.2a. In Section 6.4, we assume that each subsystem in the power network has a phase measurement unit (PMU), a frequency sensor, and a controllable load. In practice, the installation of these sensors and actuators are expensive, and we would like to deliberately trade off the closed loop performance and the number of sensors and actuators being used. A challenging problem is to determine the optimal locations of these sensors and actuators due to its combinatorial complexity. In this subsection, we apply the regularization for design (RFD) [40] framework to jointly design the localized optimal controller and the optimal locations of sensors and actuators in the power network. This is achieved by solving the localized $\mathcal{H}_2$ optimal control with sensor actuator regularization in (7.29).

In order to allow more flexibility on sensor actuator placement, we increase the localized region of each process and sensor noise from its two-hop neighbors to its four-hop neighbors. This implies that each subsystem j needs to exchange the information up to its four-hop neighbors, and use the restricted plant model within its four-hop neighbors to synthesize the LLQG controller. The length of the FIR constraint $\mathcal{F}_T$ is increased to $T = 30$. The $\mathcal{H}_2$ cost achieved by the LLQG controller is given by 13.3210, which is 0.03% degradation compared to the one achieved by an idealized centralized $\mathcal{H}_2$ optimal controller. We assume that the relative price between each frequency sensor, PMU, and controllable load are 1, 100, and 300, respectively. This is to model the fact that actuators are typically more expensive than sensors, and PMU are typically more expensive than frequency sensor. The price for the same type of sensors and actuators at different locations are the same.

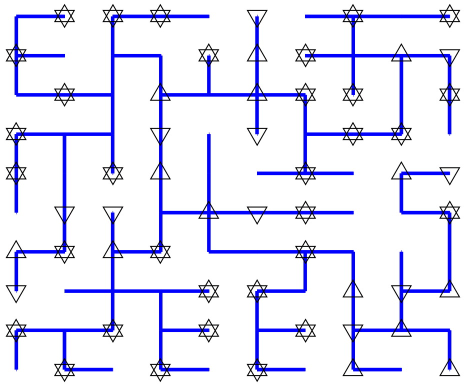


Figure 7.1: The upward-pointing triangles represent the subsystems in which the PMU is removed. The downward-pointing triangles represent the subsystems in which the controllable load (actuator) is removed.

We run the reweighted ℓ_1 algorithm for (7.29) 7 times, and remove the sensors and actuators for those with their sensor or actuator norm smaller than 0.02. Originally, there are 100 controllable loads, 100 PMUs, and 100 frequency sensors in the power network. After solving (7.29), we successfully remove 43 controllable loads and 46 PMUs in the network (no frequency sensors are removed due to the chosen relative pricing). The locations of the removed sensors and actuators are shown in Figure 7.1. We argue that this sensing and actuation architecture is very sparse. In particular, we only use 57 controllable loads to control process noise from 200 states and sensor noise from 154 states, while ensuring that the system response for *all* the disturbances are localized FIR.

For the system with reduced number of sensors and actuators, the $\mathcal{H}_2$ cost achieved by the LLQG controller is given by 17.8620. As a comparison, the cost achieved by a proper centralized $\mathcal{H}_2$ optimal controller is 16.2280, and the cost achieved by a strictly proper centralized $\mathcal{H}_2$ optimal controller is 18.4707. Note that when the sensing and actuation interface become sparser, the performance gap between the centralized and the localized controller becomes larger. This is an inevitable tradeoff because the system tends to spread the control effort to a larger region when the the sensing and actuation interface get sparse. Nevertheless, we note that the performance degradation is only 10% compared to the proper centralized $\mathcal{H}_2$ optimal scheme. In addition, our LLQG controller, despite having the additional localized region constraint, FIR constraint, and communication delay constraint, still outperforms the strictly proper centralized $\mathcal{H}_2$ optimal controller in 3.4%.

7.4.2 Localized Mixed $\mathcal{H}_2/\mathcal{L}_1$ Optimal Control

We solve the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control problem in (7.31) on the 10×10 mesh example shown in Figure 6.2a. We iteratively reduce the $\mathcal{L}_1$ sublevel set of (7.31) to exploit the tradeoff between average-case performance and worst-case performance. We plot the normalized $\mathcal{H}_2$ norm and the normalized $\mathcal{L}_1$ norm in Figure 7.2. The left-top point in Figure 7.2 is the localized $\mathcal{H}_2$ solution. When we start reducing the $\mathcal{L}_1$ sublevel set, the $\mathcal{H}_2$ norm of the closed loop response gradually increases. Note that there usually exists a sweet spot on the tradeoff curve such that the average-case performance ($\mathcal{H}_2$ norm) and the worst-case performance ($\mathcal{L}_1$ norm) is well balanced.

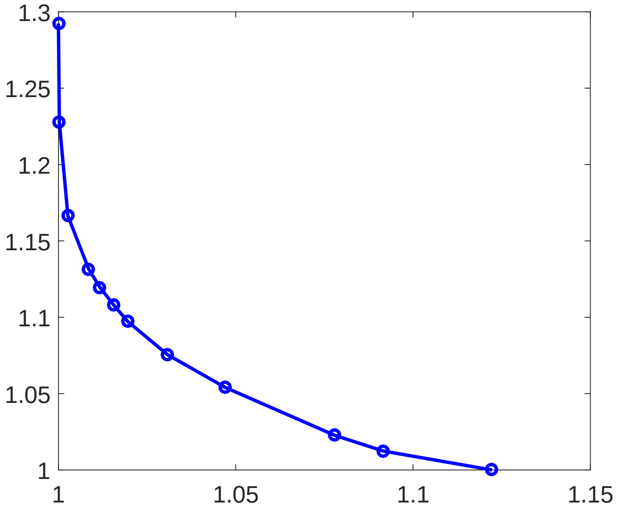


Figure 7.2: The vertical axis represents the normalized $\mathcal{L}_1$ norm of the closed loop, and the horizontal axis represents the normalized $\mathcal{H}_2$ norm of the closed loop.

APPENDIX

7.A Dimension Reduction Algorithm

Consider problem (7.12) with a specific j. Recall from (7.2) that the number of rows of the transfer matrix $\mathbf{\Phi}(:, c_j)$ is given by $(n_x + n_u)$. Let $\bar{s}_j$ be the largest subset of $\{1, \dots, n_x + n_u\}$ such that the localized constraint $\mathcal{L}(\bar{s}_j, c_j)$ in (7.12c) is exactly a zero matrix. When the localized constraint is imposed, we must have $\mathbf{\Phi}(\bar{s}_j, c_j) = 0$. This part can be eliminated in the objective (7.12a) and the constraints (7.12b) - (7.12c). Let $s_j = \{1, \dots, n_x + n_u\} - \bar{s}_j$ be the complement set of $\bar{s}_j$. When the localized constraint is imposed, problem (7.12) can be simplified into

$$\begin{aligned} \underset{\mathbf{\Phi}(s_j, c_j)}{\text{minimize}} \quad & \bar{g}_j(\mathbf{\Phi}(s_j, c_j)) && (7.32a) \\ \text{subject to} \quad & \mathbf{Z}_{AB}(:, s_j)\mathbf{\Phi}(s_j, c_j) = I(:, c_j) && (7.32b) \\ & \mathbf{\Phi}(s_j, c_j) \in \mathcal{L}(s_j, c_j) \cap \mathcal{F}_T \cap \bar{\mathcal{X}}_j, && (7.32c) \end{aligned}$$

where $\bar{g}_j(\cdot)$ and $\bar{\mathcal{X}}_j$ are the restriction of $g_j(\cdot)$ and $\mathcal{X}_j$ on the constraint $\mathbf{\Phi}(\bar{s}_j, c_j) = 0$, respectively. From the definition of the set s_j, it is straightforward to see that (7.32) is equivalent to (7.12).

Next, when the system matrices (A, B_2) are fairly sparse, the transfer matrix $\mathbf{Z}_{AB}$ is also sparse. Recall that the number of rows of $\mathbf{Z}_{AB}$ is given by n_x. Let $\bar{t}_j$ be the largest subset of $\{1, \dots, n_x\}$ such that the augmented matrix $\begin{bmatrix} \mathbf{Z}_{AB}(\bar{t}_j, s_j) & I(\bar{t}_j, c_j) \end{bmatrix}$ is a zero matrix. Let $t_j = \{1, \dots, n_x\} - \bar{t}_j$ be the complement set of $\bar{t}_j$. We can further reduce (7.32) into (7.13). From the definition of the set t_j, we know that (7.13) is equivalent to (7.32) and therefore equivalent to (7.12).

Remark 28. *The dimension reduction algorithm proposed here is a generalization of the algorithms proposed in Chapters 5 - 6. Specifically, the algorithms described in Chapters 5 - 6 only work for the d-localized constraint. The dimension reduction algorithm proposed in this section can handle arbitrary localized constraint, and for system matrices (A, B_2) with arbitrary sparsity pattern.*

7.B Convergence of ADMM Algorithm (7.22)

Assumptions 4 - 6 imply feasibility and strong duality of (7.17) (or its equivalent formulation (7.21)). From Assumptions 5 and 6, we know that the extended-real-value functionals $h^{(r)}(\cdot)$ and $h^{(c)}(\cdot)$ defined in (7.20) are closed, proper, and convex. Under these assumptions, problem (7.21) satisfies the convergence conditions in [5].

From [5], we have objective convergence, dual variable convergence ($\mathbf{\Lambda}^k \rightarrow \mathbf{\Lambda}^*$ as $k \rightarrow \infty$), and residual convergence ($\mathbf{\Phi}^k - \mathbf{\Psi}^k \rightarrow 0$ as $k \rightarrow \infty$). As long as the problem (7.21) approaches feasibility and the optimal value of (7.21) converges to the global optimal, we can use a feasible solution of (7.21) to construct a controller that achieves the desired optimal system response using (7.4). Note that the optimization problem (7.21) may not have a unique optimal point, so the optimization variables $\mathbf{\Phi}^k$ and $\mathbf{\Psi}^k$ do not necessary converge. If we further assume that the SLO $g(\cdot)$ is strongly convex with respect to $\mathbf{\Phi}$, then problem (7.21) has a unique optimal solution $\mathbf{\Phi}^*$. In this case, objective convergence implies primal variable convergence, so we have $\mathbf{\Phi}^k \rightarrow \mathbf{\Phi}^*$ and $\mathbf{\Psi}^k \rightarrow \mathbf{\Phi}^*$ as $k \rightarrow \infty$.

The stopping criteria is designed from [5], in which we use $||\mathbf{\Phi}^k - \mathbf{\Psi}^k||_{\mathcal{H}_2}$ as primal infeasibility and $||\mathbf{\Psi}^k - \mathbf{\Psi}^{k-1}||_{\mathcal{H}_2}$ as dual infeasibility. The square of these two functions can be calculated in a localized and distributed way. The algorithm (7.22a) - (7.22c) terminates when $||\mathbf{\Phi}^k - \mathbf{\Psi}^k||_{\mathcal{H}_2} < \epsilon^{pri}$ and $||\mathbf{\Psi}^k - \mathbf{\Psi}^{k-1}||_{\mathcal{H}_2} < \epsilon^{dual}$ are satisfied for some feasibility tolerances ϵ^{pri} and ϵ^{dual}.

In practice, we may not know the feasibility of (7.17) in advanced. In other words, we do not know whether Assumption 4 holds. Consider the case that the ADMM subroutines (7.22a) - (7.22c) are solvable, but problem (7.21) is infeasible. In this situation, the stopping criteria on primal infeasibility $||\mathbf{\Phi}^k - \mathbf{\Psi}^k||_{\mathcal{H}_2} < \epsilon^{pri}$ may not be satisfied. To avoid infinite number of iterations, we set a limit on the number of iterations in the ADMM algorithm. The convergence result of the ADMM algorithm should be regarded as a scalable algorithm to check the feasibility of the CLS-SLS problem (7.17). If the ADMM algorithm does not converge, then we know that the system (A, B_2, C_2) is not feasible with respect to the given SLC. The design of a feasible SLC for state feedback system is described in details in Section 5.3, in which we present a method that allows for the joint design of an actuator architecture and the corresponding feasible spatiotemporal SLC. This method can be extended to output feedback system by solving the localized $\mathcal{H}_2$ optimal control problem with sensor actuator regularization (7.29).

7.C Express ADMM Solution using Proximal Operators

Here we explain how to express the solutions of (7.22a) and (7.22b) using proximal operators. We focus our discussion on (7.22b), or its equivalent formulation in (7.23), while the same argument holds for (7.22a) as well. Recall that the set $\mathcal{S}^{(c)}$ in (7.23c) is given by $\mathcal{S}^{(c)} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}^{(c)}$. We use the column-wise partition and the dimension reduction techniques described in Section 7.2 and Appendix 7.A (the procedure that simplifies (7.8) into (7.13)) to simplify (7.23) into

$$\begin{aligned} \underset{\mathbf{\Psi}(s_j,c_j)}{\text{minimize}} \quad & g_j^{(c)}(\mathbf{\Psi}(s_j,c_j)) + \frac{\rho}{2}||\mathbf{\Psi}(s_j,c_j) - \mathbf{\Phi}^{k+1}(s_j,c_j) - \mathbf{\Lambda}^k(s_j,c_j)||_{\mathcal{H}_2}^2 && (7.33\text{a}) \\ \text{subject to} \quad & \mathbf{Z}_{AB}(t_j,s_j)\mathbf{\Psi}(s_j,c_j) = J_B(t_j,c_j) && (7.33\text{b}) \\ & \mathbf{\Psi}(s_j,c_j) \in \mathcal{L}(s_j,c_j) \cap \mathcal{F}_T \cap \mathcal{X}_j^{(c)} && (7.33\text{c}) \end{aligned}$$

for $j = 1, \ldots p$. In (7.33), the transfer matrices $\mathbf{\Psi}(s_j,c_j)$, $\mathbf{\Phi}^{k+1}(s_j,c_j)$, and $\mathbf{\Lambda}^k(s_j,c_j)$ are all FIRs with horizon T. We express the optimization variables of problem (7.33) using a column vector defined by

$$\Psi_{v(j)} = \text{vec}(\begin{bmatrix} \Psi(s_j,c_j)[0] & \cdots & \Psi(s_j,c_j)[T] \end{bmatrix}),$$

where $\Psi_{v(j)}$ is the vectorization of all the spectral components of $\mathbf{\Psi}(s_j,c_j)$. Similarly, we define $\Phi_{v(j)}^{k+1}$ and $\Lambda_{v(j)}^k$ the vectorization of all the spectral components of $\mathbf{\Phi}^{k+1}(s_j,c_j)$ and $\mathbf{\Lambda}^k(s_j,c_j)$, respectively. The optimization problem (7.33) can then be written in the form of

$$\begin{aligned} \underset{\Psi_{v(j)}}{\text{minimize}} \quad & g_{v(j)}(\Psi_{v(j)}) + \frac{\rho}{2}||\Psi_{v(j)} - \Phi_{v(j)}^{k+1} - \Lambda_{v(j)}^k||_2^2 && (7.34\text{a}) \\ \text{subject to} \quad & \Psi_{v(j)} \in \mathcal{S}_{v(j)}, && (7.34\text{b}) \end{aligned}$$

where $g_{v(j)}(\cdot)$ is the vectorization form of $g_j^{(c)}(\cdot)$, and $\mathcal{S}_{v(j)}$ is the set constraint imposed by (7.33b) - (7.33c). Define the indicator function by

$$\mathcal{I}_{\mathcal{D}}(x) = \begin{cases} 0 & x \in \mathcal{D} \\ \infty & x \notin \mathcal{D}. \end{cases}$$

We rewrite (7.34) as an unconstrained problem given by

$$\underset{\Psi_{v(j)}}{\text{minimize}} \quad \mathcal{I}_{\mathcal{S}_{v(j)}}(\Psi_{v(j)}) + g_{v(j)}(\Psi_{v(j)}) + \frac{\rho}{2}||\Psi_{v(j)} - \Phi_{v(j)}^{k+1} - \Lambda_{v(j)}^k||_2^2. \tag{7.35}$$

The solution of (7.35) can be expressed using the proximal operator as

$$\Psi_{v(j)}^{k+1} = \mathbf{prox}_{\mathcal{I}_{\mathcal{S}_{v(j)}} + \frac{1}{\rho}g_{v(j)}}(\,\Phi_{v(j)}^{k+1} + \Lambda_{v(j)}^k\,). \tag{7.36}$$

Equation (7.36) is a solution of (7.33). Therefore, the ADMM update (7.22b) can be carried out by the proximal operators (7.36) for $j = 1, \ldots p$, all in the reduced dimension.

Chapter 8

CONCLUSIONS AND FUTURE WORKS

8.1 Summary

In this dissertation, we defined and analyzed the system level approach to controller synthesis, and showed its applications to design controllers for large-scale distributed systems. The system level approach consists of three main elements: System Level Parameterizations (SLPs), System Level Constraints (SLCs), and System Level Synthesis (SLS) problems. We showed that all achievable and stable system responses can be characterized via the SLPs given in Theorems 1 and 2. We further showed that these system responses could be used to parameterize internally stabilizing controllers that achieved them, and proposed a novel controller implementation (3.16) with structure shown in Figure 3.2. We then argued that this novel controller implementation had the important benefit of allowing for SLCs to be naturally imposed on it, and showed in Chapter 4 that using this controller structure and SLCs, we can characterize the broadest known class of constrained internally stabilizing controllers that admit a convex representation. In addition, we combined SLPs and SLCs to formulate the SLS problem in (4.1), and showed that it recovered as a special case many well studied constrained optimal controller synthesis problems from the literature.

In Chapters 5 - 7, we introduced the class of convex localized separable SLS (CLS-SLS) problems. When the localized SLC is properly specified, we showed that the CLS-SLS problems can be solved in a localized and scalable way using distributed optimization, with $O(1)$ parallel computational complexity compared to the size of the global network. We gave several examples of CLS-SLS problems, including the localized linear quadratic regulator (LLQR) in Chapter 5, localized linear quadratic Gaussian (LLQG) in Chapter 6, the localized $\mathcal{H}_2$ optimal control with sensor actuator regularization and the localized mixed $\mathcal{H}_2/\mathcal{L}_1$ optimal control problem in Chapter 7. We demonstrated the LLQR method on a randomized heterogeneous power network example with 51200 states. We showed that the LLQR controller for this large-scale example can be synthesized in 23 minutes using a personal computer, while the theoretical computation time for the traditional centralized LQR using the same computer is 200 days, and the distributed LQR is intractable. As illustrated in

Tables 5.2 and 6.1, the CLS-SLS framework provides superior scalability over the centralized and distributed approach, which is extremely favorable for large-scale applications.

8.2 Potential Applications and Future Works

Here we present a few potential applications as well as the future works of the SLS framework.

8.2.1 Smart Grid

The trend of future power grid is that there will be more renewable energy resources such as wind farm or solar panel integrated into the power network. While those distributed energy resources (DERs) offer many potential opportunities to the network, they also introduce large, rapid, and random fluctuations in power supply, voltage, and frequency into the grid. The regulation and control of the power system dynamics in the presence of uncertainty and disturbance therefore become an important issue in the era of smart grid [12, 79, 80].

Consider the frequency regulation problem as an example. When power demand or supply fluctuates, the power frequency on a bus may deviate from its nominal value. This deviation will then change the power flow nearby, resulting a disturbance propagating in the power network. This problem is minor in the current power grid, as the frequency dynamics is stabilized by the huge inertia of rotating machines. However, as more and more generators and loads are driven by power electronics in the future power grid, the loss of inertia (damping) in the power network raise significant challenges to the frequency regulation problem. One potential solution is introducing active fast timescale controller on millions of active endpoints (photovaltaics, wind turbines, inverters, storage devices, ...etc) in the power network. The frequency regulation problem then becomes a large-scale optimal control problem, which can be solved using the CLS-SLS framework.

8.2.2 Transportation Systems

The study of Automated Highway Systems (AHS) that utilize vehicle-to-vehicle or vehicle-to-infrastructure communication has gained interest since the 1990s. This infrastructural change opens the possibility to design a highly efficient traffic control scheme. The CLS-SLS framework offers a potential to use vehicle-level control action to alter the macroscopic behavior of traffic flow.

We demonstrate a preliminary result of applying LLQR on shock wave localization in a transportation network in [56]. Shock waves are generated in transportation systems due to a discontinuity in the density profile caused by large disturbances. The creation and propagation of shock waves in traffic leads to decreased throughput and several million gallons of wasted fuel [58]. In [56], we use the discretized cell transmission model [10] to describe the traffic flow. This model can further be simplified into a hybrid state space model with two modes. We then extend the LLQR theory to this class of hybrid system. Preliminary results show some positive effects of our control scheme on shock wave attenuation. In the future, we will try to extend the result using a more realistic and less simplified model.

8.2.3 Software-Defined Networking

Software-defined networking (SDN) is a new networking paradigm that separates the network control logic (control plane) from the forwarding hardware (data plane). This opens the possibility of much more programmable networking, and the network management problem can be viewed as a control problem. While the control plane of SDN is operated to achieve global and centralize objectives, the implementation of the control algorithm is physically distributed due to scalability, fast response, and geographical constraints. In [44], the authors study the performance tradeoffs among myopic (completely decentralized), coordinated (distributed), and centralized SDN architectures on the admission control problems. It is worth it to see if the CLS-SLS framework can be applied to the same control problem but with larger scale.

8.2.4 Layered Control Architecture

It should be noted that the SLS problems and the traditional optimal control problems focus primarily on the design of the feedback control laws for the purpose of disturbance suppression and reference tracking. The overall design of a cyber-physical system often involve more high-level objectives, including planning, scheduling, resource allocation, and network utility maximization. An interesting future research direction is to integrate the SLS optimization problem into the layering as optimization (LAO) framework [8, 43] to design layered control architecture for cyber-physical systems.

BIBLIOGRAPHY

[1] Brian DO Anderson and John B Moore. *Optimal filtering*. Courier Corporation, 2012.

[2] Bassam Bamieh, Fernando Paganini, and Munther A. Dahleh. “Distributed Control of Spatially Invariant Systems”. In: *Automatic Control, IEEE Transactions on* 47.7 (2002), pp. 1091–1107.

[3] Bassam Bamieh and Petros G Voulgaris. “A convex characterization of distributed control problems in spatially invariant systems with communication constraints”. In: *Systems & Control Letters* 54.6 (2005), pp. 575–583.

[4] Vincent Blondel and John N Tsitsiklis. “NP-hardness of some linear control design problems”. In: *SIAM Journal on Control and Optimization* 35.6 (1997), pp. 2118–2127.

[5] Stephen P. Boyd et al. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. In: *Foundations and Trends in Machine Learning* 3.1 (2011), pp. 1–122.

[6] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd. “Enhancing Sparsity by Reweighted ℓ_1 Minimization”. In: *Journal of Fourier Analysis and Applications* 14.5-6 (2008), pp. 877–905. ISSN: 1069-5869. DOI: `10.1007/s00041-008-9045-x`.

[7] Federico S Cattivelli and Ali H Sayed. “Diffusion strategies for distributed Kalman filtering and smoothing”. In: *Automatic Control, IEEE Transactions on* 55.9 (2010), pp. 2069–2084.

[8] Mung Chiang et al. “Layering as optimization decomposition: A mathematical theory of network architectures”. In: *Proceedings of the IEEE* 95.1 (2007), pp. 255–312.

[9] Patrick L. Combettes and Jean-Christophe Pesquet. “Proximal Splitting Methods in Signal Processing”. In: *arXiv preprint* arXiv:0912.3522 (2010). URL: `http://arxiv.org/abs/0912.3522v4.pdf`.

[10] Carlos F Daganzo. “The cell transmission model, part II: network traffic”. In: *Transportation Research Part B: Methodological* 29.2 (1995), pp. 79–93.

[11] M.A. Dahleh and Jr. Pearson J. “l^1-optimal feedback controllers for MIMO discrete-time systems”. In: *Automatic Control, IEEE Transactions on* 32.4 (Apr. 1987), pp. 314–322.

[12] Florian Dörfler et al. “Sparsity-promoting optimal wide-area control of power networks”. In: *IEEE Transactions on Power Systems* 29.5 (2014), pp. 2281–2291.

[13] J. C. Doyle et al. “State-space solutions to standard H_2 and H_∞ control problems”. In: *IEEE Transactions on Automatic Control* 34.8 (Aug. 1989), pp. 831–847. ISSN: 0018-9286. DOI: 10.1109/9.29425.

[14] Geir E. Dullerud and Raffaello D’Andrea. “Distributed Control of Heterogeneous Systems”. In: *Automatic Control, IEEE Transactions on* 49.12 (2004), pp. 2113–2128.

[15] Geir E. Dullerud and Fernando Paganini. *A Course In Robust Control Theory: A Convex Approach*. Springer-Verlag, 2000.

[16] Krishnamurthy Dvijotham, Emanuel Todorov, and Maryam Fazel. “Convex Structured Controller Design in Finite Horizon”. In: *Control of Network Systems, IEEE Transactions on* 2.1 (2015), pp. 1–10.

[17] Makan Fardad, Fu Lin, and Mihailo R. Jovanovic. “Sparsity-Promoting Optimal Control for a Class of Distributed Systems”. In: *2011 IEEE American Control Conference (ACC)*. June 2011.

[18] Brian F Farrell and Petros J Ioannou. “State estimation using a reduced-order Kalman filter”. In: *Journal of the Atmospheric Sciences* 58.23 (2001), pp. 3666–3680.

[19] Ghazal Fazelnia, Ramtin Madani, and Javad Lavaei. “Convex Relaxation for Optimal Distributed Control Problem”. In: *2014 53rd IEEE Conference on Decision and Control (CDC)*. 2014.

[20] Ichiro Fukumori and Paola Malanotte-Rizzoli. “An approximate Kalman filter for ocean data assimilation: An example with an idealized Gulf Stream model”. In: *Journal of Geophysical Research: Oceans (1978–2012)* 100.C4 (1995), pp. 6777–6793.

[21] Carlos E Garcia and Manfred Morari. “Internal model control. A unifying review and some new results”. In: *Industrial & Engineering Chemistry Process Design and Development* 21.2 (1982), pp. 308–323.

[22] Tom Goldstein et al. “Fast Alternating Direction Optimization Methods”. In: *SIAM Journal on Imaging Sciences* 7.3 (2014), pp. 1588–1623. DOI: 10.1137/120896219.

[23] Yu-Chi Ho and K.-C. Chu. “Team decision theory and information structures in optimal control problems–Part I”. In: *Automatic Control, IEEE Transactions on* 17.1 (1972), pp. 15–22.

[24] Yih-Fang Huang et al. “State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid”. In: *Signal Processing Magazine, IEEE* 29.5 (2012), pp. 33–43.

[25] Rudolf Emil Kalman, Peter L Falb, and Michael A Arbib. *Topics in mathematical system theory*. Vol. 33. McGraw-Hill New York, 1969.

[26] Usman A Khan and José MF Moura. “Distributing the Kalman filter for large-scale systems”. In: *Signal Processing, IEEE Transactions on* 56.10 (2008), pp. 4919–4935.

[27] Andrew Lamperski and John C. Doyle. “Output Feedback $\mathcal{H}_2$ Model Matching for Decentralized Systems with Delays”. In: *2013 IEEE American Control Conference (ACC)*. June 2013.

[28] Andrew Lamperski and John C. Doyle. “The $\mathcal{H}_2$ Control Problem for Quadratically Invariant Systems with Delays”. In: *Automatic Control, IEEE Transactions on* 60.7 (2015), pp. 1945–1950.

[29] Andrew Lamperski and Laurent Lessard. “Optimal Decentralized State-Feedback Control with Sparsity and Delays”. In: *Automatica* 58 (2015), pp. 143–151.

[30] L. Lessard, M. Kristalny, and A. Rantzer. “On structured realizability and stabilizability of linear systems”. In: *American Control Conference (ACC), 2013*. June 2013, pp. 5784–5790.

[31] L. Lessard and S. Lall. “Quadratic invariance is necessary and sufficient for convexity”. In: *Proceedings of the 2011 American Control Conference*. June 2011, pp. 5360–5362.

[32] Laurent Lessard. “State-space solution to a minimum-entropy $\mathcal{H}_\infty$-optimal control problem with a nested information constraint”. In: *2014 53rd IEEE Conference on Decision and Control (CDC)*. 2014. URL: `http://arxiv.org/pdf/1403.5020v2.pdf`.

[33] Laurent Lessard and Sanjay Lall. “Convexity of Decentralized Controller Synthesis”. In: *IEEE Transactions on Automatic Control, To appear* (2016). URL: `http://arxiv.org/pdf/1305.5859v2.pdf`.

[34] Laurent Lessard and Sanjay Lall. “Optimal Controller Synthesis for the Decentralized Two-Player Problem with Output Feedback”. In: *2012 IEEE American Control Conference (ACC)*. June 2012.

[35] Fu Lin, Makan Fardad, and Mihailo R. Jovanovic. “Design of Optimal Sparse Feedback Gains via the Alternating Direction Method of Multipliers”. In: *Automatic Control, IEEE Transactions on* 58.9 (2013), pp. 2426–2431.

[36] CJ Long et al. “Large scale Kalman filtering solutions to the electrophysiological source localization problem-a MEG case study”. In: *Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE*. IEEE. 2006, pp. 4532–4535.

[37] A. Mahajan et al. “Information structures in optimal decentralized control”. In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. 2012, pp. 1291–1306. DOI: `10.1109/CDC.2012.6425819`.

[38] N. Matni. "Communication delay co-design in H2 decentralized control Using atomic norm minimization". In: *52nd IEEE Conference on Decision and Control*. Dec. 2013, pp. 6522–6529. DOI: `10.1109/CDC.2013.6760921`.

[39] N. Matni. "Communication Delay Co-Design in H2 Distributed Control Using Atomic Norm Minimization". In: *IEEE Transactions on Control of Network Systems* PP.99 (2015), pp. 1–1. ISSN: 2325-5870. DOI: `10.1109/TCNS.2015.2497100`.

[40] N. Matni and V. Chandrasekaran. "Regularization for Design". In: *IEEE Transactions on Automatic Control* PP.99 (2016), pp. 1–1. ISSN: 0018-9286. DOI: `10.1109/TAC.2016.2517570`.

[41] N. Matni and V. Chandrasekaran. "Regularization for design". In: *53rd IEEE Conference on Decision and Control*. Dec. 2014, pp. 1111–1118. DOI: `10.1109/CDC.2014.7039530`.

[42] Nikolai Matni. "Distributed Control Subject to Delays Satisfying an $\mathcal{H}_\infty$ Norm Bound". In: *2014 53rd IEEE Conference on Decision and Control (CDC)*. 2014. URL: `http://arxiv.org/pdf/1402.1559.pdf`.

[43] Nikolai Matni and John C Doyle. "A Theory of Dynamics, Control and Optimization in Layered Architectures". In: *The IEEE American Control Conference (ACC)*. 2016.

[44] Nikolai Matni, Ao Tang, and John C. Doyle. "A case study in network architecture tradeoffs". In: *ACM Sigcomm Symposium on SDN Research (SOSR), To appear*. 2015.

[45] Nader Motee and Ali Jadbabaie. "Approximation Methods and Spatial Interpolation in Distributed Control Systems". In: *2009 IEEE American Control Conference (ACC)*. June 2009.

[46] A. Nayyar, A. Mahajan, and D. Teneketzis. "Decentralized Stochastic Control with Partial History Sharing: A Common Information Approach". In: *IEEE Transactions on Automatic Control* 58.7 (July 2013), pp. 1644–1658. ISSN: 0018-9286. DOI: `10.1109/TAC.2013.2239000`.

[47] Reza Olfati-Saber. "Distributed Kalman filtering for sensor networks". In: *Decision and Control, 2007 46th IEEE Conference on*. IEEE. 2007, pp. 5492–5498.

[48] Neal Parikh and Stephen Boyd. "Proximal Algorithms". In: *Found. Trends Optim.* 1.3 (Jan. 2014), pp. 127–239. ISSN: 2167-3888.

[49] Xin Qi et al. "Structured optimal and robust control with multiple criteria: A convex solution". In: *Automatic Control, IEEE Transactions on* 49.10 (2004), pp. 1623–1640.

[50] Anders Rantzer. "Scalable control of positive systems". In: *European Journal of Control* 24 (2015), pp. 72–80.

[51] Daniel E Rivera, Manfred Morari, and Sigurd Skogestad. "Internal model control: PID controller design". In: *Industrial & engineering chemistry process design and development* 25.1 (1986), pp. 252–265.

[52] Michael Rotkowitz, Randy Cogill, and Sanjay Lall. "Convexity of optimal control over networks with delays and arbitrary topology". In: *Int. J. Syst., Control Commun.* 2.1/2/3 (Jan. 2010), pp. 30–54. ISSN: 1755-9340. DOI: `10.1504/IJSCC.2010.031157`. URL: `http://dx.doi.org/10.1504/IJSCC.2010.031157`.

[53] Michael Rotkowitz and Sanjay Lall. "A Characterization of Convex Problems in Decentralized Control". In: *Automatic Control, IEEE Transactions on* 51.2 (2006), pp. 274–286.

[54] Şurban Sabău and Nuno C. Martins. "Youla-Like Parametrizations Subject to QI Subspace Constraints". In: *Automatic Control, IEEE Transactions on* 59.6 (2014), pp. 1411–1422.

[55] Carsten W. Scherer. "Structured $\mathcal{H}_\infty$-Optimal Control for Nested Interconnections: A State-Space Solution". In: *Systems and Control Letters* 62 (12 2013), pp. 1105–1113.

[56] Sivaranjani Seetharaman et al. "Localization of disturbances in transportation systems". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015. DOI: `10.1109/CDC.2015.7402671`.

[57] Parikshit Shah and Pablo A Parrilo. "$\mathcal{H}_2$-optimal decentralized control over posets: A state space solution for state-feedback". In: *Decision and Control (CDC), 2010 49th IEEE Conference on*. 2010.

[58] Aleksandar Stevanovic et al. "Optimizing traffic control to reduce fuel consumption and vehicular emissions". In: *Transportation Research Record: Journal of the transportation research board* 2128.1 (2009), pp. 105–113.

[59] Takashi Tanaka and Pablo A. Parrilo. "Optimal Output Feedback Architecture for Triangular LQG Problems". In: *2014 IEEE American Control Conference (ACC)*. June 2014.

[60] Paul Tseng. "Applications of a Splitting Algorithm to Decomposition in Convex Programming and Variational Inequalities". In: *SIAM Journal on Control and Optimization* 29.1 (1991), pp. 119–138. DOI: `10.1137/0329006`.

[61] John N. Tsitsiklis and Michael Athans. "On the complexity of decentralized decision making and detection problems". In: *IEEE Conference on Decision and Control (CDC)*. 1984.

[62] A.S.M. Vamsi and N. Elia. "Design of distributed controllers realizable over arbitrary directed networks". In: *Decision and Control (CDC), 2010 49th IEEE Conference on*. Dec. 2010, pp. 4795–4800.

[63] A.S.M. Vamsi and N. Elia. "Optimal distributed controllers realizable over arbitrary networks". In: *IEEE Transactions on Automatic Control* 61.1 (2016), pp. 129–144.

[64] A.S.M. Vamsi and N. Elia. "Optimal realizable networked controllers for networked systems". In: *American Control Conference (ACC), 2011*. June 2011, pp. 336–341.

[65] Yin Wang, Jose Lopez, and Mario Sznaier. "Sparse Static Output Feedback Controller Design via Convex Optimization". In: *2014 53rd IEEE Conference on Decision and Control (CDC)*. 2014.

[66] Yuh-Shyang Wang. "Localized LQR with Adaptive Constraint and Performance Guarantee". In: *to appear in 2016 55th IEEE Conference on Decision and Control (CDC)*. 2016.

[67] Yuh-Shyang Wang and Nikolai Matni. "Localized Distributed Optimal Control with Output Feedback and Communication Delays". In: *IEEE 52nd Annual Allerton Conference on Communication, Control, and Computing*. 2014. DOI: 10.1109/ALLERTON.2014.7028511.

[68] Yuh-Shyang Wang and Nikolai Matni. "Localized LQG Optimal Control for Large-Scale Systems". In: *2016 IEEE American Control Conference (ACC)*. 2016. DOI: 10.1109/ACC.2016.7525205.

[69] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. "A System Level Approach to Controller Synthesis". In: *submitted to IEEE Transactions on Automatic Control* (2016).

[70] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. "Localized LQR Control with Actuator Regularization". In: *2016 IEEE American Control Conference (ACC)*. 2016. DOI: 10.1109/ACC.2016.7526485.

[71] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. "Localized LQR Optimal Control". In: *2014 53rd IEEE Conference on Decision and Control (CDC)*. 2014. DOI: 10.1109/CDC.2014.7039638.

[72] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. "Localized System Level Synthesis for Large-Scale Systems". In: *submitted to IEEE Transactions on Automatic Control* (2016).

[73] Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. "System Level Parameterizations, Constraints and Synthesis". In: *submitted to 2017 IEEE American Control Conference (ACC)*. 2017.

[74] Yuh-Shyang Wang, Seungil You, and Nikolai Matni. "Localized Distributed Kalman Filters for Large-Scale Systems". In: *5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*. 2015. DOI: 10.1016/j.ifacol.2015.10.306.

[75] Yuh-Shyang Wang et al. “Localized Distributed State Feedback Control with Communication Delays”. In: *2014 IEEE American Control Conference (ACC)*. June 2014. DOI: `10.1109/ACC.2014.6859440`.

[76] H. S. Witsenhausen. “A counterexample in stochastic optimum control”. In: *SIAM Journal of Control* 6.1 (1968).

[77] Dante C. Youla, Hamid A. Jabr, and Joseph J. Bongiorno Jr. “Modern Wiener-Hopf Design of Optimal Controllers-Part II: The Multivariable Case”. In: *Automatic Control, IEEE Transactions on* 21.3 (1976), pp. 319–338.

[78] Ming Yuan and Yi Lin. “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), pp. 49–67.

[79] Changhong Zhao, Enrique Mallada, and Steven H Low. “Distributed generator and load-side secondary frequency control in power networks”. In: *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*. IEEE. 2015, pp. 1–6.

[80] Changhong Zhao et al. “Design and Stability of Load-Side Primary Frequency Control in Power Systems”. In: *Automatic Control, IEEE Transactions on* 59.5 (May 2014), pp. 1177–1189.

[81] K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall New Jersey, 1996.