

# System Level Synthesis

James Anderson, John C. Doyle, Steven Low, Nikolai Matni

April 4, 2019

## Abstract

This article surveys the *System Level Synthesis* framework, which presents a novel perspective on constrained robust and optimal controller synthesis for linear systems. We show how SLS shifts the controller synthesis task from the design of a controller to the design of the entire closed loop system, and highlight the benefits of this approach in terms of scalability and transparency. We emphasize two particular applications of SLS, namely large-scale distributed optimal control and robust control. In the case of distributed control, we show how SLS allows for localized controllers to be computed, extending robust and optimal control methods to large-scale systems under practical and realistic assumptions. In the case of robust control, we show how SLS allows for novel design methodologies that, for the first time, quantify the degradation in performance of a robust controller due to model uncertainty – such transparency is key in allowing robust control methods to interact, in a principled way, with modern techniques from machine learning and statistical inference. Throughout, we emphasize practical and efficient computational solutions, and demonstrate our methods on easy to understand case studies.

## 1 Introduction

The systems we seek to design and control are becoming increasingly complex, be it in their dynamics, their scale, or their interaction with the environment. To address this complexity, control methods must be able to accommodate the large-scale distributed nature of many of these systems (e.g., smart-grids, intelligent transportation systems, software defined networks). But this in itself is often not sufficient, as certain components of the system, or the environment in which it must interact, are too difficult or impossible to model from first principles, thus motivating a principled integration of data-driven techniques. While there has been a tremendous amount of work in each of these areas, usually targeting a specific problem setting, what is lacking is a unified framework that allows for all of these issues to be simultaneously addressed in a principled manner. This paper summarizes our progress towards developing such a unified framework in the context of linear optimal control. As we describe in more detail below, by moving from controller synthesis to *System Level Synthesis*, wherein the closed loop behavior and properties of the system are directly optimized over, we are able to make significant improvements over the state of the art in terms of computational scalability, richness of problems that can be tackled, and the principled integration of models produced by data-driven methods, among other things.

We begin by providing a brief overview of relevant results from classical optimal control, robust control, and distributed optimal control. We then highlight our main contributions, and end this section by providing a roadmap to the reader for the rest of the paper. Our goal is to convey the core concepts of SLS to as broad an audience as possible, while also providing deep dives

into specific domains (such as robust and distributed control), in hopes of appealing to both the interested graduate student (be they specializing in control theory, optimization, machine learning, or artificial intelligence), as well as the domain experts.

## 1.1 Prior Work

### 1.1.1 Classical Controller Parameterizations and Synthesis Results

The foundation of many optimal controller synthesis procedures is a parameterization of all internally stabilizing controllers, and the responses that they achieve, over which relevant performance measures can be easily optimized. For finite dimensional linear-time-invariant (LTI) systems, the class of internally stabilizing LTI feedback controllers is characterized by the celebrated Youla parameterization [1] and the closely related factorization approach [2]. In [1], the authors showed that the Youla parameterization defines 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 directly optimized. This allowed for the incorporation of customized design specifications on the closed loop system into the controller design process via convex optimization [3] or interpolation [4]. Subsequently, analogous parameterizations of stabilizing controllers for more general classes of systems were developed: notable examples include the polynomial approach [5] for generalized Rosenbrock systems [6] and the behavioral approach [7, 8, 9, 10] for linear differential systems. These results illustrate the power and generality of the Youla parameterization and factorization approaches to optimal control in the centralized setting. Together with state-space methods, they 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, and in the LTI setting, paved the way for the foundational results of robust and optimal control that would follow [11].

### 1.1.2 Robust Control

Building on these classical results, a rich line of work looked to explicitly account for modeling uncertainty in the design procedure. In particular, systems were typically viewed as being composed of a nominal system estimate and a set of unknown but bounded model errors. It was therefore necessary to ensure that the computed controller ensured stability and performance guarantees for any such admissible realization. When modeling errors to the nominal system are allowed to be arbitrary norm-bounded linear time-invariant (LTI) operators in feedback with the nominal plant, traditional small-gain theorems and robust synthesis techniques can be applied to exactly solve the problem [12, 13]. However, when the errors are known to have more structure there are more sophisticated techniques based on structured singular values and corresponding  $\mu$ -synthesis techniques [14, 15, 16, 17] or integral quadratic constraints (IQCs) [18].

While theoretically appealing and much less conservative than traditional small-gain approaches, the resulting synthesis methods are both computationally intractable (although effective heuristics do exist) and difficult to interpret analytically. In particular, we know of no results in the literature that bound the degradation in performance of controlling an uncertain system in terms of the size of the perturbations affecting it. While traditional results in robust control considered fixed model uncertainty, in the data-driven and learning based control setting, analytic interpretability of the effects uncertainty size are essential in obtaining sample-complexity bounds for these methods [19, 20, 21, 22].

### 1.1.3 Distributed Control

However, as control engineers shifted their attention from centralized to distributed optimal control, it was observed that the parameterization approaches that were so fruitful in the centralized setting were no longer directly applicable. In contrast to centralized systems, 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 exchange locally available information (such as sensor measurements or applied control actions) via a communication network. These information sharing constraints make the corresponding distributed optimal controller synthesis problem challenging to solve [23, 24, 25, 26, 27, 28]. In particular, imposing such structural constraints on the controller can lead to optimal control problems that are NP-hard [29, 30].

Despite these technical and conceptual challenges, a body of work [26, 31, 27, 25, 24, 28] that began in the early 2000s, and that culminated with the introduction of “quadratic invariance” (QI) in the seminal paper [25], showed that for a large class of practically relevant LTI systems, such internal structure could be integrated with 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 [32]. Even more remarkable is that this condition is tight, in the sense that QI is a necessary [33] and sufficient [25] 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 triggered an explosion of results in distributed optimal controller synthesis [34, 35, 36, 37, 38, 39, 40, 41, 42] – these results showed that the robust and optimal control methods that proved so powerful for centralized systems could be ported to the distributed setting. As far as we are aware, no such results exist for the more general classes of systems considered in [5, 7, 8, 9, 10].

However, a fact that is not emphasized in the distributed optimal control literature is that distributed controllers are actually more complex to synthesize and implement than their centralized counterparts.<sup>1</sup> In particular, a major limitation of the QI framework is that, for strongly connected systems,<sup>2</sup> it cannot provide a convex characterization of localized controllers, in which local sub-controllers only access a subset of system-wide measurements. This need for global exchange of information between sub-controllers is a limiting factor in the scalability of the synthesis and implementation of these distributed optimal controllers.

## 1.2 Main Contributions

Motivated by the issues raised above, we propose a novel parameterization of internally stabilizing controllers and the closed loop responses that they achieve, providing an alternative approach to constrained optimal and robust controller synthesis. Specifically, rather than directly designing only the feedback loop between sensors and actuators, as in the Youla framework, 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 pro-

---

<sup>1</sup>For example, see the solutions presented in [34, 35, 36, 37, 38, 39, 40, 41, 42] and the message passing implementation suggested in [42].

<sup>2</sup>We say that a plant is strongly connected if the state of any subsystem can eventually alter the state of all other subsystems.

posed method a System Level Approach (SLA) to controller synthesis, which is composed of three elements: System Level Parameterizations (SLPs), System Level Constraints (SLCs) and System Level Synthesis (SLS) problems. Further, in contrast to the QI framework, which seeks to impose structure on the input/output map between sensor measurements and control actions, the SLA imposes structural constraints on the system response itself, and shows that this structure carries over to the *internal realization* of the corresponding controller. It is this conceptual shift from structure on the input/output map to the internal realization of the controller that allows us to expand the class of structured controllers that admit a convex characterization, and in doing so, vastly increase the scalability of distributed optimal control methods. Further, by moving our focus from controllers to system responses, we are able to make transparent the effects of modeling uncertainty and approximations on controller synthesis and performance. We summarize our main contributions below.

This paper summarizes, and in some places extends, our recent theoretical and computational contributions [43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 19] to the area of constrained optimal and robust controller synthesis. In particular, we

- define and analyze the system level approach to controller synthesis, which is built around novel SLPs of all stabilizing controllers and the closed loop responses that they achieve;
- in the state-feedback setting, we show that the above parameterizations are stable under small perturbations, leading to novel robust control problems that, among other features, allow us to bound performance degradation as a function of modeling uncertainty size;
- show that SLPs allow us to constrain the closed loop response of the system to lie in arbitrary sets: we call such constraints on the system SLCs. If these SLCs admit a convex representation, then the resulting set of constrained system responses admits a convex representation as well;
- show that such 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;
- show that the set of constrained stabilizing controllers that admit a convex parameterization using SLPs and SLCs is a strict superset of those that can be parameterized using quadratic invariance – hence we provide a generalization of the QI framework, characterizing the broadest known class of constrained controllers that admit a convex parameterization;
- formulate and analyze the SLS problem, which exploits SLPs and SLCs to define the broadest known class of constrained optimal control problems that can be solved using convex programming. We show that the optimal control problems considered in the QI literature [24], as well as the recently defined localized optimal control framework [46] are all special cases of SLS problems;
- show that if *locality* constraints are imposed on the system response, and the SLCs admit a certain (partial) decomposability property, the computational complexity of controller synthesis and implementation can be made to scale as  $O(1)$  relative the size of the full system, assuming sufficient parallel computation.

### 1.3 Paper Structure

The paper is organized as follows. We begin with a gentle warmup in Section 2, wherein we consider full-state-feedback problems over a finite time horizon – by restricting ourselves to this simplified setting, we are able to build up nearly all of the core machinery of SLS with minimal technical overhead. We show how several known optimal and robust control synthesis problems can be posed in the SLS framework, essentially turning (robust) optimal control problems into (robust) optimization problems over system responses.

In Section 3 we formally present the general optimal control problem formulation, and briefly review classical solution approaches. With this information in hand, the finite time horizon results of Section 2 are extended to the infinite horizon setting in Section 4. The notion of locality is introduced in Section 4.2 and then used to show how to decompose the synthesis problem into uncoupled subproblems of reduced dimension in Section 4.3. We end with a robust extension of the previous results, and show how it can be used to guarantee near-optimal performance of computationally tractable solutions to SLS problems. Finally, in Section 5, we extend these results to the output-feedback setting. We end with conclusions in Section 6.

## 2 Working with System Responses

To introduce the notion of a system response, and to illustrate some of its benefits, we begin by considering linear optimal control problems over a finite time horizon with full state-feedback. We show how the corresponding controller synthesis task can be cast as an optimization over the closed loop behavior of the system, i.e., over system responses, as opposed to over the controller itself. By initially restricting ourselves to a finite time horizon, we are not burdened by the technical overhead associated with system stability, controller internal stability, or infinite dimensional optimization problems. In subsequent sections, we build upon the material presented here and extend these methods to infinite horizon problems, distributed control problems, and to the output-feedback setting.

### 2.1 Finite-horizon System Level Synthesis

Consider the linear-time-varying (LTV) system

$$x_{t+1} = A_t x_t + B_t u_t + w_t, \quad (2.1)$$

where  $x_t \in \mathbb{R}^n$  is the state,  $u_t \in \mathbb{R}^p$  is the control input, and  $w_t \in \mathbb{R}^n$  is an exogenous disturbance process. Our goal is to study the behavior of this system over the horizon  $t = 0, \dots, T$  when the control input  $u_t$  is a causal LTV state-feedback controller, i.e.,  $u_t = K_t(x_0, x_1, \dots, x_t)$  for some linear map  $K_t$ . To do so, we introduce the following signals and block-lower-triangular matrices:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_T \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_T \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{T-1} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} K^{0,0} & & & \\ K^{1,1} & K^{1,0} & & \\ \vdots & \ddots & \ddots & \\ K^{T,T} & \dots & K^{T,1} & K^{T,0} \end{bmatrix}, \quad (2.2)$$

where here  $\mathbf{K}$  should be thought of as the matrix representation of the convolution operator induced by the linear-time-varying controller  $K_t$  such that  $u_t = \mathbf{K}^{t,*} \mathbf{x}$ , for  $\mathbf{K}^{t,*}$  the  $t$ -th row of  $\mathbf{K}$ .

Further, we let  $Z$  be the block-downshift operator, i.e., a matrix with identity matrices along its first block sub-diagonal and zeros elsewhere, and define

$$\mathcal{A} := \text{blkdiag}(A_0, A_1, \dots, A_{T-1}, 0), \quad \mathcal{B} := \text{blkdiag}(B_0, B_1, \dots, B_{T-1}, 0).$$

Note that in the linear-time-invariant (LTI) setting,  $\mathcal{A}$  and  $\mathcal{B}$  are block diagonal matrices with the same matrix being repeated along the diagonal. Using these definitions allows us to compactly rewrite the behavior of the system (2.1) over the horizon  $t = 0, \dots, T$  as

$$\begin{aligned} \mathbf{x} &= Z\mathcal{A}\mathbf{x} + Z\mathcal{B}\mathbf{u} + \mathbf{w} \\ &= Z(\mathcal{A} + \mathcal{B}\mathbf{K})\mathbf{x} + \mathbf{w}. \end{aligned} \quad (2.3)$$

In particular, the closed loop behavior of our system under the feedback law  $\mathbf{K}$  can be entirely characterized by the following maps,

$$\begin{aligned} \mathbf{x} &= (I - Z(\mathcal{A} + \mathcal{B}\mathbf{K}))^{-1}\mathbf{w} \\ \mathbf{u} &= \mathbf{K}(I - Z(\mathcal{A} + \mathcal{B}\mathbf{K}))^{-1}\mathbf{w}, \end{aligned} \quad (2.4)$$

which describe the closed loop transfer functions from the exogenous disturbance  $\mathbf{w}$  to the state  $\mathbf{x}$  and control input  $\mathbf{u}$ , respectively. These maps are what we call *system responses*, and we now show how optimal controller synthesis can be performed by directly optimizing over these system responses, as opposed to the controller map  $\mathbf{K}$  itself.

To that end, we introduce two additional lower-block-triangular operators

$$\Phi_x = \begin{bmatrix} \Phi_x^{0,0} & & & \\ \Phi_x^{1,1} & \Phi_x^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \Phi_x^{T,T} & \dots & \Phi_x^{T,1} & \Phi_x^{T,0} \end{bmatrix}, \quad \Phi_u = \begin{bmatrix} \Phi_u^{0,0} & & & \\ \Phi_u^{1,1} & \Phi_u^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \Phi_u^{T,T} & \dots & \Phi_u^{T,1} & \Phi_u^{T,0} \end{bmatrix}. \quad (2.5)$$

Here we use  $\Phi_x$  to denote the system response describing the map from  $\mathbf{w} \rightarrow \mathbf{x}$ , and similarly,  $\Phi_u$  to denote the system response describing the map from  $\mathbf{w} \rightarrow \mathbf{u}$  – this allows us to compactly rewrite equation (2.4) as

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \mathbf{w}. \quad (2.6)$$

Our objective is to optimize directly over these system responses  $\{\Phi_x, \Phi_u\}$ : in order to do so, we must ensure that there exists a linear-time-varying controller  $\mathbf{K}$  such that equations (2.4) and (2.6) are equivalent. The next theorem shows that enforcing the system responses to obey the system dynamics is necessary and sufficient to ensure the existence of such a corresponding controller, i.e., to ensure that the pair  $\{\Phi_x, \Phi_u\}$  is *achievable*.

**Theorem 2.1.** *Over the horizon  $t = 0, \dots, T$ , the system dynamics (2.1) with block-lower-triangular state feedback law  $\mathbf{K}$  defining the control action as  $\mathbf{u} = \mathbf{K}\mathbf{x}$ , the following are true*

1. the affine subspace defined by

$$[I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \quad (2.7)$$

*parameterizes all possible system responses (2.6).*

2. for any block-lower-triangular matrices  $\{\Phi_x, \Phi_u\}$  satisfying (2.7), the controller  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  achieves the desired response.

**Proof.** Proof of 1.: Let  $\mathbf{K}$  be any block-lower-triangular operator, and  $\mathbf{u} = \mathbf{K}\mathbf{x}$ . Then as argued above, we have that

$$\begin{aligned}\mathbf{x} &= (I - Z(\mathcal{A} + \mathcal{B}\mathbf{K}))^{-1}\mathbf{w} \\ \mathbf{u} &= \mathbf{K}(I - Z(\mathcal{A} + \mathcal{B}\mathbf{K}))^{-1}\mathbf{w}.\end{aligned}$$

It is then easily seen that

$$[I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} (I - Z(\mathcal{A} + \mathcal{B}\mathbf{K}))^{-1} \\ \mathbf{K}(I - Z(\mathcal{A} + \mathcal{B}\mathbf{K}))^{-1} \end{bmatrix} = (I - Z\mathcal{A} - Z\mathcal{B}\mathbf{K})(I - Z(\mathcal{A} + \mathcal{B}\mathbf{K}))^{-1} = I.$$

Proof of 2.: First notice that the affine constraint (2.7) implies that  $\Phi_x^{t,0} = I$  for  $t = 0, \dots, T$ , and hence  $\Phi_x^{-1}$  exists. Let  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  and notice that

$$\mathbf{x} = (I - Z(\mathcal{A} + \mathcal{B}\Phi_u \Phi_x^{-1}))^{-1}\mathbf{w}.$$

But we then have that

$$(I - Z(\mathcal{A} + \mathcal{B}\Phi_u \Phi_x^{-1}))^{-1} = ((I - Z\mathcal{A})\Phi_x - Z\mathcal{B}\Phi_u)\Phi_x^{-1})^{-1} = \Phi_x((I - Z\mathcal{A})\Phi_x - Z\mathcal{B}\Phi_u)^{-1} = \Phi_x$$

where the last equality follows from the fact that  $\{\Phi_x, \Phi_u\}$  satisfy (2.7). Similarly we have that

$$\mathbf{u} = \Phi_u \Phi_x^{-1}\mathbf{x} = \Phi_u \Phi_x^{-1}\Phi_x \mathbf{w} = \Phi_u \mathbf{w},$$

where the second equality follows from the fact that  $\mathbf{x} = \Phi_x \mathbf{w}$ . ■

At this point we emphasize that there is nothing “distributed” about the formulation or Theorem 2.1. As we will see in later sections, the implications of Theorem 2.1 and its various generalizations will provide the building blocks for a unified and scalable distributed control framework.

Finally, before providing examples of how this parameterization can be used to formulate SLS problems, we make a small comment on controller realization. Even in the finite horizon setting, where no issues relating internal stability arise, one may wish to avoid inverting the matrix  $\Phi_x$  to compute the map  $\mathbf{K}$  – reasons may include numerical ill-condition and prohibitive computational cost for larger horizons. We show here that one can instead implement this inverse using a particular feedback interconnection – in particular, one can show that the following implementation:

$$\begin{aligned}u_t &= \sum_{\tau=1}^T \Phi_u(t, \tau) \hat{w}_{t-\tau} \\ \hat{x}_{t+1} &= \sum_{\tau=1}^{T-1} \Phi_x(t+1, \tau+1) \hat{w}_{t-\tau} \\ \hat{w}_t &= x_{t+1} - \hat{x}_{t+1},\end{aligned}\tag{2.8}$$

leads to the desired mapping  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  from  $\mathbf{x} \rightarrow \mathbf{u}$ . An added benefit of this realization, which we exploit heavily in the sequel, is that any structure imposed on the system level response  $\{\Phi_x, \Phi_u\}$  carries over directly to the internal structure of the controller realization, allowing a natural and transparent way to impose distributed constraints on the controller realization.

## 2.2 System Level Synthesis

The benefit of the previous parameterization is that the system behavior is now entirely defined in terms of *affine* functions of the disturbance process (i.e., (2.7)), and further to ensure achievability it suffices to constrain the maps  $\{\Phi_x, \Phi_u\}$  to lie in an affine space. We now show how this can be used to turn standard optimal control problems into convex optimization problems. The process of converting an optimal control problem to one over system responses consists of three steps:

1. Rewrite the problem using the vector notation (2.2).
2. Set  $\mathbf{x} = \Phi_x \mathbf{w}$ , and  $\mathbf{u} = \Phi_u \mathbf{w}$ , and constrain  $\{\Phi_x, \Phi_u\}$  to lie in the affine space (2.7).
3. Use knowledge about the disturbance process  $\mathbf{w}$  to derive an appropriate objective function (see below for LQR,  $\mathcal{H}_\infty$ , and  $\mathcal{L}_1$  optimal control).

We now provide some worked examples of classical centralized control problem posed in the SLS framework.

### 2.2.1 LQR with no driving noise and known $x_0$

Here we let

$$\mathbf{w} = \begin{bmatrix} x_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (2.9)$$

For simplicity of notation, we assume that  $u_T$  is included in the cost functional, i.e., that the optimal control problem that we seek to solve is given by

$$\begin{aligned} \min_{x_t, u_t} \quad & \sum_{t=0}^T x_t^\top Q_t x_t + u_t^\top R_t u_t \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t, \quad t = 0, \dots, T-1, \\ & x_0 \text{ known.} \end{aligned} \quad (2.10)$$

Using vector notation, this problem can be written as

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \right\|_F^2 \\ \text{subject to} \quad & \mathbf{x} = Z\mathcal{A}\mathbf{x} + Z\mathcal{B}\mathbf{u}, \end{aligned} \quad (2.11)$$

where

$$\mathcal{Q} := \text{blkdiag}(Q_0, Q_1, \dots, Q_T), \quad \mathcal{R} := \text{blkdiag}(R_0, R_1, \dots, R_T). \quad (2.12)$$

Now, we use the fact that  $\mathbf{x} = \Phi_x \mathbf{w} = \Phi_x(:, 0)x_0$  and  $\mathbf{u} = \Phi_u \mathbf{w} = \Phi_u(:, 0)x_0$ , where we use  $\Phi(:, 0)$  to denote the first block column of a matrix  $\Phi$ , and Theorem 2.1, to cast this problem as an optimization over system responses:

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x(:, 0) \\ \Phi_u(:, 0) \end{bmatrix} x_0 \right\|_F^2 \\ \text{subject to} \quad & [I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x(:, 0) \\ \Phi_u(:, 0) \end{bmatrix} = I, \end{aligned} \quad (2.13)$$

which can be expressed as a convex quadratic program.



### 2.2.2 LQR with driving noise and random $x_0$

Here we let  $\mathbf{w} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma_w)$ . For simplicity of notation, we assume that  $u_T$  is included in the cost functional, i.e., that the optimal control problem that we seek to solve is given by

$$\begin{aligned} \min_{x_t, u_t} \quad & \sum_{t=0}^T \mathbb{E} [x_t^\top Q_t x_t + u_t^\top R_t u_t] \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t + w_t. \end{aligned} \quad (2.14)$$

Following the same steps as the previous subsection, this problem can be rewritten as

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \Sigma_w^{1/2} \right\|_F^2 \\ \text{subject to} \quad & [I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \end{aligned} \quad (2.15)$$

where  $Q$  and  $\mathcal{R}$  are as defined above. Once again, this problem can be expressed as a convex quadratic program.

### 2.2.3 $\mathcal{H}_\infty$ optimal control

Consider the following  $\mathcal{H}_\infty$  optimal control problem

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \max_{\|\mathbf{w}\|_2 \leq 1} \quad & \sum_{t=0}^T x_t^\top Q_t x_t + u_t^\top R_t u_t \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t + w_t. \end{aligned} \quad (2.16)$$

which can be rewritten in vector notation as

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \max_{\|\mathbf{w}\|_2 \leq 1} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{x} = Z\mathcal{A}\mathbf{x} + Z\mathcal{B}\mathbf{u} + \mathbf{w}. \end{aligned} \quad (2.17)$$

Now, using that  $\mathbf{x} = \Phi_x \mathbf{w}$  and  $\mathbf{u} = \Phi_u \mathbf{w}$ , we can rewrite our control problem in terms of system responses as follows

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \max_{\|\mathbf{w}\|_2 \leq 1} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \mathbf{w} \right\|_2^2 \\ \text{subject to} \quad & [I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I. \end{aligned} \quad (2.18)$$

Now, it suffices to notice that the inner maximization over  $\mathbf{w}$  is simply the definition of the spectral norm of the weighted system responses, leading to the final optimization problem over  $\{\Phi_x, \Phi_u\}$

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{2 \rightarrow 2}^2 \\ \text{subject to} \quad & [I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \end{aligned} \quad (2.19)$$

which can be expressed as a semidefinite program.

### 2.2.4 $\mathcal{L}_1$ optimal control

The previous subsection optimized the worst case  $\ell_2 \rightarrow \ell_2$  gain of our system – a similar derivation can be used to obtain a problem that optimizes the worst case  $\ell_\infty \rightarrow \ell_\infty$  gain of our system. In particular, we skip a step and simply begin with the following problem in vector notation

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \max_{\|\mathbf{w}\|_\infty \leq 1} & \left\| \begin{bmatrix} \mathcal{Q}^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \right\|_\infty \\ \text{subject to} & \mathbf{x} = Z\mathcal{A}\mathbf{x} + Z\mathcal{B}\mathbf{u} + \mathbf{w}. \end{aligned} \quad (2.20)$$

Once again, using that  $\mathbf{x} = \Phi_x \mathbf{w}$  and  $\mathbf{u} = \Phi_u \mathbf{w}$ , we can rewrite our control problem in terms of system responses as follows

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \max_{\|\mathbf{w}\|_\infty \leq 1} & \left\| \begin{bmatrix} \mathcal{Q}^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \mathbf{w} \right\|_\infty \\ \text{subject to} & [I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I. \end{aligned} \quad (2.21)$$

Now, it suffices to notice that the inner maximization over  $\mathbf{w}$  is simply the definition of the  $\mathcal{L}_1$  norm (i.e., the matrix  $\ell_\infty \rightarrow \ell_\infty$  induced norm) of the weighted system responses, leading to the final optimization problem over  $\{\Phi_x, \Phi_u\}$

$$\begin{aligned} \min_{\Phi_x, \Phi_u} & \left\| \begin{bmatrix} \mathcal{Q}^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\infty \rightarrow \infty} \\ \text{subject to} & [I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \end{aligned} \quad (2.22)$$

which can be expressed as a linear program.

### 2.2.5 General Constrained Optimal Control

The four example problems described above can be viewed as specific instantiations of a generic problem structure. They each contain the System Level Parameterization (SLP) (2.7) as a constraint, and the cost-function is some appropriately function of the system responses  $\{\Phi_x, \Phi_u\}$ . Although not included in the examples above, we could also additionally impose that the system responses lie in some sets, i.e., that  $\{\Phi_x, \Phi_u\} \in \mathcal{S}$ , or that the state and input trajectories also lie in some sets, i.e., that  $\Phi_x \mathbf{w} \in \mathcal{X}$  and  $\Phi_u \mathbf{w} \in \mathcal{U}$  for all  $\mathbf{w} \in \mathcal{W}$ . Recent work in this direction is presented in [56, 57].

Combining all of these elements together, we obtain a System Level Synthesis (SLS) problem:

$$\min_{\Phi_x, \Phi_u} g(\Phi_x, \Phi_u) \quad (2.23a)$$

$$\text{subject to} \quad [I - Z\mathcal{A} \quad -Z\mathcal{B}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \quad (2.23b)$$

$$\{\Phi_x, \Phi_u\} \in \mathcal{S}, \quad (2.23c)$$

$$\Phi_x \mathbf{w} \in \mathcal{X}, \Phi_u \mathbf{w} \in \mathcal{U} \text{ for all } \mathbf{w} \in \mathcal{W}. \quad (2.23d)$$

Clearly, if the objective functions and constraints are chosen to be convex functions, then the SLS problem is itself convex, as is the case for all of the examples given above.

To reiterate, the SLP is the affine constraint (2.23b) that ensures the system is achievable over the horizon  $t = 0, \dots, T$ , i.e. it ensures that there exists a controller  $\mathbf{K}$  such that system response  $\{\Phi_x, \Phi_u\}$  described by (2.6) and the closed-loops maps  $\mathbf{w} \rightarrow \mathbf{x}$  and  $\mathbf{w} \rightarrow \mathbf{u}$  as given by (2.4) are equivalent. System Level Constraints (SLCs) modeled by (2.23d) force the closed loop response to lie in a pre-described set  $\mathcal{S}$ . Typical choices for additional constraints  $\mathcal{S}$  imposed on the system responses would include additional performance requirements (resulting in multi-objective control problems,) and structural constraints (as imposed by distributed constraints), or sparsity surrogate constraints (as imposed by architectural constraints). In Section 4.2 we will describe a particularly useful class of SLCs that define spatiotemporal locality constraints. Additionally, the constraints  $\Phi_x \mathbf{w} \in \mathcal{X}$  and  $\Phi_u \mathbf{w} \in \mathcal{U}$  can be used to incorporate state and input constraints. Typical choices for constraints  $\mathcal{X} \times \mathcal{U}$  on the state and input trajectories include, for example, polytopic or ellipsoidal constraints. At this point, the connection between the finite time horizon SLS problem (2.23) and the subproblems solved in model predictive control should be apparent.

### 2.3 A new perspective on robustness

Thus far, we have only shown that SLS can be used to cast standard optimal control problems as optimizations over system responses. As a preview of some of the additional benefits of working with system responses, we present here a novel robust control synthesis method that allows us to bound the performance degradation incurred by using approximate system responses that do not exactly satisfy the achievability constraints (2.7). In later sections, we extend these robustness results to the infinite horizon setting.

Our first result is a robust variant of Theorem 2.1, which characterizes the behavior achieved by a controller built from approximate system responses. Before stating the result, we let

$$\Delta = \begin{bmatrix} \Delta^{0,0} & & & \\ \Delta^{1,1} & \Delta^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \Delta^{T,T} & \dots & \Delta^{T,1} & \Delta^{T,0} \end{bmatrix} \quad (2.24)$$

be an arbitrary block-lower-triangular matrix.

**Theorem 2.2.** *Let  $\Delta$  be defined as in equation (2.24), and suppose that  $\{\Phi_x, \Phi_u\}$  satisfy*

$$[I - ZA \quad -ZB] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I + \Delta. \quad (2.25)$$

*If  $(I + \Delta^{i,0})^{-1}$  exists for  $i = 1, \dots, T$ , then the controller  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  achieves the system response*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} (I + \Delta)^{-1} \mathbf{w} \quad (2.26)$$

**Proof.** If  $(I + \Delta^{i,0})^{-1}$  exists for  $i = 1, \dots, T$ , then so does  $(I + \Delta)^{-1}$ . Therefore the constraint (2.25) is equivalent to

$$[I - ZA \quad -ZB] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} (I + \Delta)^{-1} = I. \quad (2.27)$$

Then noting that  $\mathbf{K} = \Phi_u \Phi_x^{-1} = \Phi_u (I + \Delta)^{-1} (\Phi_x (I + \Delta)^{-1})^{-1}$ , we have, by Theorem 2.1, that the controller  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  achieves the system responses (2.26). ■

We now present a simple example of how this theorem can be used to compute a controller that is robust to parametric uncertainty in the system dynamics  $(\mathcal{A}, \mathcal{B})$  – we return to this theorem and its infinite horizon analog later in the text, at which point we present several other use-cases.

Suppose that rather than having access to the true system dynamics  $(\mathcal{A}, \mathcal{B})$ , we instead only have access to estimates  $(\hat{\mathcal{A}}, \hat{\mathcal{B}})$ , and further let  $\{\hat{\Phi}_x, \hat{\Phi}_u\}$  satisfy the achievability constraints defined by  $(\hat{\mathcal{A}}, \hat{\mathcal{B}})$ . Then simple algebra shows that the approximate system responses  $\{\hat{\Phi}_x, \hat{\Phi}_u\}$  satisfy

$$\begin{bmatrix} I - Z\mathcal{A} & -Z\mathcal{B} \end{bmatrix} \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} = I + Z \begin{bmatrix} \mathcal{D}_A & \mathcal{D}_B \end{bmatrix} \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix}, \quad (2.28)$$

where  $\mathcal{D}_A := \hat{\mathcal{A}} - \mathcal{A}$  and  $\mathcal{D}_B := \hat{\mathcal{B}} - \mathcal{B}$ . We can then invoke Theorem 2.2 with

$$\Delta := Z \begin{bmatrix} \mathcal{D}_A & \mathcal{D}_B \end{bmatrix} \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix}$$

to conclude that the controller  $\hat{\mathbf{K}} = \hat{\Phi}_u \hat{\Phi}_x^{-1}$ , computed using only the system estimates  $(\hat{\mathcal{A}}, \hat{\mathcal{B}})$ , achieves the following response on the actual system  $(\mathcal{A}, \mathcal{B})$ :

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} (I + \Delta)^{-1} \mathbf{w}.$$

If we further suppose that we have known bounds on our modeling error, i.e., that  $\|\mathcal{D}_A\|_{2 \rightarrow 2} \leq \epsilon_A$  and  $\|\mathcal{D}_B\|_{2 \rightarrow 2} \leq \epsilon_B$ , then we can use this insight to formulate a robust SLS problem. Recall that for matrices, the induced  $2 \rightarrow 2$  norm is simply the maximum singular value. For illustrative purposes, we formulate a robust LQR problem, but the general approach is applicable to any performance metrics that have a submultiplicative property. In particular, consider the problem presented in Section 2.2.2, and assume for simplicity that  $\Sigma_w = I$ . Then, assuming we only have access to the approximate dynamics  $(\hat{\mathcal{A}}, \hat{\mathcal{B}})$ , we can pose the following robust optimization problem

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \quad & \max_{\|\mathcal{D}_A\| \leq \epsilon_A, \|\mathcal{D}_B\| \leq \epsilon_B} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} (I + \Delta)^{-1} \right\|_F^2 \\ \text{subject to} \quad & \begin{bmatrix} I - Z\hat{\mathcal{A}} & -Z\hat{\mathcal{B}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I. \end{aligned} \quad (2.29)$$

By considering a performance metric that maximizes over the admissible model uncertainty, we are able to provide robust performance guarantees over the uncertainty set. However, as posed, this problem is non-convex and does not admit a computationally efficient solution.

We can however exploit the structure of the problem to derive a tractable upper bound. In particular, notice that the effect of the model uncertainty, as captured by the  $(I + \Delta)^{-1}$  term, can be isolated by applying the submultiplicative property of the spectral and frobenius norms, and the strictly-lower-block-triangular structure of  $\Delta$  can be further exploited to compute an upper bound

on its effect. Specifically, we have that

$$\begin{aligned} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} (I + \Delta)^{-1} \right\|_F^2 &\leq \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_F^2 \left\| (I + \Delta)^{-1} \right\|_{2 \rightarrow 2}^2 \\ &\leq \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_F^2 \left( \sum_{t=0}^{T-1} \|\Delta\|_{2 \rightarrow 2}^t \right)^2, \end{aligned}$$

where the first inequality follows from the submultiplicative property of the spectral and Frobenius norms, and the second inequality by noticing that because  $\Delta$  is strictly-lower-block-triangular, the inverse can be expressed as a finite series and applying the submultiplicative property and the triangle inequality.

Finally, we note that using Proposition 3.5 from [19], we can upper bound the spectral norm of  $\Delta$  as follows:

$$\|\Delta\|_{2 \rightarrow 2} = \left\| Z \begin{bmatrix} \mathcal{D}_A & \mathcal{D}_B \end{bmatrix} \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} \right\|_{2 \rightarrow 2} \leq \sqrt{2} \left\| \begin{bmatrix} \epsilon_A \Phi_x \\ \epsilon_B \Phi_u \end{bmatrix} \right\|_{2 \rightarrow 2}.$$

Leveraging these two inequalities, we can now write a tractable (quasi-convex) optimization problem that provides an upper-bound to our original robust optimization problem (2.29).

$$\begin{aligned} \min_{\tau} \left( \sum_{t=0}^{T-1} \tau^t \right)^2 \min_{\Phi_x, \Phi_u} &\left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & \mathcal{R}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_F^2 \\ \text{subject to} &\begin{bmatrix} I - Z\hat{A} & -Z\hat{B} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \\ &\sqrt{2} \left\| \begin{bmatrix} \epsilon_A \Phi_x \\ \epsilon_B \Phi_u \end{bmatrix} \right\|_{2 \rightarrow 2} \leq \tau. \end{aligned} \tag{2.30}$$

Thus, by leveraging the transparent mapping of system uncertainty to system performance, we were able to derive a tractable upper bound that produces a controller that is both guaranteed to be robustly stabilizing and performant. Further, as we show in Section 4.5, this transparency also allows us to derive bounds on performance degradation as a function of the uncertainty sizes  $\epsilon_A$  and  $\epsilon_B$ . Although we defer presenting this result to the infinite horizon setting, as the computations become less cumbersome in that case, we note that analogous, albeit messier, degradation bounds can be derived for the robust optimization problem (2.30).

### 3 Preliminaries and Notation

We now formally introduce the infinite horizon optimal control problems that we solve in this paper.

#### 3.1 Notation

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  $i$ th spectral component of a transfer function  $\mathbf{G}$ , i.e.,  $\mathbf{G}(z) = \sum_{i=0}^{\infty} \frac{1}{z^i} G(i)$  for  $|z| > 1$ . We use  $\mathcal{F}_T$  to denote the space of finite impulse response (FIR) transfer matrices with horizon  $T$ , i.e.,  $\mathcal{F}_T := \{\mathbf{G} \in \mathcal{RH}_\infty \mid \mathbf{G} = \sum_{i=0}^T \frac{1}{z^i} G(i)\}$ . We frequently use the notation  $\mathbf{G} \in \frac{1}{z} \mathcal{RH}_\infty$  to denote that  $\mathbf{G}$  is strictly proper. Informally this can be parsed as  $z\mathbf{G} \in \mathcal{RH}_\infty$ . Finally, we use  $\mathbf{G}(T_1 : T_2)$  to denote the projection of  $\mathbf{G}$  onto  $\mathcal{F}_{T_1}^\perp \cap \mathcal{F}_{T_2}$ , i.e.,  $\mathbf{G}(T_1 : T_2) = \sum_{i=T_1}^{T_2} \frac{1}{z^i} G(i)$ .

### 3.2 System Model

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

$$x[t+1] = Ax[t] + B_1w[t] + B_2u[t] \quad (3.1a)$$

$$\bar{z}[t] = C_1x[t] + D_{11}w[t] + D_{12}u[t] \quad (3.1b)$$

$$y[t] = C_2x[t] + D_{21}w[t] + D_{22}u[t] \quad (3.1c)$$

where  $x$ ,  $u$ ,  $w$ ,  $y$ ,  $\bar{z}$  are the state vector, control action, external disturbance, measurement, and regulated output, respectively. Equation (3.1) can be written in state space form as

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

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

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

$$\xi[t+1] = A_k\xi[t] + B_ky[t] \quad (3.2a)$$

$$u[t] = C_k\xi[t] + D_ky[t], \quad (3.2b)$$

where  $\xi$  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  $\mathbf{P}$  and the controller  $\mathbf{K}$  is shown in Figure 1.

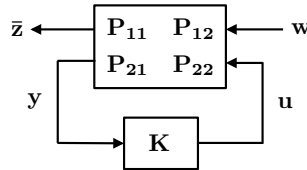


Figure 1: Interconnection of the plant  $\mathbf{P}$  and controller  $\mathbf{K}$ .

The following assumptions are made throughout the paper.

**Assumption 1.** *The interconnection in Figure 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.*

Note that assumption ?? The goal of the optimal control problem is to find a controller  $\mathbf{K}$  to stabilize the plant  $\mathbf{P}$  and minimize a suitably chosen norm<sup>3</sup> of the closed loop transfer matrix from external disturbance  $\mathbf{w}$  to regulated output  $\bar{\mathbf{z}}$ . This leads to the following centralized optimal control formulation:

$$\begin{aligned} & \underset{\mathbf{K}}{\text{minimize}} && \|\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\| \\ & \text{subject to} && \mathbf{K} \text{ internally stabilizes } \mathbf{P}. \end{aligned} \tag{3.3}$$

### 3.3 Youla Parameterization

A common technique to solve the optimal control problem (3.3) is via the Youla parameterization, which is based on a doubly co-prime factorization of the plant, defined as follows.

**Definition 1.** 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 [58].

Let  $\mathbf{Q}$  be the Youla parameter. From [58], problem (3.3) can be reformulated in terms of the Youla parameter as

$$\begin{aligned} & \underset{\mathbf{Q}}{\text{minimize}} && \|\mathbf{T}_{11} + \mathbf{T}_{12}\mathbf{Q}\mathbf{T}_{21}\| \\ & \text{subject to} && \mathbf{Q} \in \mathcal{RH}_\infty \end{aligned} \tag{3.4}$$

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  $\mathbf{T}_{21} = \mathbf{U}_l\mathbf{P}_{21}$ .

The benefit of optimizing over the Youla parameter  $\mathbf{Q}$ , rather than the controller  $\mathbf{K}$ , is that (3.4) is convex with respect to the Youla parameter. One can then incorporate various convex design specifications [3] in (3.4) to customize the controller synthesis task. Once the optimal Youla parameter  $\mathbf{Q}$ , or a suitable approximation thereof, is found in (3.4), we reconstruct the controller by setting  $\mathbf{K} = (\mathbf{Y}_r - \mathbf{U}_r\mathbf{Q})(\mathbf{X}_r - \mathbf{V}_r\mathbf{Q})^{-1}$ .

### 3.4 Structured Controller Synthesis and QI

We now move our discussion to the distributed optimal control problem. We follow the paradigm adopted in [25, 34, 35, 36, 37, 38, 39, 40, 41], 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

---

<sup>3</sup>Typical choices for the norm include  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$ .

between sub-controllers subject to delays imposed by the communication network,<sup>4</sup> which manifest as subspace constraints on the controller itself.<sup>5</sup>

Let  $\mathcal{C}$  be a subspace enforcing the information sharing constraints imposed on the controller  $\mathbf{K}$ . A distributed optimal control problem can then be formulated as [25, 59, 33, 60]:

$$\begin{aligned} & \underset{\mathbf{K}}{\text{minimize}} && \|\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\| \\ & \text{subject to} && \mathbf{K} \text{ internally stabilizes } \mathbf{P}, \mathbf{K} \in \mathcal{C}. \end{aligned} \tag{3.5}$$

A summary of the main results from the distributed optimal control literature [25, 34, 35, 36, 37, 38, 39, 40, 41] can be given as follows: if the subspace  $\mathcal{C}$  is quadratically invariant with respect to  $\mathbf{P}_{22}$  (i.e.,  $\mathbf{K}\mathbf{P}_{22}\mathbf{K} \in \mathcal{C}, \forall \mathbf{K} \in \mathcal{C}$ ) [25], then the set of all stabilizing controllers lying in subspace  $\mathcal{C}$  can be parameterized by those stable transfer matrices  $\mathbf{Q} \in \mathcal{RH}_\infty$  satisfying  $\mathfrak{M}(\mathbf{Q}) \in \mathcal{C}$ , for  $\mathfrak{M}(\mathbf{Q}) := \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1} = (\mathbf{Y}_r - \mathbf{U}_r\mathbf{Q})\mathbf{U}_l$ .<sup>6</sup> Further, these conditions can be viewed as tight, in the sense that quadratic invariance is also a necessary condition [59, 33] for a subspace constraint  $\mathcal{C}$  on the controller  $\mathbf{K}$  to be enforced via a convex constraint on the Youla parameter  $\mathbf{Q}$ .

This allows the optimal control problem (3.5) to be recast as the following convex model matching problem:

$$\begin{aligned} & \underset{\mathbf{Q}}{\text{minimize}} && \|\mathbf{T}_{11} + \mathbf{T}_{12}\mathbf{Q}\mathbf{T}_{21}\| \\ & \text{subject to} && \mathbf{Q} \in \mathcal{RH}_\infty, \mathfrak{M}(\mathbf{Q}) \in \mathcal{C}. \end{aligned} \tag{3.6}$$

### 3.5 QI imposes limitations on controller sparsity

When working with large-scale systems, it is natural to impose that sub-controllers only collect information from a local subset of all other sub-controllers. This can be enforced by setting the subspace constraint  $\mathcal{C}$  in problem (3.5) to encode a suitable sparsity pattern  $\mathbf{K}_{ij} = 0$ ,<sup>7</sup> for some  $i, j$ . However, if the plant  $\mathbf{P}_{22}$  is dense (i.e., if the underlying system is strongly connected), which may occur even if the system matrices  $(A, B_2, C_2)$  are sparse, then *any* such sparsity constraint is not QI condition with respect to the plant  $\mathbf{P}_{22}$ : this follows immediately from the algebraic definition of QI  $\mathbf{K}\mathbf{P}_{22}\mathbf{K} \in \mathcal{C}, \forall \mathbf{K} \in \mathcal{C}$ . As QI is a necessary and sufficient condition for the subspace constraint  $\mathbf{K} \in \mathcal{C}$  to be enforced via a convex constraint on the Youla parameter  $\mathbf{Q}$ , we conclude that for strongly connected systems, any sparsity constraint imposed on the controller  $\mathbf{K}$  can only be enforced via a non-convex constraint on Youla parameter. A major motivation for SLS was to circumvent this limitation of the QI framework.

## 4 State-Feedback System Level Synthesis

In this section, we extend the results of Section 2 to the infinite horizon setting, and propose a novel parameterization of internally stabilizing state-feedback controllers centered around *system responses*, which are defined by the closed loop maps from process disturbances to state and control

<sup>4</sup>Note that this delay may range from 0, modeling instantaneous communication between sub-controllers, to infinite, modeling no communication between sub-controllers.

<sup>5</sup>For continuous time systems, the delays can be encoded via subspaces that may reside within  $\mathcal{RH}_\infty$  as opposed  $\mathcal{RH}_\infty$ .

<sup>6</sup>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  $\mathbf{Q}$ .

<sup>7</sup> $\mathbf{K}_{ij}$  denotes the  $(i, j)$ -entry of the transfer matrix  $\mathbf{K}$ .



action. We show that for a given system, the set of stable closed loop system responses that are achievable by an internally stabilizing LTI controller is an affine subspace of  $\mathcal{RH}_\infty$ , and that the corresponding internally stabilizing controller achieving the desired system response admits a particularly simple and transparent realization.

Consider a state feedback model given by

$$\mathbf{P} = \left[ \begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ I & 0 & 0 \end{array} \right]. \quad (4.1)$$

In this case, the state dynamics described in (3.1) reduce to

$$x[t+1] = Ax[t] + B_2u[t] + B_1w[t]. \quad (4.2)$$

The  $z$ -transform of these state dynamics is then given by

$$(zI - A)\mathbf{x} = B_2\mathbf{u} + \delta_{\mathbf{x}}, \quad (4.3)$$

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

Consider now the (dynamic) state-feedback control policy  $\mathbf{u} = \mathbf{K}\mathbf{x}$  and substitute this into (4.3). After a simple rearrangement the maps  $\Phi_x : \delta_{\mathbf{x}} \rightarrow \mathbf{x}$  and  $\Phi_u : \delta_{\mathbf{x}} \rightarrow \mathbf{u}$  are seen to be

$$\begin{aligned} \Phi_x &= (zI - A - B_2\mathbf{K})^{-1} \\ \Phi_u &= \mathbf{K}(zI - A - B_2\mathbf{K})^{-1}. \end{aligned} \quad (4.4)$$

The resemblance between (2.4) and (4.4) is immediate.

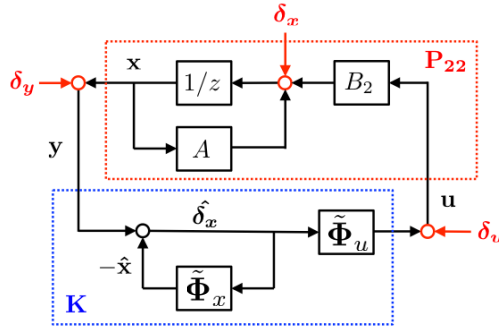


Figure 2: The proposed state feedback controller structure, with  $\tilde{\Phi}_x = I - z\Phi_x$  and  $\tilde{\Phi}_u = z\Phi_u$ .

We now provide an infinite horizon analogue to Theorem 2.1, that ensures that the system response  $\{\Phi_x, \Phi_u\}$  are achievable by the proposed state-feedback policy  $\mathbf{u} = \mathbf{K}\mathbf{x}$ . As we are now operating over an infinite horizon, we must take care to provide an internally stable realization for the controller.

**Theorem 4.1.** *Consider the LTI system (4.2), evolving under a dynamic state-feedback control policy  $\mathbf{u} = \mathbf{K}\mathbf{x}$ . The following statements are true:*

1. The affine subspace defined by

$$[zI - A \quad -B_2] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I, \quad \Phi_x, \Phi_u \in \frac{1}{z}\mathcal{RH}_\infty \quad (4.5)$$

parameterizes all system responses from  $\delta_x$  to  $(\mathbf{x}, \mathbf{u})$  as defined in (4.4), achievable by an internally stabilizing state feedback controller  $\mathbf{K}$ .

2. For any transfer matrices  $\{\Phi_x, \Phi_u\}$  satisfying (4.5), the controller  $\mathbf{K} = \Phi_u \Phi_x^{-1}$ , implemented as shown in Figure 2 via the following equations:

$$\begin{aligned} \mathbf{u} &= z\Phi_u \hat{\delta}_x \\ \hat{\delta}_x &= \mathbf{x} + (I - z\Phi_x)\hat{\delta}_x, \end{aligned}$$

is internally stabilizing and achieves the desired system response (4.4).

The similarities between the finite horizon case given in Theorem 2.1 and those presented in Theorem 4.1 are striking – this should not be surprising, as the lower-block-triangular operators we previously considered can be viewed as truncations of the matrix representations of the transfer matrices  $\{\Phi_x, \Phi_u\}$ .

At this point the observant reader may have noticed that Theorem 4.1 circumvents the standard assumptions of *stabilizability* of the pair  $(A, B_2)$ . However this not the case:

**Lemma 4.2.** *The constraint (4.5) is feasible if and only if  $(A, B_2)$  is stabilizable.*

**Proof.** We first show that the stabilizability of  $(A, B_2)$  implies that there exist transfer matrices  $\Phi_x, \Phi_u \in \frac{1}{z}\mathcal{RH}_\infty$  satisfying equation (4.5). 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 (4.3), we have  $\mathbf{x} = (zI - A - B_2F)^{-1}\delta_x$  and  $\mathbf{u} = F(zI - A - B_2F)^{-1}\delta_x$ . The system response is given by  $\Phi_x = (zI - A - B_2F)^{-1}$  and  $\Phi_u = F(zI - A - B_2F)^{-1}$ , which lie in  $\frac{1}{z}\mathcal{RH}_\infty$  and are a solution to (4.5).

For the opposite direction, we note that  $\Phi_x, \Phi_u \in \mathcal{RH}_\infty$  implies that these transfer matrices do not have poles outside the unit circle  $|z| \geq 1$ . From (4.3), we further observe that  $[zI - A \quad -B_2]$  is right invertible in the region where  $\Phi_x$  and  $\Phi_u$  do not have poles, with  $[\Phi_x^\top \quad \Phi_u^\top]^\top$  being its right inverse. This then implies that  $[zI - A \quad -B_2]$  has full row rank for all  $|z| \geq 1$ . This is equivalent to the PBH test [13] for stabilizability, proving the claim. ■

This lemma therefore provides a sanity check: if the system is not stabilizable, then there do not exist any stable system responses, and hence the constraints are infeasible. We now prove Theorem 4.1.

**Proof.**[Theorem 4.1] This proof is nearly identical to that of Theorem 2.1, but with the added complication of ensuring internal stability of the resulting controller.

Proof of 1.: Let  $\mathbf{K}$  be any internally stabilizing controller, and set  $\mathbf{u} = \mathbf{K}\mathbf{x}$ . Then as argued above, we have that

$$\begin{aligned} \mathbf{x} &= (zI - (A + B_2\mathbf{K}))^{-1}\delta_x \\ \mathbf{u} &= \mathbf{K}(zI - (A + B_2\mathbf{K}))^{-1}\delta_x. \end{aligned} \quad (4.6)$$

It is then easily seen that

$$[zI - A \quad -B_2] \begin{bmatrix} (zI - (A + B_2\mathbf{K}))^{-1} \\ \mathbf{K}(zI - (A + B_2\mathbf{K}))^{-1} \end{bmatrix} = (zI - A - B_2\mathbf{K})(zI - (A + B_2\mathbf{K}))^{-1} = I. \quad (4.7)$$

Proof of 2.: First notice that the affine constraint (4.5) implies that  $\Phi_x(1) = I$ , and hence  $\Phi_x^{-1}$  exists. We begin by showing that the controller  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  achieves the desired response, and then show that the implementation suggested in Figure 2 is internally stabilizing. Let  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  and notice that

$$\mathbf{x} = (zI - (A + B_2 \Phi_u \Phi_x^{-1}))^{-1} \delta_x. \quad (4.8)$$

But we then have that

$$(zI - (A + B_2 \Phi_u \Phi_x^{-1}))^{-1} = ((zI - A) \Phi_x - B_2 \Phi_u) \Phi_x^{-1} = \Phi_x ((zI - A) \Phi_x - B_2 \Phi_u)^{-1} = \Phi_x \quad (4.9)$$

where the last equality follows from the fact that  $\{\Phi_x, \Phi_u\}$  satisfy (4.5). Similarly we have that

$$\mathbf{u} = \Phi_u \Phi_x^{-1} \mathbf{x} = \Phi_u \Phi_x^{-1} \Phi_x \delta_x = \Phi_u \delta_x, \quad (4.10)$$

where the second equality follows from the fact that  $\mathbf{x} = \Phi_x \delta_x$ .

Finally, we turn to proving the internal stability of the proposed controller realization shown in Figure 2, where here  $\tilde{\Phi}_x = I - z\Phi_x$  and  $\tilde{\Phi}_u = z\Phi_u$ . We first note that from Figure 2, we can express the state feedback controller  $\mathbf{K}$  as  $\mathbf{K} = \tilde{\Phi}_u (I - \tilde{\Phi}_x)^{-1} = (z\Phi_u)(z\Phi_x)^{-1} = \Phi_u \Phi_x^{-1}$ .

It can be checked that  $z\tilde{\Phi}_x, \tilde{\Phi}_u \in \mathcal{RH}_\infty$ , and hence the internal feedback loop between  $\hat{\delta}_x$  and the reference state trajectory  $\hat{\mathbf{x}}$  is well defined. As is standard, we introduce external perturbations  $\delta_x, \delta_y$ , and  $\delta_u$  into the system and note that the perturbations entering other links of the block diagram can be expressed as a combination of  $(\delta_x, \delta_y, \delta_u)$  being acted upon by some stable transfer matrices.<sup>8</sup> Hence the standard definition of internal stability applies, and we can use a bounded-input bounded-output argument (e.g., Lemma 5.3 in [58]) to conclude that it suffices to check the stability of the nine closed loop transfer matrices from perturbations  $(\delta_x, \delta_y, \delta_u)$  to the internal variables  $(\mathbf{x}, \mathbf{u}, \hat{\delta}_x)$  to determine the internal stability of the structure as a whole.

Routine calculations show that the closed loop transfer matrices from  $(\delta_x, \delta_y, \delta_u)$  to  $(\mathbf{x}, \mathbf{u}, \hat{\delta}_x)$  are given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \hat{\delta}_x \end{bmatrix} = \begin{bmatrix} \Phi_x & -\tilde{\Phi}_x - \Phi_x A & \Phi_x B_2 \\ \Phi_u & \tilde{\Phi}_u - \Phi_u A & I + \Phi_u B_2 \\ \frac{1}{z} I & I - \frac{1}{z} A & \frac{1}{z} B_2 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_u \end{bmatrix}. \quad (4.11)$$

As all nine transfer matrices in (4.11) are stable, the implementation in Figure 2 is internally stable. Furthermore, the desired system response  $\{\Phi_x, \Phi_u\}$ , from  $\delta_x$  to  $(\mathbf{x}, \mathbf{u})$ , is achieved. ■

With Theorem 4.1 in hand we can now proceed as in the finite horizon case and define the System Level Synthesis problem. Let us assume that a suitable cost functional  $g$  has been chosen – suitable examples include  $\mathcal{L}_1, \mathcal{H}_\infty$ , or  $\mathcal{H}_2$  objective functions [58]. The general form of an SLS problem takes the form

$$\begin{aligned} & \min_{\Phi_x, \Phi_u} g(\Phi_x, \Phi_u) \\ & \text{subject to} \quad [zI - A \quad -B_2] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \\ & \quad \quad \quad \Phi_x, \Phi_u \in \frac{1}{z} \mathcal{RH}_\infty \cap \mathcal{S}. \end{aligned} \quad (4.12)$$

If the objective function  $g(\cdot)$  and constraint set  $\mathcal{S}$  are convex, then optimization problem (4.12) is a convex but infinite-dimensional problem. The constraint set  $\mathcal{S}$  can be used to capture additional constraints to be imposed on the the system response, which in turn translate to constraints

<sup>8</sup>The matrix  $A$  may define an unstable system, but viewed as an element of  $\mathcal{F}_0$ , defines a stable (FIR) transfer matrix.

on the internal realization of the controller as illustrated in Figure 2. We note here that if the system is controllable, then a useful SLC to add is that the system responses are FIR, i.e., that  $\{\Phi_x, \Phi_u\} \in \mathcal{F}_T$  for some suitably chosen horizon  $T$ . In that case, the optimization problem (4.12) immediately becomes finite dimensional (assuming that the remaining constraints admit finite dimensional representations), and further, the controller implementation shown in Figure 2 reduces to an interconnection of FIR filter banks, greatly simplifying controller deployment. Finally, we show in Section 4.5 that, modulo some minor technical assumptions and slight modifications to problem (4.12), taking a sufficiently long FIR horizon  $T$  results in near optimal performance.

#### 4.1 Distributed Control, Locality, and Scalability

The key insight that we exploit in this section is that distributed constraints (in the form of sparsity and delay constraints) can be imposed on the internal blocks of the controller realization shown in Figure 2 via subspace constraints on the system responses. This idea of imposing structure on the internal controller realization (i.e., on  $\{\Phi_x, \Phi_u\}$ ), instead of on the controller itself (i.e., on the map  $\mathbf{K} = \Phi_u \Phi_x^{-1}$ ), is what allows us to circumvent the limitations of existing results in the distributed control literature. This insight, coupled with the observation that many optimal control problems of interest can be decomposed into parallel subproblems, allows us to scale out distributed control methods to the large-scale setting with ease.

Before delving into the technical details, we present a simple example that illustrates some of the shortcomings in the distributed controls literature that we are able to overcome.

**Example 1.** Consider the optimal control problem:

$$\begin{aligned} & \text{minimize}_u \quad \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \mathbb{E} \|x[t]\|_2^2 \\ & \text{subject to} \quad x[t+1] = Ax[t] + u[t] + w[t], \end{aligned} \tag{4.13}$$

with disturbance  $w[t] \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$ . 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  $u^*$  for this LQR problem is easily seen to be given by  $u^*[t] = -Ax[t]$ .

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  $i$ th node with the corresponding state/control pair  $(x_i, u_i)$ . In this case, we have that the optimal control policy  $u^*$  can be implemented in a localized manner. In particular, in order to implement the state feedback policy for the  $i$ th 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 below, 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 (4.13) by converting it to its equivalent  $\mathcal{H}_2$  model matching problem (3.5) 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 any sparsity constraint in the form of  $\mathbf{K}_{ij} = 0$  is not QI with respect to the plant  $\mathbf{P}_{22} = (zI - A)^{-1}$ . To see this, note that if the graph  $\mathcal{G}$  is strongly connected, then  $\mathbf{P}_{22}$  is a dense transfer function: it then follows immediately that any subspace  $\mathcal{C}$  enforcing sparsity constraints on  $\mathbf{K}$  fails to satisfy  $\mathbf{K}\mathbf{P}_{22}\mathbf{K} \in \mathcal{C}$ ,  $\forall \mathbf{K} \in \mathcal{C}$ , and hence is not QI with respect to  $\mathbf{P}_{22}$ . The results of [33] further allow us to

conclude that computing such a structured controller can never be done using convex programming when using the Youla parameterization.

In contrast, in the case of a full control ( $B_2 = I$ ) problem, the condition (4.5) simplifies to  $(zI - A)\Phi_x - \Phi_u = I$ ,  $\Phi_x, \Phi_u \in \frac{1}{z}\mathcal{RH}_\infty$ . 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  $\Phi_x$  and  $\Phi_u$  to be a subset of that of  $A$ . It can be checked that  $\Phi_x = \frac{1}{z}I$ , and  $\Phi_u = -\frac{1}{z}A$  satisfy the above constraints, and recover the globally optimal controller  $\mathbf{K} = -A$ .

## 4.2 Spatial and Temporal Locality

As mentioned above, the constraint set  $\mathcal{S}$  can be used to impose distributed constraints on the controller realization – in this section we describe a useful such class of constraints that enforce *locality* in the system responses. In particular, in addition to the FIR constraints described above, which can be interpreted as localizing the system responses in time, we also ask that the system response be spatially localized.

The notion of temporal locality ensures that the closed-loop disturbance response has a finite-impulse response. Furthermore, it ensures that the resulting System Level Synthesis problem is a finite-dimensional mathematical program. Spatial locality ensures that when a disturbance enters the system, its effect is felt only in a local neighborhood of the origin. Spatiotemporal locality, as well as being desirable from a performance perspective, will also make computation tractable. We begin by introducing spatial locality.

The network structure of the system is naturally modeled by a graph  $\mathcal{G}(V, E)$ . In this setting the vertices  $v_i \in V$  denote subsystems and the unweighted edges,  $e_{ij} := (v_i, v_j) \in E \subset V \times V$  with  $v_i, v_j \in V$ , indicate that subsystem  $j$  affects subsystem  $i$ . As a distance metric on  $\mathcal{G}$  we use the length of the shortest path from  $v_i$  to  $v_j$  which we denote by  $\mathbf{dist}(v_i \rightarrow v_j)$ .

**Definition 1.** For the unweighted graph  $\mathcal{G}$ , the  $d$ -outgoing set of subsystem  $i$  is  $\mathbf{out}_{v_i}(d) := \{v_j \mid \mathbf{dist}(v_i \rightarrow v_j) \leq d\}$ . Analogously, the  $d$ -incoming set of subsystem  $i$  is  $\mathbf{in}_{v_i} := \{v_j \mid \mathbf{dist}(v_i \rightarrow v_j) \leq d\}$ .

The definition of locality we use is built directly on the notion of the incoming and outgoing sets. Informally, the system is  $d$ -localizable if every subsystem in the network can synthesize a controller using a plant model which is limited to subsystems contained in its  $d$ -outgoing set and implemented using signals coming only from subsystems in its  $d$ -incoming set. Note that we choose  $d$  to be significantly smaller than the longest path in  $\mathcal{G}$ .

**Definition 2.** Given the state dynamics (4.2) and the map  $x = \Phi_x \mathbf{w}$ . Denote the transfer matrix describing the perturbation from signal  $w_j$  at subsystem  $j$  to the state of subsystem  $i$ ,  $x_i$ , by  $[\Phi_x]_{ij}$ . Then the map  $\Phi_x$  is  $d$ -localizable if and only if for every subsystem  $j$ ,  $[\Phi_x]_{ij} = 0$  for all  $i \notin \mathbf{out}_j(d)$ . The definition for  $d$ -localizability of  $\Phi_u$  where  $u = \Phi_u \mathbf{w}$ , follows in the same manner but with perturbations to control action  $u_i$  at subsystem  $i$ .

**Definition 3.** The subspace  $\mathcal{L}_d$  is said to be a  $d$ -localized constraint if it constrains  $\Phi_x$  to be  $d$ -localized and  $\Phi_u$  to be  $(d + 1)$ -localized.

We now introduce temporal-locality: The set  $\mathcal{F}_T$  is a convex *System Level Constraint* (SLC) that ensures the system response  $\{\Phi_x, \Phi_u\}$  has a finite impulse response of horizon length  $T$ .

Consider a transfer matrix  $\mathbf{G}$ , then:

$$\mathcal{F}_T := \left\{ \mathbf{G} \in \mathcal{RH}_\infty \mid \mathbf{G} = \sum_{i=0}^T \frac{1}{z^i} \mathbf{G}[i] \right\},$$

We make an important observation regarding computation here. When the SLC set  $\mathcal{S}$  intersects  $\mathcal{F}_T$ , the resulting SLS problem (4.12) becomes finite-dimensional. We now have all the ingredients necessary to define a broad class of locality constraints.

**Definition 4.** *The system given by the state equations (4.2) is  $(d, T)$ -localizable if (4.12) is feasible with  $\mathcal{S} = \mathcal{L}_d \cap \mathcal{F}_T$  for some  $(d, T) \in \mathbf{Z}_+ \times \mathbf{Z}_+$ .*

The final class we consider describe delay constraints. We allow for three types of delays in the network; actuation delay, communication delay, and sensing delay denoted by  $k_a, k_c$ , and  $k_s$  respectively. Actuation delay describes the time taken for control action  $u_i$  to affect state  $x_i$  of its subsystem. Communication delay is the amount of time it takes for a subsystem-controller to transmit a signal to its immediate neighbors. Finally, the sensing delay is the time taken for a subsystem-controller  $u_i$  to access its state variable  $x_i$ . For brevity it is assumed that  $(k_a, k_c, k_s)$  are integers and normalized with respect to the sampling period of the system.

We are now in position where we can combine the  $(d, T)$ -localization subspace constraint and the communication subspace constraints as constraints on the system response  $\{\Phi_x, \Phi_u\}$ . The most transparent way to implement these constraints is to enforce sparsity constraints on the entries of the system response spectral factors. Recall that  $\Phi_x = \sum_{k=0}^{\infty} z^{-k} \Phi_x[k]$  and  $\Phi_u = \sum_{k=0}^{\infty} z^{-k} \Phi_u[k]$ , then the combined  $(d, T)$ -localization and communication subspace constraints can be encoded by specifying the support of the spectral factors. Define  $\alpha = \lfloor \frac{k - k_a - k_s}{k_c} \rfloor$ , then we have

$$\begin{aligned} \text{supp}(\Phi_x[k]) &\subseteq \text{supp}((A + I)^{\min(d, \alpha)}) \\ &\text{for } k = 1 \dots T \\ \text{supp}(\Phi_u[k]) &\subseteq \text{supp}((A + I)^{\min(d+1, \alpha)}) \end{aligned}$$

and  $\Phi_x[k] = \mathbf{0}$ ,  $\Phi_u[k] = \mathbf{0}$  for all  $k > T$ . The reader is referred to [61] for a detailed description of the derivation of  $\alpha$  and for some explicit cases where combinations of  $d$  and the delay parameters are guaranteed to lead to feasible solutions. The delay constraints which are encoded as sparsity constraints on the spectral factors are convex constraints. For notational convenience we will use the set  $\mathcal{D}$  to denote these sparsity constraints. Thus a  $(d, T)$ -locality constraint with delays is given by  $\mathcal{S} = \mathcal{L}_d \cap \mathcal{F}_T \cap \mathcal{D}$ .

### 4.3 Scalability via Decomposition

We now introduce the concept of column-separability for cost functionals and constraints. It is this column-wise separability that will allow us to achieve  $O(1)$  synthesis complexity with respect to the state dimension of (4.2).

For compactness of notation, in this section we define the transfer matrix  $\Phi = [\Phi_x^T \ \Phi_u^T]^T$  and assume it dimension  $n_r \times n_c$ . Let  $\{c_1, \dots, c_p\}$  partition the set  $\{1, \dots, n_c\}$ . Let us partition  $\Phi$  as  $\{\Phi(:, c_1) \dots, \Phi(:, c_p)\}$ . This idea of partitioning a decision variable by its columns leads to the notion of column-wise separability:

**Definition 5.** The functional  $g(\Phi)$  is column-wise separable with respect to the partition  $c_1, \dots, c_p$  if it can be written as

$$g(\Phi) = \sum_{j=1}^p g_j(\Phi(:, c_j))$$

for some functionals  $g_j$  for  $j = 1, \dots, p$ .

Some examples of column-wise separable norms are described below. We note that the  $\mathcal{H}_\infty$ -norm is *not* column- (nor row-) wise separable.

**Example 2** (Frobenius norm). The square of the Frobenius norm of a  $m \times n$  matrix  $\Phi$  is given by

$$\|\Phi\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n \Phi_{ij}^2.$$

This objective function is column-wise separable with respect to arbitrary column-wise partition. Consider the explicit example of

$$\Phi = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

partitioned as  $\{1, 3\}, \{2\}$ , then

$$\|\Phi\|_F^2 = \left\| \begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix} \right\|_F^2 + \left\| \begin{bmatrix} 2 \\ 5 \end{bmatrix} \right\|_F^2.$$

**Example 3** ( $\mathcal{H}_2$  norm). The square of the  $\mathcal{H}_2$  norm of a transfer matrix  $\Phi$  is given by

$$\|\Phi\|_{\mathcal{H}_2}^2 = \sum_{t=0}^{\infty} \|\Phi[t]\|_F^2, \quad (4.14)$$

which is column-wise separable with respect to arbitrary column-wise partition. Now consider the  $\mathcal{H}_2$ -norm of a transfer matrix realization of a system with a finite impulse response of horizon  $T$ , i.e.  $\Phi \in \mathcal{F}_T$ . In this case the summation in (4.14) is finite.

**Example 4** (Element-wise  $\ell_1$ ). Motivated by the separability of the  $\mathcal{H}_2$  norm, we define the element-wise  $\ell_1$  norm (denoted by  $\mathcal{E}_1$ ) of a transfer matrix  $\Phi \in \mathcal{F}_T$  as

$$\|\Phi\|_{\mathcal{E}_1} = \sum_i \sum_j \sum_{t=0}^T |\Phi_{ij}[t]|.$$

**Definition 6.** The constraint-set  $\mathcal{S}$  in (4.12) is column-wise separable with respect to the partition  $c_1, \dots, c_p$  when the condition

$$\Phi \in \mathcal{S} \Leftrightarrow \Phi(:, c_j) \in \mathcal{S}_j \text{ for } j = 1, \dots, p$$

is satisfied for some sets  $\mathcal{S}_j$  for  $j = 1, \dots, p$ , where  $\mathcal{S}_j = \mathcal{L}(:, c_j) \cap \mathcal{X}_j \cap \mathcal{F}_T$ .

**Example 5** (Affine Subspace). *The affine subspace constraint*

$$\mathbf{G}\Phi = \mathbf{H}$$

is column-wise separable with respect to arbitrary column-wise partition. Specifically, we have

$$\mathbf{G}\Phi(:, c_j) = \mathbf{H}(:, c_j)$$

for  $c_j$  any subset of  $\{1, \dots, n\}$ . An important special case of column-wise separable affine subspace constraints is the system level parametrization constraint (4.12), here we have

$$\mathbf{G} = [zI - A \quad -B_2], \quad \Phi = \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix}, \text{ and } \mathbf{H} = I.$$

**Example 6** (Locality and FIR Constraints). *The locality and FIR constraints introduced in Section 4.2*

$$\Phi \in \mathcal{L} \cap \mathcal{F}_T$$

are column-wise separable with respect to arbitrary column-wise partition. This follows from the fact that both locality and FIR constraints can be encoded via sparsity structure: the resulting linear subspace constraint is trivially column-wise separable.

With the definition of column-wise separability firmly established the following chain of problems will make clear how functions and constraint sets that satisfy these definitions provide computational tractability to large-scale problems. First, consider the generic optimization problem

$$\underset{\Phi}{\text{minimize}} \quad g(\Phi) \tag{4.15a}$$

$$\text{subject to} \quad \Phi \in \mathcal{S}, \tag{4.15b}$$

where the decision variable is  $\Phi$ , an  $m \times n$  transfer matrix,  $g(\cdot)$  a functional objective, and  $\mathcal{S}$  a set constraint. Assume now that the objective function (4.15a) and the set constraint (4.15b) are both column-wise separable with respect to a column-wise partition  $\{c_1, \dots, c_p\}$ . When the objective function and constraints are both column-wise separable we are said to have a *column-wise separable problem*. Specifically, (4.15) can be partitioned into  $p$  parallel subproblems as

$$\underset{\Phi(:, c_j)}{\text{minimize}} \quad g_j(\Phi(:, c_j)) \tag{4.16a}$$

$$\text{subject to} \quad \Phi(:, c_j) \in \mathcal{S}_j \tag{4.16b}$$

for  $j = 1, \dots, p$ . We will now show that System Level Synthesis problems of the form (4.12) under separability conditions can be written in the form of (4.16) and explicitly form the problem. First, let us define

$$\mathbf{Z}_{AB} = [zI - A \quad -B_2],$$

set  $\Phi$  as defined in example 5, and let the the system level constraint be given by  $\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}$ . The state-feedback SLS problem (4.12) can then be written as

$$\min_{\Phi} \quad g(\Phi) \tag{4.17a}$$

$$\text{subject to} \quad \mathbf{Z}_{AB} = I \tag{4.17b}$$

$$\Phi \in \mathcal{S}. \tag{4.17c}$$



As we have already shown, the affine subspace constraint (4.17b) is column-wise separable with respect to any column partitioning. If  $g(\cdot)$  is chosen to be column-wise separable, for example it is given by the  $\mathcal{H}_2$ - or  $\mathcal{E}_1$ -norms, then (4.17) is a column-wise separable optimization problem if the SLC (4.17c) is column-wise separable. Given that we have shown  $\mathcal{F}_T \cap \mathcal{L}$  to be separable, all that is needed is for  $\mathcal{S}$  to be column-wise separable is for  $\mathcal{X}$  to also be separable. When this is true, we can express the set constraint  $\mathcal{S}_j$  in (4.16b) as  $\mathcal{S}_j = \mathcal{L}(:, c_j) \cap \mathcal{F}_T \cap \mathcal{X}_j$  for some  $\mathcal{X}_j$  for each  $j$ . Finally, using the column partitions, (4.17) can be partitioned into  $p$  parallel subproblems of the form

$$\underset{\Phi(:, c_j)}{\text{minimize}} \quad g_j(\Phi(:, c_j)) \quad (4.18a)$$

$$\text{subject to} \quad \mathbf{Z}_{AB} \Phi(:, c_j) = I(:, c_j) \quad (4.18b)$$

$$\Phi(:, c_j) \in \mathcal{L}(:, c_j) \cap \mathcal{F}_T \cap \mathcal{X}_j \quad (4.18c)$$

for  $j = 1, \dots, p$ . The column-wise separable SLS problem (4.18) is exactly in the form of the generic column-wise separable optimization problem (4.16).

Observe that the  $p$  sub-problems are completely decoupled and can be solved in parallel, moreover, because of the constraint  $\mathcal{L}$  the dimension of each the sub-problem will be smaller than that of the global problem (4.12) – this is because the decomposed constraint  $\mathcal{L}(:, c_j)$  has sparse support and can be permuted into partition of zero and non-zero elements, the zero elements can then be dropped. Effectively, this means that each sub-problem only has to work with a local system model. Subsystem  $i$  comprising  $(A_{ii}, A_{ij}, B_{ii})$  is a constructed from suitably defined sub-blocks of  $(A, B_2)$  as defined in (4.2).

The following simplified example should provide some additional intuition into the role of spatio-temporal locality.

**Example 7** (Symmetric Chain). *Consider the network dynamical system with the dynamics at node  $i$  given by*

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

*The set  $\mathcal{N}_i$  denotes the incoming set of node  $i$ , in this case the system graph is a bi-directional chain, i.e. the adjacency matrix has zeros everywhere apart from at the sub- and super-diagonal entries. Furthermore, we assume that the control inputs and disturbances only directly affect local subsystems. In Figure 3 a diagram of the system response to a particular disturbance  $(\delta_{\mathbf{x}})_i$  is depicted. The horizontal-axis denotes time and the vertical-axis indicates position in the chain. At time  $t$  an impulse hits node  $i$  (chosen as the center of the chain). The figure shows 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  $(\delta_{\mathbf{x}})_i$  imposed by the localized SLC.*

### 4.3.1 Dimension Reduction

In the previous section, the idea of column-wise separability for functions and constraints was introduced. It was shown that SLS problems (4.12) defined by column-wise separable components could be easily solved by a set of parallel sub-problems as described by (4.18). In this subsection,

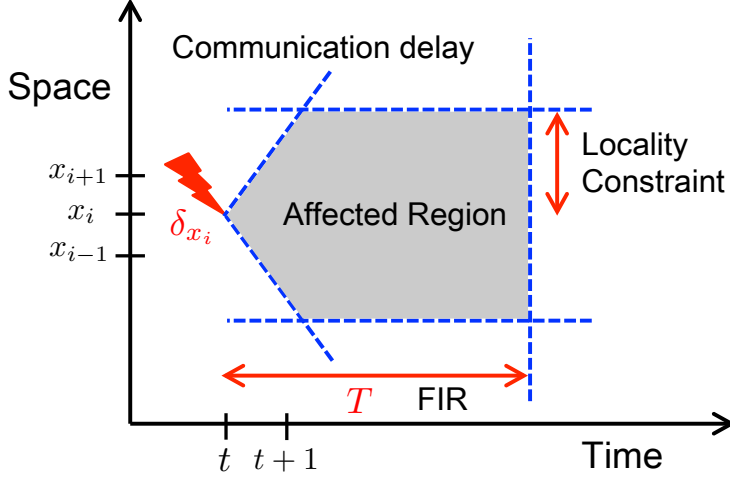


Figure 3: The state response to a disturbance hitting the chain at time  $t$ .

we take advantage of the fact that locality constraints  $\mathcal{L}(:, c_j)$  and network structure likely enforce a sparsity pattern on the data and decision variables. Such sparsity leads to many of the entries in the columns of  $\Phi(:, c_j)$  being constrained to be zero, thus many of the elements of  $\mathbf{Z}_{AB}$  are not needed - we now proceed to demonstrate how this can be capitalized on.

Let  $s_j$  and  $t_j$  be sets of positive integers, and assume that we have a column-wise separable SLS problem (4.18) with column partitions  $\{c_1, \dots, c_p\}$ . Later in the section we will describe how to construct the sets  $s_j$  and  $t_j$ , for now we focus on the intuition behind the algorithm. It can be taken for granted that the cardinality of the set  $s_j$  is less than the number of rows in  $\Phi(:, c_j)$ . Likewise, the cardinality of  $t_j$  will be less than the number of rows in  $\mathbf{Z}_{AB}$ . Consider the  $j^{\text{th}}$  subproblem from (4.18), it will be shown that it is equivalent to the following optimization problem:

$$\underset{\Phi(s_j, c_j)}{\text{minimize}} \quad \bar{g}_j(\Phi(s_j, c_j)) \quad (4.19a)$$

$$\text{subject to} \quad \mathbf{Z}_{AB}(t_j, s_j)\Phi(s_j, c_j) = I(t_j, c_j) \quad (4.19b)$$

$$\Phi(s_j, c_j) \in \mathcal{L}(s_j, c_j) \cap \mathcal{F}_T \cap \bar{\mathcal{X}}_j, \quad (4.19c)$$

where  $\Phi(s_j, c_j)$  corresponds to a sub-matrix of  $\Phi$  composed of the rows and columns of  $\Phi$  specified by the sets  $s_j$  and  $c_j$  respectively. The cost functional  $\bar{g}_j$  is simply the functional  $g_j$  restricted to the reduced set of variables, likewise  $\bar{\mathcal{X}}_j$  is simply  $\mathcal{X}_j$  restricted to the appropriate subset. 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 computational complexity of solving (4.19) depends on the choice of cost functional  $g$  (and hence  $\bar{g}_j$ ) and the number of variables in the problem. It should be clear by looking at (4.19) that the cardinality of the sets  $c_j, s_j, t_j$  determine the computational complexity of the resulting reduced-order SLP. For example, the cardinality of the set  $s_j$  is equal to the number of nonzero rows of the locality constraint  $\mathcal{L}(:, c_j)$ . The sparsity of the system matrices is also central to the cardinality of these sets. When the locality 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 (4.18) reduces to a local optimization subproblem (4.19) which depends only on the local plant model  $\mathbf{Z}_{AB}(t_j, s_j)$ .

Combining column-wise separability with the dimension-reduction procedure described above, we have the following algorithm:

**Input:**  $g(\cdot)$ ,  $\mathcal{S}$ , and  $\mathbf{Z}_{AB}$

1. Find the column-wise partitioning  $\{c_1, \dots, c_p\}$  such that (4.17) decomposes into (4.18).
2. **For**  $j = 1, \dots, p$   
 Construct the sets  $s_j, t_j$  such that (4.18) reduces to (4.19).  
 Solve (4.19).

**Output:**  $\Phi$

**End For**

3. From sub-solutions  $\Phi(s_j, c_j)$  with  $j = 1, \dots, p$ , construct  $\Phi$ .

The above algorithm embodies the idea of locality. Not only does locality allow for decoupling of the system level synthesis problem, it also provides a natural set of reduced order models where each variable in the smaller problem corresponds to a “local” patch of the system model. The reader is referred to [55] for the details of how to construct minimal sets  $s_j$  and  $t_j$ .

#### 4.4 Localized LQR Optimal Control

Here we provide an example of a classical distributed control problem, namely that of distributed  $\mathcal{H}_2$ , or LQR, optimal control, that can be made scalable by imposing locality constraints on the system responses  $\{\Phi_x, \Phi_u\}$ , and exploiting the separability of the resulting optimization problem. We call the resulting optimization problem the Localized LQR (LLQR) problem.

In [44], we formulate the LLQR problem with uncorrelated process noise (i.e.,  $B_1 = I$ ) as

$$\underset{\{\Phi_x, \Phi_u\}}{\text{minimize}} \quad \|[C_1 \quad D_{12}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix}\|_{\mathcal{H}_2}^2 \quad (4.20a)$$

$$\text{subject to} \quad [zI - A \quad -B_2] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \quad (4.20b)$$

$$\begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \in \mathcal{L} \cap \mathcal{F}_T \cap \frac{1}{z} \mathcal{RH}_\infty. \quad (4.20c)$$

The (system level objective) SLO (4.20a) and the SLC (4.20c) are both column-wise separable with respect to arbitrary column-wise partitions. The separable property of the SLO is implied by the separability of the  $\mathcal{H}_2$  norm and the assumption that the process noise is pairwise uncorrelated. The separability of the constraints (4.20b) - (4.20c) follows from the above discussion pertaining to affine subspaces. The physical interpretation of the column-wise separable property is that we can analyze the system response of each local process disturbance  $\delta_{x_j}$  in an independent and parallel way, and then exploit the superposition principle satisfied by LTI systems to reconstruct the full solution to the LLQR problem. We also note that removing the locality and FIR constraints does

not affect column-wise separability, and hence the standard LQR problem with uncorrelated noise is also column-wise separable.

In general, imposing FIR or locality constraints degrades the transient performance of the controller, so our controller is not  $\mathcal{H}_2$  optimal. However, we will see in the next section via a numerical example that by choosing appropriate  $(d, T)$  parameters, the difference becomes negligible. In Section 4.5.3 we will derive bounds on the gap between the optimal solution and an FIR approximation.

#### 4.4.1 Performance Comparison

In this section, we synthesize our LLQR optimal controller for a specific plant, and compare the performance with different classes of  $\mathcal{H}_2$  optimal controllers from [36]. In particular, we consider the centralized, delayed centralized, and optimal distributed (with quadratically invariant (QI) information sharing constraints)  $\mathcal{H}_2$  controllers.

The plant model is given by  $(A, B)$ , where  $A$  is a tridiagonal matrix with dimension 59 given by

$$A = \begin{bmatrix} 1 & 0.2 & \cdots & 0 \\ -0.2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0.2 \\ 0 & \cdots & -0.2 & 1 \end{bmatrix}.$$

The instability of the plant is quantified by the spectral radius of  $A$ , which is  $\rho(A) = 1.0768 > 1$ .  $B$  is a  $59 \times 20$  matrix with the  $(6n + 1, 2n + 1)$ -th and  $(6n + 2, 2n + 2)$ -th entries being 1 and zero elsewhere,  $n = 0, \dots, 9$ . Even though our method can be applied to an arbitrary plant topology, we use this simple plant model as it leads to easily visualized representations of the aforementioned space-time regions.

For the localized controller, we choose  $(d, T) = (9, 29)$ . For all controllers with communication delay constraints, we assume that the communication network has the same topology as the physical network, but the speed is  $t_c = 1.5$  times faster than the speed at which dynamics propagate through the plant. For our localized distributed controllers, we have

$$\begin{aligned} \text{supp}(\Phi_x[t]) &= \bigcup_{t=1}^T \frac{1}{z^t} \text{supp}\left(A^{\min(d, \lfloor t_c(t-1) \rfloor)}\right) \\ \text{supp}(\Phi_u[t]) &= \bigcup_{t=2}^T \frac{1}{z^t} \text{supp}\left(B^\top\right) \text{supp}\left(A^{\min(d+1, \lfloor t_c(t-2) \rfloor)}\right) \end{aligned} \quad (4.21)$$

for each value of  $t$  up to  $T$ . The union operation for binary matrices is taken as the element-wise OR operation. We then solve (4.20) with  $C_1$  and  $D_{12}$  set to identity.

We illustrate the difference between the different control schemes by plotting the space-time evolution of a single disturbance hitting the middle state. A concrete realization of a localized forward space-time region is in Figures 4(h) and 4(i): the effect of the disturbance is limited in both state and control action in time and space.

Next, we calculate the optimal value for each controller and summarize the results in Table 3. The objective is normalized with respect to that of the centralized  $\mathcal{H}_2$  controller. Clearly, our

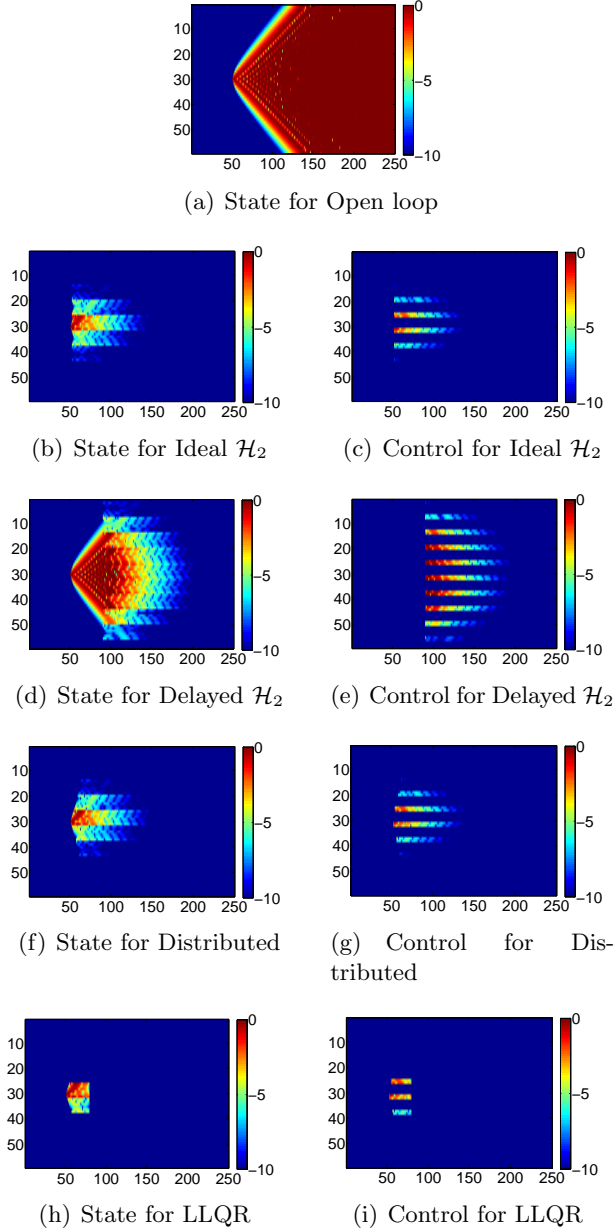


Figure 4: The log absolute value of state and control for a given disturbance at time  $t = 50$ . The horizontal axis represents time and the vertical axis represents state in space. The legend on the right shows the meaning of the colors.

localized control scheme can achieve similar performance to that of the centralized one. Numerical evidence seems to suggest that this property holds for most plants that are localizable. For an ideal  $\mathcal{H}_2$  controller, the closed loop response decays exponentially in both time and space, as indicated by Figures 4(b) and 4(c). Therefore, it is generally possible to find a favorable  $(d, T)$  to synthesize the localized controller such that the closed loop transient response does not degrade much – this is akin to the insight in [27] used to localize the implementation of funnel-causal systems.

In summary, our result demonstrates that the LLQR optimal controller can be synthesized and implemented in a localized manner, but can achieve transient response close to that of an unconstrained optimal controller.

Table 1: Comparison between Different Control Schemes

	Ideal $\mathcal{H}_2$	Delayed	Distributed	LLQR
Comm Speed	Inf	1.5	1.5	1.5
Control Time	Inf	Inf	Inf	29
Locality	Max(58)	Max(58)	Max(58)	9
Objective	1	126.7882	1.1061	1.1142

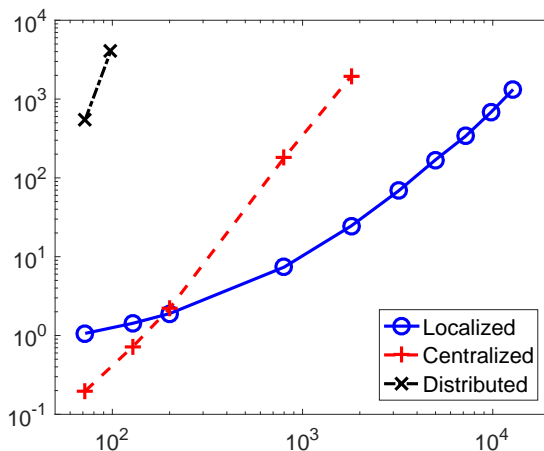


Figure 5: The horizontal axis denotes the number of states of the system, and the vertical axis is the computation time in seconds. The FIR horizon length is  $T = 7$ .

Figure 5 clearly shows the computation time needed for the distributed controller grows rapidly when the size of problem increases. The gradient of the curve corresponding to the localized controller is about 1.4 which is slightly larger than the theoretical value of 1. This issue is likely due to memory management issues and not related to the fundamental optimization problems. We further note that each of the distributed problems were solved in series on a single laptop, the performance would be significantly improved with a true parallel implementation.

## 4.5 Robustness

The previous sections highlighted the benefits of spatiotemporal locality: imposing FIR constraints trivially lead to finite-dimensional optimization problems (and simple filter-bank based controller implementations), and spatial locality leads to huge wins in terms of controller synthesis and implementation complexity. However, both of these constraints can often be fairly restrictive in practice: indeed, if a system is stabilizable, but not controllable, we do not expect any such constraints to hold exactly. This section extends the robustness results presented in Section 2.3

to the infinite horizon setting, allowing us to circumvent these issues. Further, we show that this allows us to formalize the folk theorem that “good controllers are easy to compute.”

#### 4.5.1 A robustness result

We begin with the infinite horizon analogue of the robustness result presented in Section 2.3 that provides necessary and sufficient conditions under which a controller implemented using transfer matrices that only approximately satisfy the achievability constraint (4.5) is stabilizing. We then leverage this result and the use of “virtual” actuation and communication resources at design time to define a principled approach to approximate controller synthesis.

**Theorem 4.3.** *Let  $(\hat{\Phi}_x, \hat{\Phi}_u, \Delta)$  be a solution to*

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} = I + \Delta, \quad \hat{\Phi}_x, \hat{\Phi}_u \in \frac{1}{z} \mathcal{RH}_\infty. \quad (4.22)$$

*Then, the controller implementation*

$$\hat{\delta}_x = \mathbf{x} - \hat{\mathbf{x}} \quad (4.23a)$$

$$\mathbf{u} = z\hat{\Phi}_u\hat{\delta}_x \quad (4.23b)$$

$$\hat{\mathbf{x}} = (z\hat{\Phi}_x - I)\hat{\delta}_x. \quad (4.23c)$$

*internally stabilizes the system  $(A, B_2)$  if and only if  $(I + \Delta)^{-1}$  is stable. Furthermore, the actual system responses achieved are given by*

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} (I + \Delta)^{-1} \delta_x. \quad (4.24)$$

**Proof.** For a state-feedback system (4.2) with controller implementation (4.23) using transfer matrices  $\{\hat{\Phi}_x, \hat{\Phi}_u\}$ , the following holds

$$(zI - A)\mathbf{x} = B_2\mathbf{u} + \delta_x \quad (4.25a)$$

$$\mathbf{u} = z\hat{\Phi}_u\hat{\delta}_x + \delta_u \quad (4.25b)$$

$$\mathbf{x} = z\hat{\Phi}_x\hat{\delta}_x - \delta_y, \quad (4.25c)$$

where  $\delta_y$  and  $\delta_u$  are respectively perturbations on the measurement and control action (as illustrated in Figure 2) introduced to verify the internal stability of the resulting closed loop system.

Substituting (4.25b) and (4.25c) into (4.25a), we have

$$z(zI - A)\hat{\Phi}_x\hat{\delta}_x - (zI - A)\delta_y = zB_2\hat{\Phi}_u\hat{\delta}_x + B_2\delta_u + \delta_x.$$

Moving  $\hat{\delta}_x$  to the left-hand-side and using relation (4.22), we have

$$z(I + \Delta)\hat{\delta}_x = (zI - A)\delta_y + B_2\delta_u + \delta_x,$$

Let  $\mathbf{I}_\Delta := (I + \Delta)^{-1}$ . We then have that the closed loop transfer matrices from  $(\delta_x, \delta_y, \delta_u)$  to  $\hat{\delta}_x$  are given by

$$\hat{\delta}_x = \frac{1}{z}\mathbf{I}_\Delta\delta_x + \mathbf{I}_\Delta\left(I - \frac{1}{z}A\right)\delta_y + \frac{1}{z}\mathbf{I}_\Delta B_2\delta_u. \quad (4.26)$$

Substituting (4.26) into (4.25b) and (4.25c), we have the closed loop transfer matrices from  $(\delta_x, \delta_y, \delta_u)$  to  $(\mathbf{x}, \mathbf{u}, \hat{\delta}_x)$  summarized in Table 2. Clearly, if  $\mathbf{I}_\Delta$  is stable, then all the transfer matrices in Table 2 are stable. If  $\mathbf{I}_\Delta$  is unstable, then the closed loop maps from  $\delta_x$  to  $\hat{\delta}_x$  will be unstable, and the controller does not internally stabilize the system. Therefore, the stability of  $\mathbf{I}_\Delta = (I + \Delta)^{-1}$  is necessary and sufficient condition for the controller implementation (4.23) to internally stabilize the system  $(A, B_2)$ . Finally, Table 2 shows that the actual response achieved from  $\delta_x \rightarrow (\mathbf{x}, \mathbf{u})$  is given by (4.24).  $\blacksquare$

Table 2: Closed Loop Maps With Non-localizability

	$\delta_x$	$\delta_y$	$\delta_u$
$\mathbf{x}$	$\hat{\Phi}_x \mathbf{I}_\Delta$	$\hat{\Phi}_x \mathbf{I}_\Delta (zI - A) - I$	$\hat{\Phi}_x \mathbf{I}_\Delta B_2$
$\mathbf{u}$	$\hat{\Phi}_u \mathbf{I}_\Delta$	$\hat{\Phi}_u \mathbf{I}_\Delta (zI - A)$	$I + \hat{\Phi}_u \mathbf{I}_\Delta B_2$
$\hat{\delta}_x$	$\frac{1}{z} \mathbf{I}_\Delta$	$\mathbf{I}_\Delta (I - \frac{1}{z} A)$	$\frac{1}{z} \mathbf{I}_\Delta B_2$

Theorem 4.3 can now be combined with small gain theorems to provide simple sufficient conditions for robust stability.

**Corollary 4.4** (Sufficient conditions for robustness). *Under the conditions of Theorem 4.3, the closed loop system is stable if  $\max\{\|\Delta\|_{\mathcal{H}_\infty}, \|\Delta\|_{\mathcal{L}_1}, \|\Delta^\top\|_{\mathcal{L}_1}\} < 1$ .*

It therefore follows that if a set of transfer matrices  $\{\hat{\Phi}_x, \hat{\Phi}_u, \Delta\}$  satisfy the following constraints:

$$[zI - A \quad -B_2] \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} = I + \Delta, \quad \hat{\Phi}_x, \hat{\Phi}_u \in \frac{1}{z} \mathcal{RH}_\infty, \quad \|\Delta\|_\bullet < 1, \quad (4.27)$$

for  $\bullet \in \{\mathcal{H}_\infty, \mathcal{L}_1, \mathcal{E}_1\}$ , then a controller (4.23) implemented using the transfer matrices  $\{\hat{\Phi}_x, \hat{\Phi}_u\}$  is globally stabilizing for the system dynamics (4.2).

From Theorem 4.3, we see that the actual closed loop map from the disturbance  $\delta_x$  to state  $\mathbf{x}$  and control action  $\mathbf{u}$  achieved by a controller implemented using the approximate system responses  $\{\hat{\Phi}_x, \hat{\Phi}_u\}$  is given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} (I + \Delta)^{-1} \delta_x. \quad (4.28)$$

Integrating the approximate achievability constraint (4.22), the sufficient conditions for stability imposed in by Corollary 4.4, and the system responses that they achieve (4.28) into a SLS optimization problem, we obtain the following:



$$\begin{aligned}
& \underset{\hat{\Phi}_x, \hat{\Phi}_u, \Delta}{\text{minimize}} && \left\| [C_1 \ D_{12}] \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} (I + \Delta)^{-1} B_1 \right\| \\
& \text{s.t.} && [zI - A \ -B_2] \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} = I + \Delta \\
& && \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} \in \mathcal{S}, \quad \|\Delta\| < 1 \\
& && \hat{\Phi}_x, \hat{\Phi}_u \in \frac{1}{z} \mathcal{RH}_\infty.
\end{aligned} \tag{4.29}$$

Optimization problem (4.29) is non-convex: however, if the norm chosen in the objective function is sub-multiplicative, we can upper bound the objective function by

$$\frac{\|B_1\|}{1 - \|\Delta\|} \left\| [C_1 \ D_{12}] \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} \right\|, \tag{4.30}$$

where we have used the sub-multiplicative property, that  $\|\Delta\| < 1$  and the power-series expansion of the the inverse  $(I + \Delta)^{-1} = \sum_{k=0}^{\infty} \Delta^k$ . As the next lemma shows, this bound is indeed quasi-convex and hence can be effectively optimized.

**Lemma 4.5.** *For a convex function  $f : \mathcal{D} \rightarrow \mathbb{R}$ , a non-negative convex function  $g : \mathcal{D} \rightarrow \mathbb{R}_+$ , both defined on some domain  $\mathcal{D}$ , and a convex set  $C \subseteq \mathcal{D}$ , it holds that*

$$\min_{x \in C} \frac{f(x)}{1 - g(x)} = \min_{\gamma \in [0,1]} \frac{1}{1-\gamma} \min_{x \in C} \{f(x) \mid g(x) \leq \gamma\} \tag{4.31}$$

**Proof.**(Sketch) As  $g(x)$  is non-negative, we can write

$$\begin{aligned}
\min_{x \in C} \frac{f(x)}{1 - g(x)} &= \min_{\gamma, x \in C} \frac{f(x)}{1 - \gamma} \quad \text{s.t.} \quad g(x) \leq \gamma \\
&= \min_{\gamma} \frac{1}{1-\gamma} \min_{x \in C} \{f(x) \mid g(x) \leq \gamma\},
\end{aligned}$$

which proves the result. Finally, it is easily verified that the above problem is quasi-convex, and hence uni-modal in  $\gamma$ .  $\blacksquare$

Thus applying the bound (4.30) and Lemma 4.5, we can compute an upper bound to optimization problem (4.29) by solving the following quasi-convex problem:

$$\begin{aligned}
& \min_{\gamma \in [0,1]} \frac{1}{1-\gamma} \underset{\hat{\Phi}_x, \hat{\Phi}_u, \Delta}{\text{minimize}} && \left\| [C_1 \ D_{12}] \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} \right\| \\
& \text{s.t.} && [zI - A \ -B_2] \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} = I + \Delta \\
& && \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} \in \mathcal{S}, \quad \|\Delta\| \leq \gamma \\
& && \hat{\Phi}_x, \hat{\Phi}_u \in \frac{1}{z} \mathcal{RH}_\infty.
\end{aligned} \tag{4.32}$$

We now give two examples of how this robustness result can be used. The first is an infinite horizon analogue to the modeling error result presented in Section 2.3, and the second shows how spatiotemporal approximations to the infinite horizon SLS problem can be taken and still achieve provably near optimal performance while still benefiting from the favorable computational properties of localized synthesis.

### 4.5.2 Robust control with sub-optimality guarantees

We return to the problem setting where estimates  $(\hat{A}, \hat{B})$  of a true system  $(A, B)$  satisfy

$$\|\Delta_A\|_{2 \rightarrow 2} \leq \epsilon_A, \quad \|\Delta_B\|_{2 \rightarrow 2} \leq \epsilon_B$$

with  $\Delta_A := \hat{A} - A$  and  $\Delta_B := \hat{B} - B$  and where we wish to minimize the worst-case LQR cost over the parametric uncertainty.

Leveraging the results of the previous section, and following a similar argument as that in Section 2.3, we can cast a robust LQR problem as follows

$$\begin{aligned} \text{minimize}_{\gamma \in (0,1)} \quad & \frac{1}{1-\gamma} \min_{\hat{\Phi}_x, \hat{\Phi}_u} \left\| \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_2} \\ \text{s.t.} \quad & \begin{bmatrix} zI - \hat{A} & -\hat{B} \end{bmatrix} \begin{bmatrix} \hat{\Phi}_x \\ \hat{\Phi}_u \end{bmatrix} = I, \quad \sqrt{2} \left\| \begin{bmatrix} \epsilon_A \hat{\Phi}_x \\ \epsilon_B \hat{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \gamma \\ & \hat{\Phi}_x, \hat{\Phi}_u \in \frac{1}{z} \mathcal{RH}_\infty. \end{aligned} \quad (4.33)$$

We note that this optimization objective is jointly quasi-convex in  $(\gamma, \hat{\Phi}_x, \hat{\Phi}_u)$ . Hence, as a function of  $\gamma$  alone the objective is quasi-convex, and furthermore is smooth in the feasible domain. Therefore, the outer optimization with respect to  $\gamma$  can effectively be solved with methods like golden section search. We remark that the inner optimization is a convex problem, though an infinite dimensional one, but as we show in the next subsection, a simple finite dimensional approximation to this problem leads to nearly equivalent performance.

We further remark that because  $\gamma \in [0, 1)$ , any feasible solution  $(\hat{\Phi}_x, \hat{\Phi}_u)$  to optimization problem (4.33) generates a controller  $\hat{\mathbf{K}} = \hat{\Phi}_u \hat{\Phi}_x^{-1}$  satisfying the conditions of Corollary 4.4, and hence stabilizes the true system  $(A, B)$ . Therefore, even if the solution is approximated, as long as it is feasible, it will be stabilizing. We show now that for sufficiently small estimation error bounds  $\epsilon_A$  and  $\epsilon_B$ , we can further bound the sub-optimality of the performance achieved by our robustly stabilizing controller relative to that achieved by the optimal LQR controller  $\mathbf{K}_*$ .

**Theorem 4.6** (Theorem 4.1, [19]). *Let  $J_*$  denote the minimal LQR cost achievable by any controller for the dynamical system with transition matrices  $(A, B)$ ,  $J(A, B, \mathbf{K})$  be the cost achieved by any controller  $\mathbf{K}$  on the system defined by transition matrices  $(A, B)$ , and let  $\mathbf{K}_*$  denote the optimal controller. Let  $(\hat{A}, \hat{B})$  be estimates of the transition matrices such that  $\|\Delta_A\|_{2 \rightarrow 2} \leq \epsilon_A$ ,  $\|\Delta_B\|_{2 \rightarrow 2} \leq \epsilon_B$ . Then, if  $\hat{\mathbf{K}} = \hat{\Phi}_u \hat{\Phi}_x^{-1}$  is synthesized via (4.33) the relative error in the LQR cost is*

$$\frac{J(A, B, \hat{\mathbf{K}}) - J_*}{J_*} \leq 5(\epsilon_A + \epsilon_B \|\mathbf{K}_*\|_{2 \rightarrow 2}) \left\| (zI - (A + B\mathbf{K}_*))^{-1} \right\|_{\mathcal{H}_\infty}, \quad (4.34)$$

as long as  $(\epsilon_A + \epsilon_B \|\mathbf{K}_*\|_{2 \rightarrow 2}) \left\| (zI - (A + B\mathbf{K}_*))^{-1} \right\|_{\mathcal{H}_\infty} \leq 1/5$ .

This result offers a guarantee on the performance of the SLS synthesized controller as a function of the uncertainty in our model parameters – as far as we are aware, this is the first such result that provides a quantitative measure of performance degradation incurred by a robust controller as a function system uncertainty size. We refer the reader to [19] for the proof this result, but

remark here that Theorem 4.6 allows one to transparently connect statistical estimation results (which provide non-asymptotic guarantees on the size of the estimation errors  $\epsilon_A$  and  $\epsilon_B$ ) with robust control, allowing for end-to-end sample complexity bounds to be obtained. In this sense, the robust version of SLS provides a bridge between machine learning and robust control.

### 4.5.3 Spatiotemporal approximations

We begin with the simpler setting of considering only an FIR approximation. In particular, consider the general SLS optimization problem (4.12), where for simplicity, we assume that there are no additional constraints  $\mathcal{S}$ , resulting in a convex but infinite-horizon optimal control problem. For concreteness, we will consider the following mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  optimal control problem – these methods extend to the more general setting in a natural way:

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \quad & \left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_2} + \lambda \left\| \begin{bmatrix} Q_\infty^{1/2} & \\ & R_\infty^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \\ \text{subject to} \quad & [zI - A \quad -B_2] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \\ & \Phi_x, \Phi_u \in \frac{1}{z} \mathcal{RH}_\infty, \end{aligned} \tag{4.35}$$

where  $\lambda > 0$  is a user-specified weighting parameter.

Although there is no known exact solution to this problem, we show here that the robustness results of the previous section can be used to formulate a finite-dimensional approximation to (4.35) that has provably near optimal performance. In particular, we consider the following approximate SLS problem:

$$\begin{aligned} \min_{\gamma \in (0,1)} \frac{1}{1-\gamma} \quad & \left( \min_{\Phi_x, \Phi_u} \left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_2} + \lambda \left\| \begin{bmatrix} Q_\infty^{1/2} & \\ & R_\infty^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \right) \\ \text{subject to} \quad & [zI - A \quad -B_2 \quad -I] \begin{bmatrix} \Phi_x \\ \Phi_u \\ \frac{1}{z^T} V \end{bmatrix} = I \\ & \|V\|_{2 \rightarrow 2} \leq \gamma, \Phi_x, \Phi_u \in \mathcal{F}_T. \end{aligned} \tag{4.36}$$

Before analyzing performance of the controller implemented using the solution to optimization problem (4.36), let us highlight the changes that have been made. First, we have imposed that the synthesized system responses  $\{\Phi_x, \Phi_u\}$  lie in  $\mathcal{F}_T$ , i.e., that they be FIR of horizon  $T$ . Although we argued that this was a desirable constraint to impose, as it immediately yields a finite-dimensional optimization problem, in practice, the system  $(A, B_2)$  may not be controllable, meaning that such constraints are infeasible for any value of  $T$ . Further, it is well known that imposing such FIR constraints, resulting in deadbeat control, can lead to robustness issues in implementation.

To circumvent these issues, we also introduce an additional term  $\frac{1}{z^{T-1}}V$  to the achievability constraint: this term can be thought of as the introduction of a centralized virtual actuator that can only act at time  $T - 1$ . This virtual actuator ensures that the FIR constraint is in fact feasible for any value of  $T \geq 1$  (simply set  $V = -A\Phi_x[T - 1] - B\Phi_u[T - 1]$ ). However, as this virtual actuator does not in fact exist, we must account for this in our analysis. To do so, notice that the

modified achievability constraint is equivalent to

$$[zI - A \quad -B_2] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I + \frac{1}{z^{T-1}} V,$$

allowing us to leverage Theorem 4.3 and Corollary 4.4 with  $\Delta = z^{-(T-1)}V$ . Following a similar argument to that used above, so long as  $\|z^{-(T-1)}V\|_{\mathcal{H}_\infty} = \|V\|_{2 \rightarrow 2} < 1$  (which is enforced by the constraints of the optimization problem), we can then upper bound the achieved objective function by their robust counterparts, leading to the modified objective function.

We now show that the performance of the controller synthesized using optimization problem (4.36) converges exponentially quickly to that achieved by the optimal solution to the infinite horizon problem (4.35) with respect to the approximation horizon  $T$ .

In particular, let  $\Phi_x^*$  and  $\Phi_u^*$  denote the optimal solutions to problem (4.35), and let  $\mathbf{K}_* = \Phi_u^*(\Phi_x^*)^{-1}$  be the resulting optimal controller. As  $\Phi_x^* = \sum_{t=1}^{\infty} z^{-t}\Phi_x[t] \in \frac{1}{z}\mathcal{RH}_\infty$ , there exist constants  $C_*$  and  $\rho_*$  such that  $\|\Phi_x^*[t]\|_{2 \rightarrow 2} \leq C_*\rho_*^t$  for all  $t \geq 0$ . We leverage this observation to construct a feasible solution to (4.36) using  $\Phi_x^*$  and  $\Phi_u^*$  to obtain a sub-optimality bound in terms of the horizon  $T$ .

**Theorem 4.7.** *If the approximation horizon  $T$  is sufficiently large to ensure that  $C_*\rho_*^T < 1$ , then the approximate SLS problem (4.36) is feasible, and achieves the relative performance bound:*

$$J_T(\Phi_x, \Phi_u, V, \gamma) \leq \frac{1}{1 - C_*\rho_*^T} J_* + \frac{\lambda C_*\rho_*^T}{(1 - C_*\rho_*^T)(1 - \rho_*^T)} (\|Q_\infty^{1/2}\|_{2 \rightarrow 2} + \|R_\infty^{1/2}\|_{2 \rightarrow 2} \|\mathbf{K}_*\|_{\mathcal{H}_\infty}). \quad (4.37)$$

Here  $J_T(\Phi_x, \Phi_u, V, \gamma)$  is the performance achieved by the controller synthesized using the approximate SLS problem (4.36) with horizon  $T$ , and  $J_*$  is the performance achieved by the optimal solution to (4.35).

Before presenting the proof of this result, we comment on its implications. First, we see very clearly that the gap between  $J_T$  and  $J_*$  decays exponentially with the horizon  $T$ , and further, this rate is specified by  $\rho_*$ , which governs the spectral norm decay of true system response elements. Informally, this theorem shows that if the optimal controller leads to good system performance (i.e., to rapid decay of the system response elements), then it is easy to approximate.

**Proof.** Let  $(\Phi_x, \Phi_u, V, \gamma)$  denote the optimal solutions to the approximate SLS problem (4.36), and denote by  $J_T(\Phi_x, \Phi_u, V, \gamma)$  the value of the objective function over a finite horizon  $T$ .

We now construct a feasible solution to the problem using the optimal system responses  $\Phi_x^*$  and  $\Phi_u^*$ . In particular, we claim that

$$\begin{aligned} \tilde{\Phi}_x &:= \Phi_x^*(1 : T - 1) \\ \tilde{\Phi}_u &:= \Phi_u^*(1 : T - 1) \\ \tilde{V} &:= -A\Phi_x^*[T - 1] - B\Phi_u^*[T - 1] = -\Phi_x^*[T] \\ \tilde{\gamma} &:= C_*\rho_*^T, \end{aligned}$$

is a feasible solution. First, notice that for  $t = 0, \dots, T - 2$ , the achievability constraint (4.5) is equivalent to

$$\Phi_x[t + 1] = A\Phi_x[t] + B_2\Phi_u[t].$$

Since  $\{\tilde{\Phi}_x, \tilde{\Phi}_u\}$  are simply truncations of  $\{\Phi_x^*, \Phi_u^*\}$  to a horizon of  $T - 1$ , they satisfy these constraints by construction. Finally, for  $t = T - 1$ , we have that

$$\Phi_x[T] = A\Phi_x^*[T - 1] + B\Phi_u^*[T - 1] + \tilde{V} = 0,$$

where the final equality follows from our choice of  $\tilde{V}$  to effectively cancel out the tail of optimal response. Finally, note that  $\|\tilde{V}\|_{2 \rightarrow 2} = \|\Phi_x^*[T]\|_{2 \rightarrow 2} \leq C_* \rho_*^T < 1$ , where the final inequality follows from our assumption on  $T$ .

Therefore, by optimality, it follows that

$$\begin{aligned} J_T(\Phi_x, \Phi_u, V, \gamma) &\leq J_T(\tilde{\Phi}_x, \tilde{\Phi}_u, \tilde{V}, \tilde{\gamma}) \\ &= \frac{1}{1 - C_* \rho_*^T} \left( \left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{\Phi}_x \\ \tilde{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_2} + \lambda \left\| \begin{bmatrix} Q_\infty^{1/2} & \\ & R_\infty^{1/2} \end{bmatrix} \begin{bmatrix} \tilde{\Phi}_x \\ \tilde{\Phi}_u \end{bmatrix} \right\|_{\mathcal{H}_\infty} \right). \end{aligned}$$

Our final step is to relate the norms achieved by the FIR approximations  $\Phi_x^*(0 : T - 1)$  and  $\Phi_u^*(0 : T - 1)$  to those achieved by  $\Phi_x^*$  and  $\Phi_u^*$ . We begin by noting that for any stable transfer matrix  $\mathbf{M} \in z^{-1}\mathcal{RH}_\infty$ , we have that  $\|\mathbf{M}(1 : T)\|_{\mathcal{H}_2} \leq \|\mathbf{M}\|_{\mathcal{H}_2}$ , for any  $T \geq 0$ . The  $\mathcal{H}_\infty$  case is slightly more cumbersome, as such an inequality does not hold exactly, but notice that we can write

$$\left\| \begin{bmatrix} Q_\infty^{1/2} & \\ & R_\infty^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x^*(1 : T - 1) \\ \Phi_u^*(1 : T - 1) \end{bmatrix} \right\|_{\mathcal{H}_\infty} \leq \left\| \begin{bmatrix} Q_\infty^{1/2} & \\ & R_\infty^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x^* \\ \Phi_u^* \end{bmatrix} \right\|_{\mathcal{H}_\infty} + \left\| \begin{bmatrix} Q_\infty^{1/2} & \\ & R_\infty^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x^*(T : \infty) \\ \Phi_u^*(T : \infty) \end{bmatrix} \right\|_{\mathcal{H}_\infty}$$

where the inequality follows from the triangle inequality. Finally, notice that last term can further be bounded as

$$\begin{aligned} \left\| \begin{bmatrix} Q_\infty^{1/2} & \\ & R_\infty^{1/2} \end{bmatrix} \begin{bmatrix} \Phi_x^*(T : \infty) \\ \Phi_u^*(T : \infty) \end{bmatrix} \right\|_{\mathcal{H}_\infty} &\leq (\|Q_\infty^{1/2}\|_{2 \rightarrow 2} + \|R_\infty^{1/2}\|_{2 \rightarrow 2} \|\mathbf{K}_*\|_{\mathcal{H}_\infty}) \|\Phi_x^*(T : \infty)\|_{\mathcal{H}_\infty} \\ &\leq (\|Q_\infty^{1/2}\|_{2 \rightarrow 2} + \|R_\infty^{1/2}\|_{2 \rightarrow 2} \|\mathbf{K}_*\|_{\mathcal{H}_\infty}) \sum_{t=T}^{\infty} \|\Phi_x^*(t)\|_{2 \rightarrow 2} \\ &\leq (\|Q_\infty^{1/2}\|_{2 \rightarrow 2} + \|R_\infty^{1/2}\|_{2 \rightarrow 2} \|\mathbf{K}_*\|_{\mathcal{H}_\infty}) C_* \sum_{t=T}^{\infty} \rho_*^t \\ &= (\|Q_\infty^{1/2}\|_{2 \rightarrow 2} + \|R_\infty^{1/2}\|_{2 \rightarrow 2} \|\mathbf{K}_*\|_{\mathcal{H}_\infty}) \frac{C_* \rho_*^T}{1 - \rho_*^T}. \end{aligned}$$

Combining this bound with the previous inequalities then yield the result.  $\blacksquare$

We conclude this section by remarking that the above proof contains a general recipe for generating such suboptimality bounds. Begin with the desired infinite horizon problem, and then write out its FIR approximation by augmenting the system with virtual actuation, and then exploiting the robustness results in Theorem 4.3 and Corollary 4.4. Then, construct a feasible solution to the approximate problem using the optimal solution to the infinite horizon problem and follow the general approach of the proof above to obtain a suboptimality bound by exploiting the exponential decay of stable LTI systems. Finally, to extend these results from temporal (FIR) approximations to spatiotemporal (localized) approximations, it suffices to impose a localized SLC on the system responses, and further to assume that the optimal solutions also suitably decay in space according to some function  $f(d)$ —the resulting sub-optimality bounds will be similar, but now decaying exponentially as a function of the horizon  $T$ , and according to  $f(d)$  with respect to the localized region diameter  $d$ .

#### 4.5.4 Examples

Now that we have established a theoretical robustness result we shall use this subsection to illustrate the effect of Theorem 4.3 when  $\|\Delta\| \neq 0$ . Consider the symmetric bi-directional chain topology, where each node in the chain is a scalar LTI system evolving according to the dynamics

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

where  $\alpha$  is chosen to make the open-loop system unstable with a spectral radius of 1.1. The coupling strength  $\kappa$  is set to 1. 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 network at locations specified by  $i = 5j - 4$  and  $5j$  for  $j = 1, \dots, 20$ . The objective function is the quadratic term  $\|\mathbf{x}\|_2^2 + \gamma\|\mathbf{u}\|_2^2$ .

We begin by showing the system response for a system with actuation at every other node. By choosing  $(d, T)$ -locality constraints with  $d = 4$ ,  $T = 15$  and restricting the communication speed to be equal to the speed at which the plant dynamics propagate, the system response cannot be perfectly localized. However, by employing Theorem 4.3, the robust SLS problem finds a solution such that  $\|\Delta\| < 1$ . The state and control action from the resulting controller are shown in Figure 6.

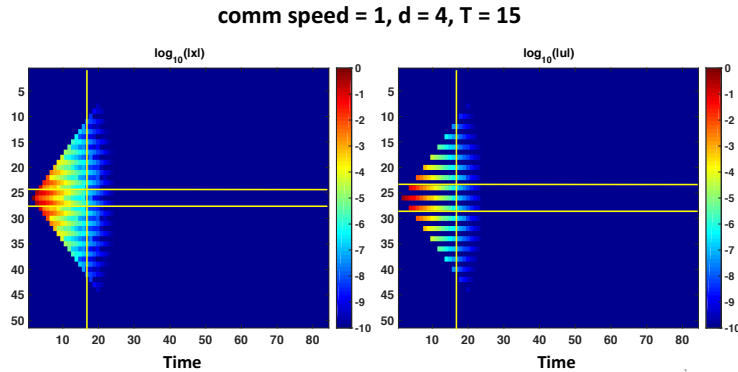


Figure 6: With the dynamics propagating at the same speed as the controllers communicate and with such sparse actuation, we are now unable to localize in space and time as desired. However, the control action and state deviation is reasonably well contained.

In subsequent experiments shown in Figures 7–9 the system responses are shown for systems where we trade off the actuator density (Fig 7),  $d$ -hop locality (Figure 8), and FIR horizon length (Figure 9). In each case we show a parameter value that results in a localizable system response (shown on the left) and the virtually localizable response (shown on the right).

#### 4.6 Summary

In this section, we extended the state-feedback finite-time horizon results from Section 2 to the infinite horizon settings. We further showed that the SLS framework allows us to impose convex locality constraints on distributed controllers, greatly improving the scalability of synthesis methods via a decomposition and dimensionality reduction based method. We also showed how robust counterparts to SLS problems can be used to transparently accommodate modeling errors, as well as how they can be used to derive finite dimensional approximations to infinite horizon problems

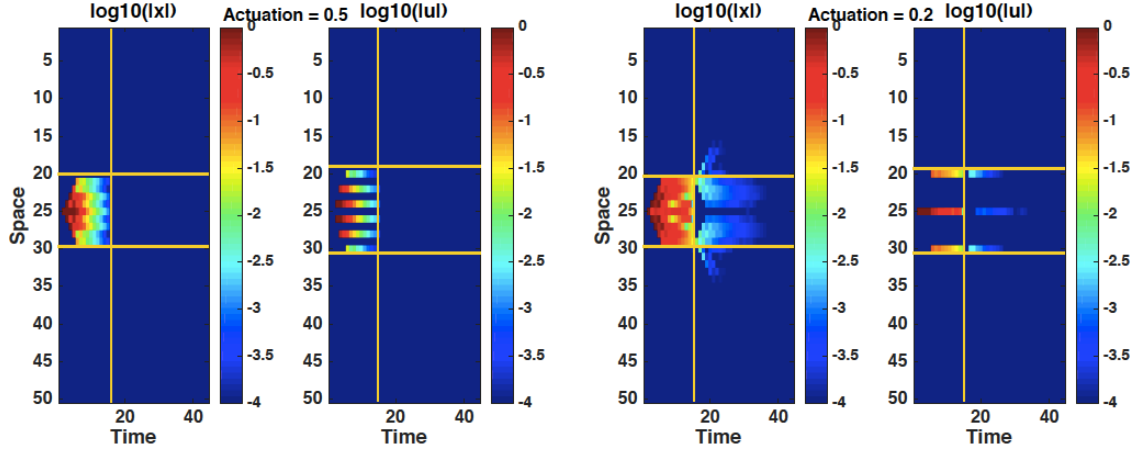


Figure 7: As the number of actuators available for control decreases it becomes more difficult to obtain a localized response. The  $\mathcal{H}_2$ -norm for the actuation = 0.5 is 79.5, and for actuation = 0.2 it is 97.52 with,  $\|\Delta\|_{\mathcal{E}_1} = 0.25$ .

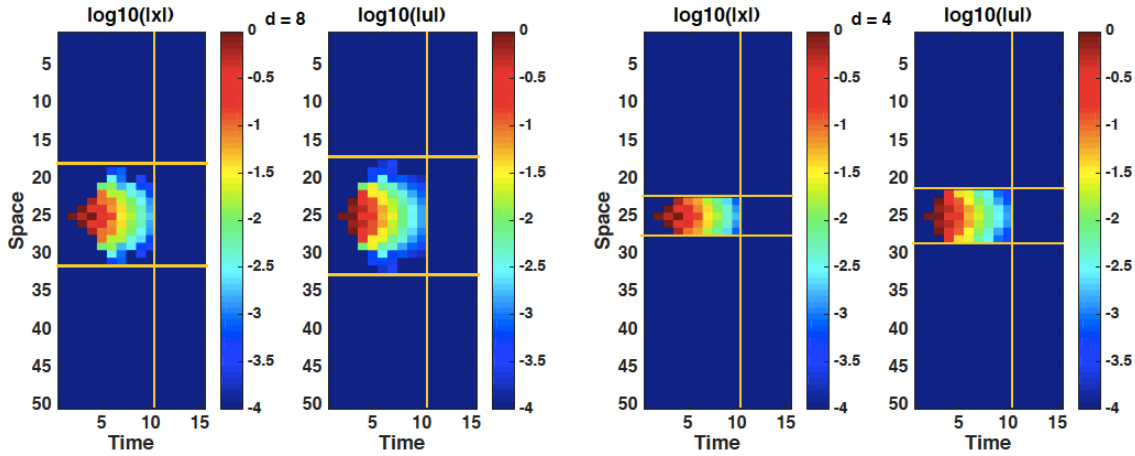


Figure 8: The  $d$ -hop sparsity is decreased from  $d = 8$  to  $d = 4$ . The closed loop  $\mathcal{H}_2$ -norm is 72.57 and 72.59 for  $d = 8$  and  $d = 4$  respectively. No virtual localization was required.

with provable performance guarantees. In addition to the results presented here, there has also been interesting work looking at linking the SLS state-feedback formulation to classical notions such as passivity [62] and infinite-dimensional system theory and the spatial invariance framework [63]. In [64] the  $\Delta$ -block is used to model discretization errors when a sparse approximate model is constructed from a structured continuous-time dynamics.

## 5 Output Feedback

This section extends many of the ideas presented in the previous section to the output feedback setting. As we will see, many of the core concepts developed in the state-feedback setting (working

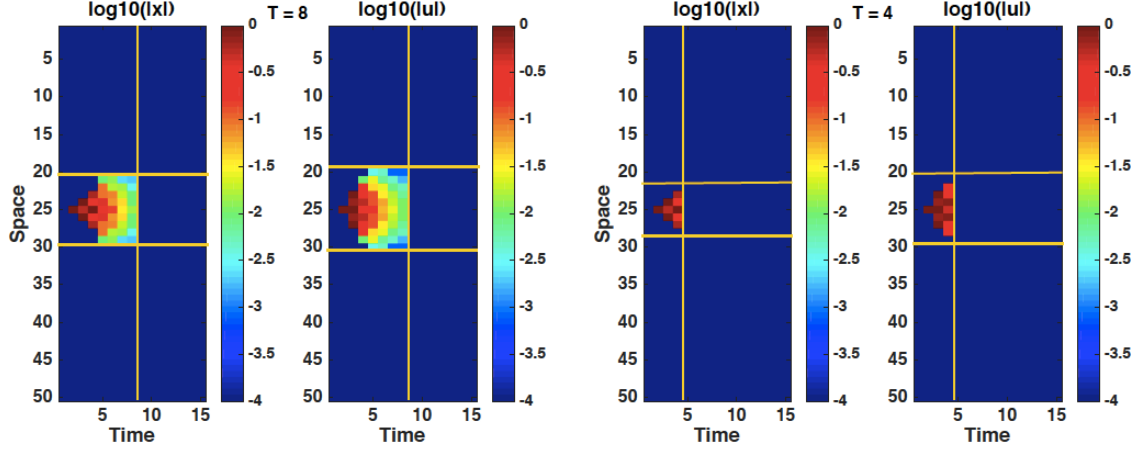


Figure 9: The FIR Horizon length is decreased from  $T = 8$  to  $T = 4$ . In neither case was virtual actuation required. The  $\mathcal{H}_2$ -norm was 72.57 for  $T = 8$  and 72.96 for  $T = 4$ .

directly with system responses, affine achievability constraints) extend in a natural way to the output feedback setting. However, these extensions often come at the expense of more complex formulae and computational procedures.

### 5.1 Output Feedback with $D_{22} = 0$

We begin by extending Theorem 4.1 to the output feedback setting. Without loss of generality (see later in the section), we consider a strictly proper plant described by

$$\mathbf{P} = \left[ \begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array} \right]. \quad (5.1)$$

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 (5.1) can be written as

$$\begin{aligned} x[t+1] &= Ax[t] + B_2 u[t] + \delta_x[t] \\ y[t] &= C_2 x[t] + \delta_y[t]. \end{aligned} \quad (5.2)$$

Analogous to the state-feedback case, we define a system response  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$  from perturbations  $(\delta_x, \delta_y)$  to state and control inputs  $(\mathbf{x}, \mathbf{u})$  via the following relation:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}. \quad (5.3)$$

Substituting the output feedback control law  $\mathbf{u} = \mathbf{K}\mathbf{y}$  into the z-transform of system equation (5.2), we obtain

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

For a proper controller  $\mathbf{K}$ , the transfer matrix  $(zI - A - B_2 \mathbf{K} C_2)$  is always invertible, hence we obtain the following equivalent expressions for the system response (5.3) in terms of an output



feedback controller  $\mathbf{K}$ :

$$\begin{aligned}
\Phi_{xx} &= (zI - A - B_2\mathbf{K}C_2)^{-1} \\
\Phi_{ux} &= \mathbf{K}C_2\Phi_{xx} \\
\Phi_{xy} &= \Phi_{xx}B_2\mathbf{K} \\
\Phi_{uy} &= \mathbf{K} + \mathbf{K}C_2\Phi_{xx}B_2\mathbf{K}.
\end{aligned} \tag{5.4}$$

We now present one of the main results of this section: an algebraic characterization of the set  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$  of output-feedback system responses that are achievable by an internally stabilizing controller  $\mathbf{K}$ .

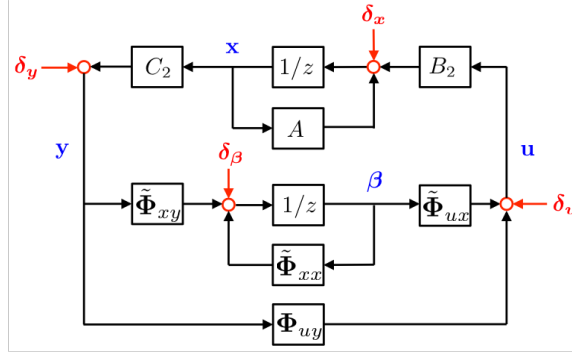


Figure 10: The proposed output feedback controller structure, with  $\tilde{\Phi}_{xx} = z(I - z\Phi_{xx})$ ,  $\tilde{\Phi}_{ux} = z\Phi_{ux}$ , and  $\tilde{\Phi}_{xy} = -z\Phi_{xy}$ .

**Theorem 5.1.** *For the output feedback system (5.1), the following are true:*

(a) *The affine subspace described by:*

$$[zI - A \quad -B_2] \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} = [I \quad 0] \tag{5.5a}$$

$$\begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{5.5b}$$

$$\Phi_{xx}, \Phi_{ux}, \Phi_{xy} \in \frac{1}{z}\mathcal{RH}_\infty, \quad \Phi_{uy} \in \mathcal{RH}_\infty \tag{5.5c}$$

*parameterizes all system responses (5.4) achievable by an internally stabilizing controller  $\mathbf{K}$ .*

(b) *For any transfer matrices  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$  satisfying (5.5), the controller  $\mathbf{K} = \Phi_{uy} - \Phi_{ux}\Phi_{xx}^{-1}\Phi_{xy}$  achieves the desired response (5.4).<sup>9</sup> Further, if the controller is implemented as in Fig. 10, then it is internally stabilizing.*

As in the state-feedback case, although not explicitly enforced, one cannot circumvent the need for a stabilizable and detectable system, as formalized in the following lemma.

<sup>9</sup>Note that for any transfer matrices  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$  satisfying (5.5), the transfer matrix  $\Phi_{xx}$  is always invertible because its leading spectral component  $\frac{1}{z}I$  is invertible. The same holds true for the transfer matrices defined in equation (5.4).

**Lemma 5.2.** *The triple  $(A, B_2, C_2)$  is stabilizable and detectable if and only if the affine subspace described by (5.5) is non-empty.*

**Proof.** We begin by noting the analysis for the state feedback problem in Section 4 can be applied to the state estimation problem by considering the dual to a full control system (c.f., §16.5 in [58]). For instance, the following corollary to Lemma 4.2 gives an alternative definition of the detectability of pair  $(A, C_2)$  [48].

**Corollary 5.3.** *The pair  $(A, C_2)$  is detectable if and only if the following conditions are feasible:*

$$\begin{bmatrix} \Phi_{xx} & \Phi_{xy} \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = I \quad (5.6a)$$

$$\Phi_{xx}, \Phi_{xy} \in \frac{1}{z} \mathcal{RH}_\infty. \quad (5.6b)$$

Now assume the triple  $(A, B_2, C_2)$  is stabilizable and detectable. As argued in Lemma 4.2 and Corollary 5.3, this implies that there exist stable transfer matrices  $(\Phi_{xx}^1, \Phi_{ux}^1)$  satisfying (4.5), and stable transfer matrices  $(\Phi_{xx}^2, \Phi_{xy}^2)$  satisfying (5.6). Using these transfer matrices, we construct the stable transfer matrices, which can be verified to lie in the affine space (5.5)

$$\Phi_{xx} = \Phi_{xx}^1 + \Phi_{xx}^2 - \Phi_{xx}^1(zI - A)\Phi_{xx}^2 \quad (5.7a)$$

$$\Phi_{ux} = \Phi_{ux}^1 - \Phi_{ux}^1(zI - A)\Phi_{xx}^2 \quad (5.7b)$$

$$\Phi_{xy} = \Phi_{xy}^2 - \Phi_{xx}^1(zI - A)\Phi_{xy}^2 \quad (5.7c)$$

$$\Phi_{uy} = -\Phi_{ux}^1(zI - A)\Phi_{xy}^2. \quad (5.7d)$$

■

This lemma provides a sanity check, and in particular tells us that the set of achievable stable system responses for the triple  $(A, B_2, C_2)$  is non-empty if and only if the system is stabilizable and detectable.

**Proof.**[Proof of Theorem 5.1]

Proof of 1. Consider the output feedback system (5.1). Let  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$ , with  $\mathbf{x} = \Phi_{xx}\delta_x + \Phi_{xy}\delta_y$  and  $\mathbf{u} = \Phi_{ux}\delta_x + \Phi_{uy}\delta_y$ , be the system response achieved by an internally stabilizing control law  $\mathbf{u} = \mathbf{K}\mathbf{y}$ . Then  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$  lies in the affine subspace described by (5.5).

To see this, let the internally stabilizing controller  $\mathbf{K}$  have state space realization (3.2). Combining (3.2) with the system equation (5.2), we obtain the closed loop dynamics

$$\begin{bmatrix} z\mathbf{x} \\ z\boldsymbol{\xi} \end{bmatrix} = \begin{bmatrix} A + B_2D_kC_2 & B_2C_k \\ B_kC_2 & A_k \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} I & B_2D_k \\ 0 & B_k \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}.$$

From the assumption that  $\mathbf{K}$  is internally stabilizing, we know that the state matrix of the above equation is a stable matrix (Lemma 5.2 in [58]). The system response achieved by  $\mathbf{u} = \mathbf{K}\mathbf{y}$  is given by

$$\begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} = \left[ \begin{array}{cc|cc} A + B_2D_kC_2 & B_2C_k & I & B_2D_k \\ B_kC_2 & A_k & 0 & B_k \\ \hline I & 0 & 0 & 0 \\ D_kC_2 & C_k & 0 & D_k \end{array} \right], \quad (5.8)$$

Table 3: Closed Loop Maps from Perturbations to Internal Variables

	$\delta_x$	$\delta_y$	$\delta_u$	$\delta_\beta$
$\mathbf{x}$	$\Phi_{xx}$	$\Phi_{xy}$	$\Phi_{xx}B_2$	$\frac{1}{z}\Phi_{xy}C_2$
$\mathbf{u}$	$\Phi_{ux}$	$\Phi_{uy}$	$I + \Phi_{ux}B_2$	$\frac{1}{z}\Phi_{uy}C_2$
$\mathbf{y}$	$C_2\Phi_{xx}$	$I + C_2\Phi_{xy}$	$C_2\Phi_{xx}B_2$	$\frac{1}{z}C_2\Phi_{xy}C_2$
$\beta$	$-\frac{1}{z}B_2\Phi_{ux}$	$-\frac{1}{z}B_2\Phi_{uy}$	$-\frac{1}{z}B_2\Phi_{ux}B_2$	$\frac{1}{z}I - \frac{1}{z^2}(A + B_2\Phi_{uy}C_2)$

which satisfies (5.5c). In addition, it can be shown by routine calculation that (5.8) satisfies both (5.5a) and (5.5b).

Proof of 2. Here we show that for any system response  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$  lying in the affine subspace (5.5), we can construct an internally stabilizing controller  $\mathbf{K}$  that leads to the desired system response (5.4). From the relations in (5.4), we notice the identity  $\mathbf{K} = \Phi_{uy} - \mathbf{K}C_2\Phi_{xx}B_2\mathbf{K} = \Phi_{uy} - \Phi_{ux}\Phi_{xx}^{-1}\Phi_{xy}$ . This relation leads to the controller structure given in Figure 10, with  $\tilde{\Phi}_{xx} = z(I - z\Phi_{xx})$ ,  $\tilde{\Phi}_{ux} = z\Phi_{ux}$ , and  $\tilde{\Phi}_{xy} = -z\Phi_{xy}$ . As was the case for the state feedback setting, it can be verified that all three of these transfer matrices lie in  $\mathcal{RH}_\infty$ . Therefore, the structure given in Figure 10 is well defined. In addition, all of the blocks in Figure 10 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  $\delta_x, \delta_y, \delta_u$ , and  $\delta_\beta$  to the system. The perturbations appearing on other links of the block diagram can all be expressed as a combination of the perturbations  $(\delta_x, \delta_y, \delta_u, \delta_\beta)$  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  $(\delta_x, \delta_y, \delta_u, \delta_\beta)$  to controller signals  $(\mathbf{x}, \mathbf{u}, \mathbf{y}, \beta)$  to determine the internal stability of the structure [58].

For any system response  $\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}$  lying in the affine subspace defined by (5.5), we construct a controller using the structure given in Figure 10. We now check the stability of the closed loop transfer matrices from the perturbations  $(\delta_x, \delta_y, \delta_u, \delta_\beta)$  to the internal variables  $(\mathbf{x}, \mathbf{u}, \mathbf{y}, \beta)$ . We have the following equations from Figure 10:

$$\begin{aligned}
 z\mathbf{x} &= A\mathbf{x} + B_2\mathbf{u} + \delta_x \\
 \mathbf{y} &= C_2\mathbf{x} + \delta_y \\
 z\beta &= \tilde{\Phi}_{xx}\beta + \tilde{\Phi}_{xy}\mathbf{y} + \delta_\beta \\
 \mathbf{u} &= \tilde{\Phi}_{ux}\beta + \Phi_{uy}\mathbf{y} + \delta_u.
 \end{aligned}$$

Combining these equations with the relations in (5.5a)–(5.5b), we summarize the closed loop transfer matrices from  $(\delta_x, \delta_y, \delta_u, \delta_\beta)$  to  $(\mathbf{x}, \mathbf{u}, \mathbf{y}, \beta)$  in Table 3.

Equation (5.5c) implies that all sixteen transfer matrices in Table 3 are stable, so the implementation in Figure 10 is internally stable. Furthermore, the desired system response from  $(\delta_x, \delta_y)$  to  $(\mathbf{x}, \mathbf{u})$  is achieved. ■

The controller implementation of Figure 10 is governed by the following equations:

$$\begin{aligned} z\boldsymbol{\beta} &= \tilde{\boldsymbol{\Phi}}_{xx}\boldsymbol{\beta} + \tilde{\boldsymbol{\Phi}}_{xy}\mathbf{y} \\ \mathbf{u} &= \tilde{\boldsymbol{\Phi}}_{ux}\boldsymbol{\beta} + \boldsymbol{\Phi}_{uy}\mathbf{y}, \end{aligned} \quad (5.9)$$

which can be informally interpreted as an extension of the state-space realization (3.2) of a controller  $\mathbf{K}$ . In particular, the realization equations (5.9) can be viewed as a state-space like implementation where the constant matrices  $A_K, B_K, C_K, D_K$  of the state-space realization (3.2) are replaced with stable proper transfer matrices  $\tilde{\boldsymbol{\Phi}}_{xx}, \tilde{\boldsymbol{\Phi}}_{ux}, \tilde{\boldsymbol{\Phi}}_{xy}, \boldsymbol{\Phi}_{uy}$ . The benefit of this implementation is that arbitrary convex constraints imposed on the transfer matrices  $\tilde{\boldsymbol{\Phi}}_{xx}, \tilde{\boldsymbol{\Phi}}_{ux}, \tilde{\boldsymbol{\Phi}}_{xy}, \boldsymbol{\Phi}_{uy}$  carry over directly to the controller implementation. We show in Section 5.5, as in the state-feedback setting, that this transparency 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. In contrast, we recall that imposing general convex constraints on the controller  $\mathbf{K}$  or directly on its state-space realization  $A_K, B_K, C_K, D_K$  do not lead to convex optimal control problems.

**Remark 5.4.** *The controller implementation (5.9) admits the following equivalent representation*

$$\begin{bmatrix} \tilde{\boldsymbol{\Phi}}_{xx} & \tilde{\boldsymbol{\Phi}}_{xy} \\ \tilde{\boldsymbol{\Phi}}_{ux} & \boldsymbol{\Phi}_{uy} \end{bmatrix} \begin{bmatrix} z\boldsymbol{\beta} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix}, \quad (5.10)$$

allowing for an interesting interpretation of the controller  $\mathbf{K} = \boldsymbol{\Phi}_{uy} - \tilde{\boldsymbol{\Phi}}_{ux}\tilde{\boldsymbol{\Phi}}_{xx}^{-1}\tilde{\boldsymbol{\Phi}}_{xy}$  in terms of Rosenbrock system matrix representations [6]. In particular, the system response (5.3) specifies a Rosenbrock system matrix representation of the controller that achieves it.

## 5.2 Specialized Implementations for Open-loop Stable Systems

In this subsection, we propose two specializations of the controller implementation in Figure 10 for open loop stable systems. From Table 3, if we set  $\boldsymbol{\delta}_u$  and  $\boldsymbol{\delta}_\beta$  to 0, it follows that  $\boldsymbol{\beta} = -\frac{1}{z}B_2\mathbf{u}$ . This leads to a simpler controller implementation given by  $\mathbf{u} = \boldsymbol{\Phi}_{uy}\mathbf{y} - \boldsymbol{\Phi}_{ux}B_2\mathbf{u}$ , with the corresponding controller structure shown in Figure 11(b). This implementation can also be obtained from the identity  $\mathbf{K} = (I + \boldsymbol{\Phi}_{ux}B_2)^{-1}\boldsymbol{\Phi}_{uy}$ , which follows from the relations in (5.4). 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 11(b), the closed loop transfer matrices from perturbations to the internal variables are given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\Phi}}_{xx} & \tilde{\boldsymbol{\Phi}}_{xy} & \tilde{\boldsymbol{\Phi}}_{xx}B_2 & (zI - A)^{-1}B_2 \\ \tilde{\boldsymbol{\Phi}}_{ux} & \boldsymbol{\Phi}_{uy} & I + \tilde{\boldsymbol{\Phi}}_{ux}B_2 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_x \\ \boldsymbol{\delta}_y \\ \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\beta \end{bmatrix}. \quad (5.11)$$

When  $A$  defines a stable system, the implementation in Figure 11(b) 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 (5.11), the effect of the perturbation  $\delta_\beta$  can lead to instability of the closed loop system. This structure thus shows the necessity of introducing and analyzing the effects of perturbations  $\delta_\beta$  on the controller internal state.

Alternatively, if we start with the identity  $\mathbf{K} = \Phi_{uy}(I + C_2\Phi_{xy})^{-1}$ , which also follows from (5.4), we obtain the controller structure shown in Figure 11(c). The closed loop map from perturbations to internal signals is then given by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \beta \end{bmatrix} = \begin{bmatrix} \Phi_{xx} & \Phi_{xy} & \Phi_{xx}B_2 \\ \Phi_{ux} & \Phi_{uy} & I + \Phi_{ux}B_2 \\ C_2(zI - A)^{-1} & I & C_2(zI - A)^{-1}B_2 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_u \end{bmatrix}.$$

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  $\beta$ .

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, we can show that the controller structure in Figure 11(b) is an alternative realization of the internal model control principle (IMC) [65, 66] as applied to the Youla parameterization. Specifically, for open loop stable systems, the Youla parameter is given by  $\mathbf{Q} = \mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}$ . As is shown in Lemma 5 of [54], the Youla parameter  $\mathbf{Q}$  is equal to the system response  $\Phi_{uy}$  for open loop stable systems. We then have

$$\mathbf{u} = \Phi_{uy}\mathbf{y} - \Phi_{ux}B_2\mathbf{u} \quad (5.12a)$$

$$= \mathbf{Q}\mathbf{y} - \Phi_{uy}C_2(zI - A)^{-1}B_2\mathbf{u} \quad (5.12b)$$

$$= \mathbf{Q}\mathbf{y} - \mathbf{Q}\mathbf{P}_{22}\mathbf{u} \quad (5.12c)$$

$$= \mathbf{Q}(\mathbf{y} - \mathbf{P}_{22}\mathbf{u}), \quad (5.12d)$$

where (5.12b) is obtained by substituting  $\Phi_{ux} = \Phi_{uy}C_2(zI - A)^{-1}$  from (5.5b) into (5.12a). Equation (5.12d) is exactly IMC. Thus, we see that IMC is equivalent to our proposed parameterization (and the simplified representation shown in Figure 11(b)) for open loop stable systems.

### 5.3 Output Feedback with $D_{22} \neq 0$

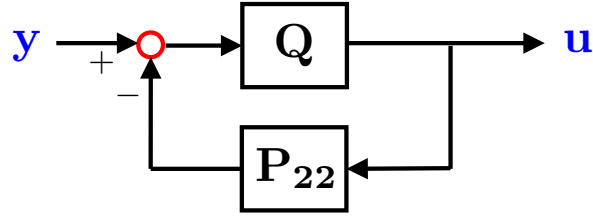
Finally, for a general proper plant model (3.1) with  $D_{22} \neq 0$ , we define a new measurement  $\bar{y}[t] = y[t] - D_{22}u[t]$ . This leads to the controller structure shown in Figure 12. In this case, the closed loop transfer matrices from  $\delta_u$  to the internal variables become

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \mathbf{y} \\ \beta \end{bmatrix} = \begin{bmatrix} \Phi_{xx}B_2 + \Phi_{xy}D_{22} \\ I + \Phi_{ux}B_2 + \Phi_{uy}D_{22} \\ C_2\Phi_{xx}B_2 + D_{22} + C_2\Phi_{xy}D_{22} \\ -\frac{1}{z}B_2(\Phi_{ux}B_2 + \Phi_{uy}D_{22}) \end{bmatrix} \delta_u.$$

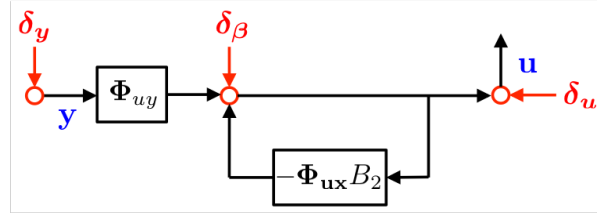
The remaining entries of Table 3 remain the same. Therefore, the controller structure shown in Figure 12 internally stabilizes the plant.

## System Level and Youla Parameterizations

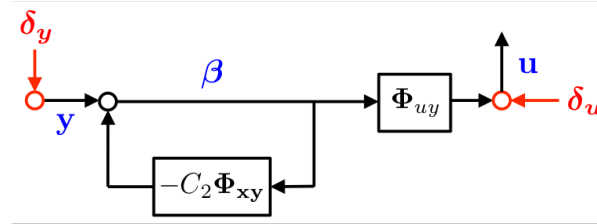
A key difference between the SL and Youla parameterizations is the manner in which they characterize the achievable closed loop responses of a system. The Youla parameterization provides



(a) Internal Model Control



(b) Structure 1



(c) Structure 2

Figure 11: Alternative controller structures for stable systems.

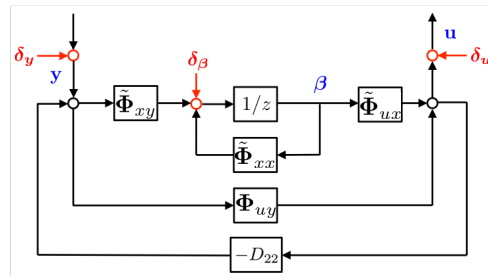


Figure 12: The proposed output feedback controller structure for  $D_{22} \neq 0$ .

an image space representation of the achievable system responses, parameterized explicitly by the free Youla parameter. This parameterization lends itself naturally to efficient computation via the standard and theoretically supported approach [3] of restricting the Youla parameter and objective function to be FIR. However, despite this ease of computation, as alluded to in Section 1.1, imposing sparsity constraints on the controller via the Youla parameter in general intractable.

In contrast, the proposed SL parameterization specifies a kernel space representation of achiev-

able system responses, parameterized implicitly by the affine space (5.5a) - (5.5b). While our discussion highlights the benefits and flexibility of the SL approach, there is the important caveat that the affine constraints (5.5a) - (5.5b) are in general infinite dimensional. Hence, although the parameterization is a convex one, it does not immediately lend itself to efficient computation. However, much as in the state-feedback setting, imposing (approximate) FIR constraints on the system responses leads to a finite-dimensional optimization problem.

## 5.4 Output Feedback System Level Synthesis

We conclude this subsection by formulating the general output feedback SLS problem. Proceeding as in Section 4, we use Theorem 5.1 to pose the following optimal control problem

$$\begin{aligned} \min_{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}} \quad & g(\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}) \\ \text{subject to} \quad & \text{constraints (5.5)} \\ & \Phi_x, \Phi_u \in \frac{1}{z} \mathcal{RH}_\infty \cap \mathcal{S}, \end{aligned} \tag{5.13}$$

where as before,  $g(\cdot)$  is an appropriately specified objective function, constraints (5.5) ensure achievability of the resulting system responses, and  $\mathcal{S}$  can be used to impose SLCs. A typical choice for the objective function  $g(\cdot)$  is

$$g(\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}) = \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \right\|, \tag{5.14}$$

for  $\|\cdot\|$  a suitably chosen system norm such as  $\mathcal{H}_2$ ,  $\mathcal{H}_\infty$ , or  $\mathcal{L}_1$ .

In the next subsection, we show how the imposition of localized SLCs once again leads to scalable controller synthesis in the distributed setting.

## 5.5 Distributed Control, Locality, and Scalability

In this section, we extend the separable objectives and constraints introduced previously to encompass a class of *partially* separable objectives and constraints, and show how these properties can be exploited to solve the general form of the output feedback SLS problem (5.13) when locality constraints are imposed in a scalable manner by using distributed optimization techniques such as ADMM.<sup>10</sup> We further show that many constrained optimal control problems are partially separable. Examples include the (localized)  $\mathcal{H}_2$  optimal control problem with joint sensor and actuator regularization and the (localized) mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem.

We begin by defining the partial separability of a generic optimization problem. We then specialize our discussion to partially separable instances of the SLS problem (5.13), before ending this section with a discussion of computational techniques that can be used to accelerate the ADMM algorithm as applied to solving partially separable SLS problems.

Before proceeding, we remark that although, as we show below, many SLOs and SLCs of interest satisfy our definition of partial separability, some classical measures of performance do not. For instance the  $\mathcal{H}_\infty$  norm of the closed loop response is not partially separable. This however hints at an interesting direction for future work: can suitable notions of robustness to  $\ell_2$  bounded disturbances be defined for the large-scale setting that are partially separable?

<sup>10</sup>Note that (5.13) is neither column-wise nor row-wise separable due to the coupling constraints (5.5a) and (5.5b).

## 5.6 Partially Separable Problems

We begin with a generic optimization problem given by

$$\underset{\Phi}{\text{minimize}} \quad g(\Phi) \tag{5.15a}$$

$$\text{subject to} \quad \Phi \in \mathcal{S}, \tag{5.15b}$$

where  $\Phi$  is a  $m \times n$  transfer matrix,  $g(\cdot)$  a functional objective, and  $\mathcal{S}$  a set constraint. We assume that the optimization problem (5.15) can be written in the form

$$\underset{\Phi}{\text{minimize}} \quad g^{(r)}(\Phi) + g^{(c)}(\Phi) \tag{5.16a}$$

$$\text{subject to} \quad \Phi \in \mathcal{S}^{(r)} \cap \mathcal{S}^{(c)}, \tag{5.16b}$$

where  $g^{(r)}$  and  $\mathcal{S}^{(r)}$  are row-wise separable<sup>11</sup> and  $g^{(c)}$  and  $\mathcal{S}^{(c)}$  are column-wise separable. If optimization problem (5.15) can be written in the form of problem (5.16), then optimization problem (5.15) is called *partially separable*.

Due to the coupling between the row-wise and column-wise separable components, optimization problem (5.16) admits neither a row-wise nor a column-wise decomposition. Our strategy is to instead use distributed optimization techniques such as ADMM to decouple the row-wise separable component and the column-wise separable components: we show that this leads to iterate subproblems that are column/row-wise separable, allowing the results of the previous section to be applied to solve the iterate subproblems using localized and parallel computation.

**Definition 2.** Let  $\{r_1, \dots, r_q\}$  be a partition of the set  $\{1, \dots, m\}$  and  $\{c_1, \dots, c_p\}$  be a partition of the set  $\{1, \dots, n\}$ . The functional objective  $g(\Phi)$  in (5.15a) is said to be partially separable with respect to the row-wise partition  $\{r_1, \dots, r_q\}$  and the column-wise partition  $\{c_1, \dots, c_p\}$  if  $g(\Phi)$  can be written as the sum of  $g^{(r)}(\Phi)$  and  $g^{(c)}(\Phi)$ , where  $g^{(r)}(\Phi)$  is row-wise separable with respect to the row-wise partition  $\{r_1, \dots, r_q\}$  and  $g^{(c)}(\Phi)$  is column-wise separable with respect to the column-wise partition  $\{c_1, \dots, c_p\}$ . Specifically, we have

$$\begin{aligned} g(\Phi) &= g^{(r)}(\Phi) + g^{(c)}(\Phi) \\ g^{(r)}(\Phi) &= \sum_{j=1}^q g_j^{(r)}(\Phi(r_j, :)) \\ g^{(c)}(\Phi) &= \sum_{j=1}^p g_j^{(c)}(\Phi(:, c_j)) \end{aligned} \tag{5.17}$$

for some functionals  $g_j^{(r)}(\cdot)$  for  $j = 1, \dots, q$ , and  $g_j^{(c)}(\cdot)$  for  $j = 1, \dots, p$ .

**Example 8.** Let  $\Phi$  be the optimization variable, and  $E$  and  $F$  be two matrices of compatible dimension. The objective function

$$\|E\Phi\|_{\mathcal{F}}^2 + \|\Phi F\|_{\mathcal{F}}^2$$

is then partially separable. Specifically, the first term is column-wise separable, and the second term is row-wise separable.

---

<sup>11</sup>Row-wise separability of functions and constraint sets is defined in the obvious manner, analogous to column-wise separability from Definition 5.



**Definition 3.** The set constraint  $\mathcal{S}$  in (5.15b) is said to be partially separable with respect to the row-wise partition  $\{r_1, \dots, r_q\}$  and the column-wise partition  $\{c_1, \dots, c_p\}$  if  $\mathcal{S}$  can be written as the intersection of two sets  $\mathcal{S}^{(r)}$  and  $\mathcal{S}^{(c)}$ , where  $\mathcal{S}^{(r)}$  is row-wise separable with respect to the row-wise partition  $\{r_1, \dots, r_q\}$  and  $\mathcal{S}^{(c)}$  is column-wise separable with respect to the column-wise partition  $\{c_1, \dots, c_p\}$ . Specifically, we have

$$\begin{aligned} \mathcal{S} &= \mathcal{S}^{(r)} \cap \mathcal{S}^{(c)} \\ \Phi \in \mathcal{S}^{(r)} &\iff \Phi(r_j, :) \in \mathcal{S}_j^{(r)}, \forall j \\ \Phi \in \mathcal{S}^{(c)} &\iff \Phi(:, c_j) \in \mathcal{S}_j^{(c)}, \forall j \end{aligned} \tag{5.18}$$

for some sets  $\mathcal{S}_j^{(r)}$  for  $j = 1, \dots, q$  and  $\mathcal{S}_j^{(c)}$  for  $j = 1, \dots, p$ .

**Example 9** (Affine Subspace). The affine subspace constraint

$$\mathbf{G}\Phi = \mathbf{H} \quad \text{and} \quad \Phi\mathbf{E} = \mathbf{F}$$

is partially separable. Specifically, the first constraint is column-wise separable, and the second is row-wise separable.

**Example 10** (Induced Matrix Norm). The induced 1-norm of a matrix  $\Phi$  is given by

$$\|\Phi\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |\Phi_{ij}|,$$

and the induced  $\infty$ -norm of a matrix is given by

$$\|\Phi\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |\Phi_{ij}|.$$

The following constraint is partially separable:

$$\|\Phi\|_1 \leq \gamma_1 \quad \text{and} \quad \|\Phi\|_\infty \leq \gamma_\infty.$$

Specifically, the first term is column-wise separable, and the second term is row-wise separable.

Assume now that the objective and the constraints in optimization problem (5.15) are both partially separable with respect to a row-wise partition  $\{r_1, \dots, r_q\}$  and a column-wise partition  $\{c_1, \dots, c_p\}$ . In this case, the optimization problem (5.15) is partially separable and can be rewritten in the form specified by equation (5.16). We now propose an ADMM based algorithm that exploits the partially separable structure of the optimization problem. As we show in the next subsection, when this method is applied to the SLS problem (5.13), the iterate subproblems are column/row-wise separable, allowing us to apply the methods described in Section 4.3. In particular, if locality constraints are imposed, iterate subproblems can be solved using localized and parallel computation.

Let  $\Psi$  be a duplicate of the optimization variable  $\Phi$ . We define the extended-real-value functionals  $h^{(r)}(\Phi)$  and  $h^{(c)}(\Psi)$  by

$$\begin{aligned} h^{(r)}(\Phi) &= \begin{cases} g^{(r)}(\Phi) & \text{if } \Phi \in \mathcal{S}^{(r)} \\ \infty & \text{otherwise} \end{cases} \\ h^{(c)}(\Psi) &= \begin{cases} g^{(c)}(\Psi) & \text{if } \Psi \in \mathcal{S}^{(c)} \\ \infty & \text{otherwise.} \end{cases} \end{aligned} \tag{5.19}$$

Problem (5.16) can then be reformulated as

$$\begin{aligned} & \underset{\{\Phi, \Psi\}}{\text{minimize}} && h^{(r)}(\Phi) + h^{(c)}(\Psi) \\ & \text{subject to} && \Phi = \Psi. \end{aligned} \tag{5.20}$$

Problem (5.20) can be solved using ADMM [67]:

$$\Phi^{k+1} = \underset{\Phi}{\text{argmin}} \left( h^{(r)}(\Phi) + \frac{\rho}{2} \|\Phi - \Psi^k + \Lambda^k\|_{\mathcal{H}_2}^2 \right) \tag{5.21a}$$

$$\Psi^{k+1} = \underset{\Psi}{\text{argmin}} \left( h^{(c)}(\Psi) + \frac{\rho}{2} \|\Psi - \Phi^{k+1} - \Lambda^k\|_{\mathcal{H}_2}^2 \right) \tag{5.21b}$$

$$\Lambda^{k+1} = \Lambda^k + \Phi^{k+1} - \Psi^{k+1} \tag{5.21c}$$

where the square of the  $\mathcal{H}_2$  norm is computed as in (4.14).

In Appendix B of [55], we provide stopping criterion and prove convergence of the ADMM algorithm (5.21a) - (5.21c) to an optimal solution of (5.16) (or equivalently, (5.20)) under the following assumptions.

**Assumption 3.** *Problem (5.16) has a feasible solution in the relative interior of the set  $\mathcal{S}$ .*

**Assumption 4.** *The functionals  $g^{(r)}(\cdot)$  and  $g^{(c)}(\cdot)$  are closed, proper, and convex.*

**Assumption 5.** *The sets  $\mathcal{S}^{(r)}$  and  $\mathcal{S}^{(c)}$  are closed and convex.*

We note that these assumptions are satisfied by typical SLCs that one may wish to apply. For instance, all of the examples considered in this paper satisfy these conditions so long as they specify a non-empty feasible set.

## 5.7 Partially Separable SLS Problems

We now specialize our discussion to the output feedback SLS problem (5.13). We use

$$\Phi = \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix}$$

to represent the system response we want to optimize for. Denote by  $\mathbf{Z}_{AB}$  the transfer matrix  $[zI - A \quad -B_2]$ , and by  $\mathbf{Z}_{AC}$  the transfer matrix  $[zI - A^\top \quad -C_2^\top]^\top$ . Let  $J_B$  be the matrix in the right-hand-side of (5.5a) and  $J_C$  be the matrix in the right-hand-side of (5.5b). The localized SLS problem (5.13) can then be written as

$$\underset{\Phi}{\text{minimize}} \quad g(\Phi) \tag{5.22a}$$

$$\text{subject to} \quad \mathbf{Z}_{AB}\Phi = J_B \tag{5.22b}$$

$$\Phi\mathbf{Z}_{AC} = J_C \tag{5.22c}$$

$$\Phi \in \mathcal{S} \tag{5.22d}$$

with  $\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}$ . Note that as already discussed, the affine subspace constraints (5.22b) and (5.22c) are partially separable with respect to arbitrary column-wise and row-wise partitions, respectively. Thus the SLS problem (5.22) is partially separable if the SLO (5.22a) and the SLC  $\Phi \in \mathcal{X}$  are partially separable.

In particular, 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 locality constraint and the FIR constraint are included in both  $\mathcal{S}^{(r)}$  and  $\mathcal{S}^{(c)}$ . This is the key point to allow the subroutines (5.21a) - (5.21b) of the ADMM algorithm to be solved using the techniques described in Section 4.3.

To illustrate how the iterate subproblems of the ADMM algorithm (5.21), as applied to the SLS problem (5.22), can be solved using localized and parallel computation, we analyze in more detail the iterate subproblem (5.21b), which is given by

$$\underset{\Psi}{\text{minimize}} \quad g^{(c)}(\Psi) + \frac{\rho}{2} \|\Psi - \Phi^{k+1} - \Lambda^k\|_{\mathcal{H}_2}^2 \quad (5.23a)$$

$$\text{subject to} \quad \Psi \in \mathcal{S}^{(c)} \quad (5.23b)$$

with  $\mathcal{S}^{(c)} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}^{(c)}$ . The  $\mathcal{H}_2$  norm regularizer in (5.23a) 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 (5.23) into parallel subproblems. Then, as in the state-feedback setting, we can then exploit the locality constraint  $\mathcal{L}$  to reduce the dimension of each subproblem from global to local scale. Similarly, iterate subproblem (5.21a) can also be solved using parallel and localized computation by exploiting its row-wise separability. Finally, update equation (5.21c) trivially decomposes element-wise since it is a matrix addition.

We now give some examples of partially separable SLOs.

**Example 11** (Norm Optimal Control). *Consider the SLO of the optimal control problem in (5.13) with objective function as in (5.14) and norm given by the square of the  $\mathcal{H}_2$  norm. Suppose that there exists a permutation matrix  $\Pi$  such that the matrix  $[B_1^\top \ D_{21}^\top] \Pi$  is block diagonal. It is then possible to find a column-wise partition to decompose the SLO in a column-wise manner. Similarly, suppose that there exists a permutation matrix  $\Pi$  such that the matrix  $[C_1 \ D_{12}] \Pi$  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 12** (Sensor and Actuator Norm). *Consider the weighted actuator norm defined in [68, 47], which is given by*

$$\|\mu [\Phi_{ux} \ \Phi_{uy}]\|_{\mathcal{U}} = \sum_{i=1}^{n_u} \mu_i \|e_i^\top [\Phi_{ux} \ \Phi_{uy}]\|_{\mathcal{H}_2} \quad (5.24)$$

where  $\mu$  is a diagonal matrix with  $\mu_i \geq 0$  being its  $i^{\text{th}}$  diagonal entry, and  $e_i$  is a unit vector  $i^{\text{th}}$  entry set to 1, and 0 elsewhere. The actuator norm (5.24) can be viewed as an infinite dimensional analog to the weighted  $\ell_1/\ell_2$  norm, also known as the group lasso [69] in the statistical learning literature. Adding this norm as a regularizer to a SLS problem induces row-wise sparse structure in the transfer matrix  $[\Phi_{ux} \ \Phi_{uy}]$ . Recall from Theorem 5.1 that the controller achieving the desired

system response can be implemented via (5.9). If the  $i$ th row of the transfer matrix  $\begin{bmatrix} \Phi_{ux} & \Phi_{uy} \end{bmatrix}$  is identically zero, then the  $i$ th component of the control action  $u_i$  is always equal to zero, and therefore the actuator at node  $i$  corresponding to control action  $u_i$  can be removed without changing the closed loop response. It is clear that the actuator norm defined in (5.24) is row-wise separable with respect to arbitrary row-wise partition. This still holds true when the actuator norm is defined by the  $\ell_1/\ell_\infty$  norm. Similarly, consider the weighted sensor norm given by

$$\left\| \begin{bmatrix} \Phi_{xy} \\ \Phi_{uy} \end{bmatrix} \lambda \right\|_{\mathcal{Y}} = \sum_{i=1}^{n_y} \lambda_i \left\| \begin{bmatrix} \Phi_{xy} \\ \Phi_{uy} \end{bmatrix} e_i \right\|_{\mathcal{H}_2} \quad (5.25)$$

where  $\lambda$  is a diagonal matrix with  $\lambda_i \geq 0$  being its  $i$ th diagonal entry. The sensor norm (5.25), when added as a regularizer, induces column-wise sparsity in the transfer matrix  $\begin{bmatrix} \Phi_{xy}^\top & \Phi_{uy}^\top \end{bmatrix}^\top$ . Using the controller implementation (5.9), the sensor norm can therefore be viewed as regularizing the number of sensors used by a controller. For instance, if the  $i$ th column of the transfer matrix  $\begin{bmatrix} \Phi_{xy}^\top & \Phi_{uy}^\top \end{bmatrix}^\top$  is identically zero, then the sensor at node  $i$  and its corresponding measurement  $y_i$  can be removed without changing the closed loop response. The sensor norm defined in (5.24) is column-wise separable with respect to any column-wise partition.

**Example 13** (Combination). From Definition 2, it is straightforward to see that the class of partially separable SLOs with respect to the same partitions are closed under summation. Therefore, we can combine the partially separable SLOs described above, and the resulting SLO is still partially separable. For instance, consider the SLO given by

$$\begin{aligned} g(\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}) &= \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \right\|_{\mathcal{H}_2}^2 \\ &+ \left\| \mu \begin{bmatrix} \Phi_{ux} & \Phi_{uy} \end{bmatrix} \right\|_{\mathcal{U}} + \left\| \begin{bmatrix} \Phi_{xy} \\ \Phi_{uy} \end{bmatrix} \lambda \right\|_{\mathcal{Y}} \end{aligned} \quad (5.26)$$

where  $\mu$  and  $\lambda$  are non-negative diagonal matrices specifying the relative weighting between the closed loop  $\mathcal{H}_2$  performance, and the actuator and sensor norm penalties, respectively. If there exists a permutation matrix  $\Pi$  such that the matrix  $\begin{bmatrix} C_1 & D_{12} \end{bmatrix} \Pi$  is block diagonal, then the SLO (5.26) 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.<sup>12</sup>

We now provide some examples of partially separable SLCs.

**Example 14** ( $\mathcal{L}_1$  Constraint). The  $\mathcal{L}_1$  norm of a transfer matrix is given by its worst case  $\ell_\infty$  to  $\ell_\infty$  gain. In particular, the  $\mathcal{L}_1$  norm [70] of a FIR transfer matrix  $\mathbf{G} \in \mathcal{F}_T$  is given by

$$\|\mathbf{G}\|_{\mathcal{L}_1} = \max_i \sum_j \sum_{t=0}^T |g_{ij}[t]|. \quad (5.27)$$

We can therefore add the constraint

$$\left\| \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \right\|_{\mathcal{L}_1} \leq \gamma \quad (5.28)$$

<sup>12</sup>Note that an alternative penalty is proposed in [68] for the design of joint actuation and sensing architectures; it is however more involved to define, and hence we restrict ourself to this alternative penalty for the purposes of illustrating the concept of partially separable SLOs.

to the optimization problem (5.22) for some  $\gamma$  to control the worst-case amplification of  $\ell_\infty$  bounded signals. From the definition (5.27), the SLC (5.28) is row-wise separable with respect to any row-wise partition.

**Example 15** (Combination). From Definition 3, the class of partially separable SLCs with respect to the same row and column partitions are closed under intersection, allowing for partially separable SLCs to be combined. For instance, the combination of a locality constraint  $\mathcal{L}$ , a FIR constraint  $\mathcal{F}_T$ , and an  $\mathcal{L}_1$  constraint as in equation (5.28) is partially separable. This property is extremely useful as it provides a unified framework for dealing with several partially separable constraints at once.

Using the previously described examples of partially separable SLOs and SLCs, we now consider two partially separable SLS problems: (i) the localized  $\mathcal{H}_2$  optimal control problem with joint sensor and actuator regularization, and (ii) the localized mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem. These two problems are used in Section 5.9 as case study examples.

**Example 16.** The localized  $\mathcal{H}_2$  optimal control problem with joint sensor and actuator regularization is given by

$$\underset{\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}}{\text{minimize}} \quad (5.26) \quad (5.29a)$$

$$\text{subject to} \quad (5.5a) - (5.5c) \quad (5.29b)$$

$$\begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T. \quad (5.29c)$$

where  $\mathcal{C}$  encodes the information sharing constraints of the distributed controller. If there exists a permutation matrix  $\Pi$  such that the matrix  $[C_1 \ D_{12}] \Pi$  is block diagonal, then (5.29) is partially separable.

**Remark 5.5.** When the actuator and sensor norm penalties are zero, problem (5.29) reduces to the localized LQG optimal control problem defined and solved in [46]. Further, if the locality and FIR constraints are removed, we recover the standard LQG optimal control problem, which is also seen to be partially separable.

Next we consider the localized mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem.

**Example 17.** The localized mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem is given by

$$\underset{\{\Phi_{xx}, \Phi_{ux}, \Phi_{xy}, \Phi_{uy}\}}{\text{minimize}} \quad \left\| \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \right\|_{\mathcal{H}_2}^2 \quad (5.30a)$$

$$\text{subject to} \quad (5.5a) - (5.5c), (5.28), (5.29c) \quad (5.30b)$$

which is partially separable.

The localized mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem can be used to design the tradeoff between average and worst-case performance, as measured by the  $\mathcal{H}_2$  and  $\mathcal{L}_1$  norms of the closed loop system, respectively.

## 5.8 Analytic Solution and Acceleration

Suppose that the Assumptions 3 - 5 in Section 5.6 hold. The ADMM algorithm presented in (5.21) is a special case of the proximal algorithm [71, 72, 67]. For certain type of objective functionals  $h^{(r)}(\cdot)$  and  $h^{(c)}(\cdot)$ , the proximal operators can be evaluated analytically (see Ch. 6 of [72]). In this situation, we only need to evaluate the proximal operators *once*, and iterate (5.21a) and (5.21b) in closed form. This improves the overall computation time significantly. In Appendix C of [55], it is shown how to express the solutions of (5.21a) and (5.21b) using proximal operators. Here we list a few examples for which the proximal operators can be evaluated analytically.

**Example 18.** *Consider the LLQG problem proposed in [46]. When the global optimization problem is decomposed into parallel subproblems, each subproblem is a convex quadratic program restricted to an affine set. In this case, the proximal operator is an affine function [46, 72, 67]. We only need to calculate this affine function once. The iteration in (5.21a) - (5.21c) can then be carried out using multiple matrix multiplications in the reduced dimension, which significantly improves the overall computation time.*

**Example 19.** *Consider the LLQR problem with actuator regularization [47], which is the state feedback version of (5.29). The column-wise separable component of this problem is identical to that of the LLQG example, and therefore the update subproblem (5.21b) can be solved using matrix multiplication, as described in the previous example. We further showed in [47] that the row-wise separable component of this problem can be simplified to multiple unconstrained optimization problems, each with proximal operators given by vectorial soft-thresholding [72]. This offers an efficient way to solve the update subproblem (5.21a).*

We end this section by noting that although our focus has been on how ADMM can be used to efficiently solve the localized SLS problem (5.13), there exist other distributed algorithms that can exploit partial separability to improve computational efficiency. For instance, if either  $g^{(r)}(\cdot)$  or  $g^{(c)}(\cdot)$  is strongly convex, we can use the alternating minimization algorithm (AMA) [73] to simplify the ADMM algorithm.

## 5.9 Case Studies

In this section, we apply the localized  $\mathcal{H}_2$  optimal control with sensor actuator regularization (5.29) problem and the localized mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem (5.30) to a power system inspired example. After introducing the power system model, we show that the localized  $\mathcal{H}_2$  controller, with its additional locality, FIR and communication delay constraints can achieve comparable closed loop performance to that of a centralized  $\mathcal{H}_2$  optimal controller. We further demonstrate the scalability of the proposed method by synthesizing a localized  $\mathcal{H}_2$  controller for a randomized heterogeneous networked system with 12800 states using a single laptop; for such a large system, neither the centralized nor the distributed optimal controllers can be computed. We then solve the localized  $\mathcal{H}_2$  optimal control problem with joint sensor and actuator regularization to co-design an output feedback controller and its actuation/sensing architecture. Finally, we solve the localized mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem, thus identifying the achievable tradeoff curve between the average and worst-case performance of the closed loop system.

## 5.10 Power System Model

We begin with a randomized spanning tree embedded on a  $10 \times 10$  mesh network representing the interconnection between subsystems. The resulting interconnected topology is shown in Fig. 13(a) – we assume that all edges are undirected. The dynamics of each subsystem is given by the

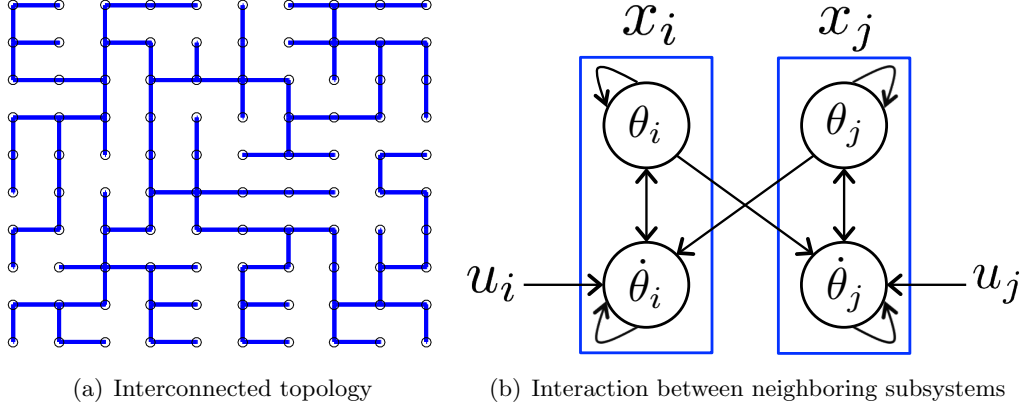


Figure 13: Simulation example interaction graph.

discretized swing equations. Consider the swing dynamics

$$m_i \ddot{\theta}_i + d_i \dot{\theta}_i = - \sum_{j \in \mathcal{N}_i} k_{ij} (\theta_i - \theta_j) + w_i + u_i \quad (5.31)$$

where  $\theta_i$ ,  $\dot{\theta}_i$ ,  $m_i$ ,  $d_i$ ,  $w_i$ ,  $u_i$  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  $x_i := [\theta_i \ \dot{\theta}_i]^\top$  be the state of bus  $i$  and use  $e^{A\Delta t} \approx I + A\Delta t$  to discretize the swing dynamics. Equation (5.31) can then be expressed in the form

$$\begin{aligned} x_i[t+1] &= A_{ii}x_i[t] + \sum_{j \in \mathcal{N}_i} A_{ij}x_j[t] + B_{ii}u_i[t] + \delta_{x_i}[t] \\ y_i[t] &= C_{ii}x_i[t] + \delta_{y_i}[t], \end{aligned} \quad (5.32)$$

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  $B_{ii} = [0 \ 1]^\top$ . 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 interaction between neighboring subsystems of the discretized model is illustrated in Figure 13(b). We initially 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 (5.31), the external disturbance  $w_i$  only directly affects the frequency deviation  $\dot{\theta}_i$ . To make the objective functional strongly convex, we introduce a small artificial disturbance on the

phase deviation  $\theta_i$  as well. We assume that the process noise on frequency and phase are uncorrelated and independently and identically distributed (i.i.d.) Gaussian random variables with zero mean, and 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 i.i.d. Gaussian random variables with zero mean, and covariance matrices both given by  $10^{-2}I$ . We choose equal penalty on the state deviation and control effort, i.e.,  $[C_1 \ D_{12}] = I$ .

Based on the above setting, we formulate a  $\mathcal{H}_2$  optimal control 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 normalize the  $\mathcal{H}_2$  norm with respect to the proper centralized  $\mathcal{H}_2$  optimal controller.

### 5.11 Localized $\mathcal{H}_2$ Optimal Control

The underlying assumption of the centralized optimal control scheme is that measurements can be instantaneously transmitted among all subsystems in the network. To incorporate realistic communication delay constraints and facilitate the scalability of controller design, we impose additional communication delay constraints, locality constraints, and a FIR constraint on the system response.

For the communication delay constraint  $\mathcal{C}$ , we assume that each subsystem takes one time step to transmit information to its neighboring subsystems. Therefore, if subsystems  $i$  and  $j$  are  $k$ -hops apart (as defined by the interaction graph illustrated in Figure 13(a)), then the control action  $u_i[t]$  at subsystem  $i$  can only use the measurements  $y_j[\tau]$  and internal controller state  $\beta_j[\tau]$  of subsystem  $j$  if  $\tau \leq t - k$ .

The interaction between subsystems illustrated in Fig. 13(b) implies that it takes two discrete time steps for a disturbance at subsystem  $j$  to propagate to its neighboring subsystems, and hence the communication speed between sub-controllers is twice as fast as the propagation speed of disturbances through the plant. For the given communication delay constraint  $\mathcal{C}$ , we use the method in [47] to design the tightest feasible locality constraint  $\mathcal{L}$ . In this example, we can localize the joint effect of the process and sensor noise at subsystem  $j$  to a region defined by its two-hop neighbors (where one hop is as defined in terms of the interaction graph of the system). This implies that the sub-controller at node  $j$  only needs to transmit its measurements  $y_j$  and controller states  $\beta_j$  within this localized region, and further only a restricted plant model (as defined by this localized region) is needed to synthesize its corresponding local control policy.

Assuming a fixed communication delay constraint  $\mathcal{C}$  and locality constraint  $\mathcal{L}$ , we first explore the tradeoff between the length  $T$  of the FIR constraint  $\mathcal{F}_T$  and the transient performance of the closed loop system. Figure 14 shows the tradeoff curve between the transient performance of the localized  $\mathcal{H}_2$  controller and the length  $T$  of the FIR constraint. For the given communication delay constraint  $\mathcal{C}$  and the locality constraint  $\mathcal{L}$ , the localized  $\mathcal{H}_2$  controller is feasible for FIR constraints  $\mathcal{F}_T$  whenever  $T \geq 3$ . 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 lengths of  $T = 7, 10,$  and  $20$ , the performance degradations of the localized FIR controllers with respect 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 locality constraint  $\mathcal{L}$  is less than 0.1%. From Figure 14, we see that the



localized  $\mathcal{H}_2$  controller, with its additional communication delay, locality and FIR constraints can achieve similar transient performance to that of an unconstrained centralized (unimplementable) optimal  $\mathcal{H}_2$  controller.

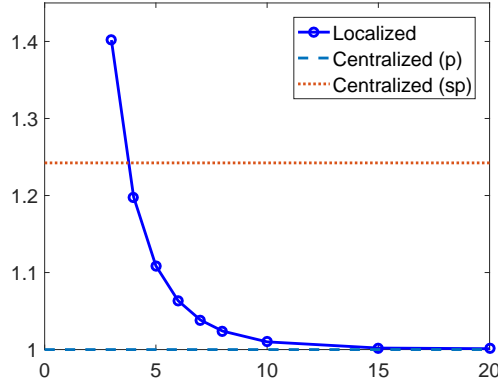


Figure 14: The vertical axis is the normalized  $\mathcal{H}_2$  norm of the closed loop when the localized  $\mathcal{H}_2$  controller is applied. The localized  $\mathcal{H}_2$  controller is subject to the constraint  $\mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T$ . The horizontal axis is the horizon  $T$  of the FIR constraint  $\mathcal{F}_T$ , which is also the settling time of an impulse disturbance. 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 localized control scheme, we choose  $T = 20$  and compare the localized optimal controller, distributed optimal controller, and the centralized optimal controller in terms of closed loop performance, complexity of controller synthesis and implementation, as well as ease of redesign,<sup>13</sup> in Table 4. The distributed optimal controller is computed using the method described in [74], in which we assume the same communication constraint  $\mathcal{C}$  as the localized optimal controller. It can be seen that the localized controller is vastly preferable in all aspects, except for a slight degradation in the closed-loop performance.

Table 4: Comparison Between Centralized, Distributed, and Localized Control

		Cent.	Dist.	Local.
Closed Loop	Affected region	Global	Global	2-hop
	Affected time	Long	Long	20 steps
	Normalized $\mathcal{H}_2$	1	1.001	1.001
Synthesis	Complexity	$O(n^3)$	$\geq O(n^3)$	$O(n)$
	Plant model	Global	Global	2-hop
	Redesign	Global	Global	Local
Implement.	Comm. Speed	$\infty$	2	2
	Comm. Range	Global	Global	2-hop

We now allow the size of the problem to vary and compare the computation time needed to

<sup>13</sup>The ability to locally redesign subsets of the control system has the appealing feature of allowing for real-time redesign in the case of local changes to the plant model.

synthesize a centralized, distributed, and localized  $\mathcal{H}_2$  optimal controller. We choose  $T = 7$  for the localized controller. The empirical relationship obtained between computation time and problem size for different control schemes is illustrated in Figure 5. It is easily seen that the computation time needed for the distributed controller grows rapidly when the size of problem increases. The slope of the line describing the computation time of the centralized controller in the log-log plot of Figure 5 is 3, which matches the theoretical complexity  $O(n^3)$ . The slope for the localized  $\mathcal{H}_2$  controller is about 1.4, which is larger than the theoretical value of 1. As mentioned previously, we believe this overhead may be caused by other computational issues, such as memory management. We note that the computational bottleneck faced in computing our large-scale example arises from using a single laptop to compute the controller (and hence the localized subproblems were 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 that we have, we can compute the optimal localized  $\mathcal{H}_2$  controller for a system with 12800 states in 22 minutes using MATLAB with an ASUS laptop with Intel i7-3610QM @ 2.3GHz CPU and 8 GB of RAM. If the computation is parallelized across all 6400 sub-systems, the synthesis algorithm can be completed within 0.2 seconds. In contrast, the theoretical time to compute the centralized  $\mathcal{H}_2$  optimal controller for the same example is more than a week.

## 5.12 Localized $\mathcal{H}_2$ Optimal Control with Joint Sensor and Actuator Regularization

We now move back to the  $10 \times 10$  mesh example shown in Figure 13(a). In the previous subsection, we assumed that each subsystem in the power network had a phase sensor, a frequency sensor, and a controllable load. In practice, the installation of these sensors and actuators is expensive: therefore we explore the tradeoff between the closed loop performance of the system 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 the combinatorial nature of this task. In this subsection, we apply the regularization for design (RFD) [68] 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 problem with joint sensor and actuator regularization (5.29).

In order to allow for more flexibility in sensor and 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 information with up to its four-hop neighbors, and use the plant model restricted to this four-hop neighborhood to synthesize the localized  $\mathcal{H}_2$  controller. Similarly, the length of the FIR constraint  $\mathcal{F}_T$  is increased to  $T = 30$ . The initial localized  $\mathcal{H}_2$  cost is given by 13.3210, which is a 0.03% degradation with respect to the optimal centralized  $\mathcal{H}_2$  controller. We assume that the relative prices 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 that PMUs are typically more expensive than frequency sensors. The price for the same types of sensors and actuators at different locations remains constant.

We begin with a dense controller architecture composed of 100 controllable loads, 100 PMUs, and 100 frequency sensors, i.e., one of each type of sensor and actuator at each node. Using optimization problem (5.29) to identify the sparsest possible architecture that still satisfies the locality, FIR and communication delay constraints, we are able to remove 43 controllable loads and

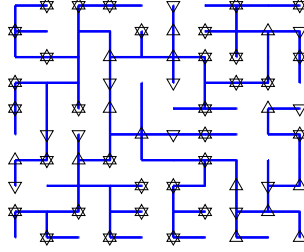


Figure 15: The upward-pointing triangles represent the subsystems in which the PMU is removed. The downward-pointing triangles represent the subsystems in which the controllable load (actuator) is removed.

46 PMUs from the system (no frequency sensors were removed due to the chosen relative pricing). The locations of the removed sensors and actuators are shown in Figure 15. We argue that this sensing and actuation interface 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 to all process and sensor disturbances remains both localized and FIR.

The localized  $\mathcal{H}_2$  cost for the system with reduced number of sensors and actuators is given by 17.8620. In comparison, the cost achieved by a proper centralized optimal controller is 16.2280, and the cost achieved by a strictly proper centralized optimal controller is 18.4707. Note that as the sensing and actuation interface becomes sparser, the performance gap between the centralized and the localized controller becomes larger. Nevertheless, we note that the performance degradation is only 10% compared to the proper centralized optimal scheme implemented using the same sparse controller architecture.

### 5.13 Localized Mixed $\mathcal{H}_2/\mathcal{L}_1$ Optimal Control

Finally, we solve the localized mixed  $\mathcal{H}_2/\mathcal{L}_1$  optimal control problem in (5.30) on the  $10 \times 10$  mesh example shown in Figure 13(a). We progressively reduce the allowable  $\mathcal{L}_1$  gain  $\gamma$ , as shown in equation (5.30), to explore the tradeoff between average and worst-case performance of a system. We plot the normalized  $\mathcal{H}_2$  norm and the normalized  $\mathcal{L}_1$  norm in Figure 16.<sup>14</sup> The left-top point in Figure 16 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, thus tracing out a tradeoff curve.

### 5.14 Summary

In this section, we extended many of the state-feedback results from Section 4 to the output feedback setting. In particular, we showed that by exploiting the partial separability of many SLS problems of interest, output feedback problems also enjoy improved the scalability via a decomposition and dimensionality reduction based method combined with distributed optimization methods such as ADMM. Lacking in the output feedback setting however is an analogous family of robustness results to those presented in Section 4.5. Although preliminary results exist [22], this remains in important direction for future work.

<sup>14</sup>We normalize the  $\mathcal{H}_2$  performance with respect to that achieved by the optimal localized  $\mathcal{H}_2$  controller, and likewise normalize the  $\mathcal{L}_1$  performance with respect to that achieved by the optimal localized  $\mathcal{L}_1$  controller.

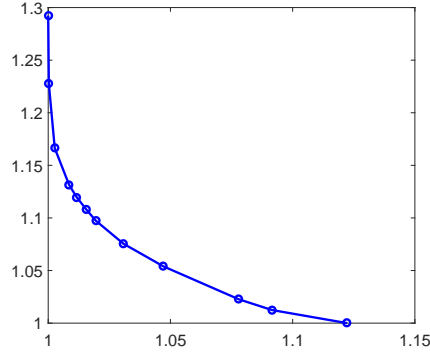


Figure 16: The vertical axis represents the normalized  $\mathcal{L}_1$  norm of the closed loop, and the horizontal axis represents the normalized  $\mathcal{H}_2$  norm of the closed loop.

## 6 Conclusions

This article reviewed the System Level Synthesis framework. Our aim was to provide a self-contained and accessible resource that summarizes progress in developing this novel approach to controller synthesis. We highlighted the benefits that SLS provides in the context of distributed control and in robust control. As we hope we were able to show, by working directly with system responses, there is a transparency in how system constraints, structure, and uncertainty affect controller synthesis, implementation, and performance, and it is this transparency that we exploit throughout to improve upon the state-of-the-art. As lengthy as this article is, we believe that we have just begun to scratch the surface of what SLS can do. Exciting current directions of research focus upon integrating SLS into model predictive control algorithms, further understanding the algebraic structure underlying localized controllers and their state-space realizations, further developing robust SLS theory (especially in the output feedback setting), and applying these tools to application areas spanning power-systems, computer networking, and machine learning/AI.

## References

- [1] D. C. Youla, H. A. Jabr, and J. J. B. Jr., “Modern wiener-hopf design of optimal controllers-part ii: The multivariable case,” *Automatic Control, IEEE Transactions on*, vol. 21, no. 3, pp. 319–338, 1976.
- [2] M. Vidyasagar, *Control System Synthesis: A Factorization Approach, Part II*. Morgan & Claypool, 2011.
- [3] S. Boyd and C. Barratt, *Linear controller design: limits of performance*. Prentice-Hall, 1991.
- [4] M. A. Dahleh and I. J. Diaz-Bobillo, *Control of uncertain systems: a linear programming approach*. Prentice-Hall, Inc., 1994.
- [5] J. Y. Ishihara and R. M. Sales, “Parametrization of admissible controllers for generalized rosenbrock systems,” in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 5. IEEE, 2000, pp. 5014–5019.

- [6] H. H. Rosenbrock and D. Owens, “Computer aided control system design,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, no. 11, pp. 794–794, 1976.
- [7] J. C. Willems and J. W. Polderman, *Introduction to mathematical systems theory: a behavioral approach*. Springer Science & Business Media, 2013, vol. 26.
- [8] J. C. Willems and H. L. Trentelman, “Synthesis of dissipative systems using quadratic differential forms: Part i,” *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 53–69, 2002.
- [9] H. L. Trentelman and J. C. Willems, “Synthesis of dissipative systems using quadratic differential forms: Part ii,” *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 70–86, 2002.
- [10] C. Praagman, H. L. Trentelman, and R. Z. Yoe, “On the parametrization of all regularly implementing and stabilizing controllers,” *SIAM Journal on Control and Optimization*, vol. 45, no. 6, pp. 2035–2053, 2007.
- [11] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, “State-space solutions to standard  $H_2$  and  $H_\infty$  control problems,” *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, Aug 1989.
- [12] M. A. Dahleh and I. J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach*, 1994.
- [13] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, 1995.
- [14] J. Doyle, “Analysis of feedback systems with structured uncertainties,” *IEE Proceedings D - Control Theory and Applications*, vol. 129, no. 6, 1982.
- [15] M. K. H. Fan, A. L. Tits, and J. C. Doyle, “Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics,” *IEEE Transactions on Automatic Control*, vol. 36, no. 1, 1991.
- [16] A. Packard and J. Doyle, “The Complex Structured Singular Value,” *Automatica*, vol. 29, no. 1, 1993.
- [17] P. M. Young, M. P. Newlin, and J. C. Doyle, “ $\mu$  Analysis with Real Parametric Uncertainty,” in *IEEE Conference on Decision and Control*, 1991.
- [18] A. Megretski and A. Rantzer, “System analysis via integral quadratic constraints,” *IEEE Transactions on Automatic Control*, vol. 42, no. 6, 1997.
- [19] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, “On the sample complexity of the linear quadratic regulator,” *arXiv preprint arXiv:1710.01688*, 2017.
- [20] S. Dean, S. Tu, N. Matni, and B. Recht, “Safely Learning to Control the Constrained Linear Quadratic Regulator.” *arXiv:1809.10121*, 2018.
- [21] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, “Regret Bounds for Robust Adaptive Control of the Linear Quadratic Regulator,” in *Neural Information Processing Systems*, 2018.

- [22] R. Boczar, N. Matni, and B. Recht, “Finite-data performance guarantees for the output-feedback control of an unknown system,” *arXiv preprint arXiv:1803.09186*, 2018.
- [23] Y.-C. Ho and K.-C. Chu, “Team decision theory and information structures in optimal control problems—part i,” *Automatic Control, IEEE Transactions on*, vol. 17, no. 1, pp. 15–22, 1972.
- [24] A. Mahajan, N. Martins, M. Rotkowitz, and S. Yuksel, “Information structures in optimal decentralized control,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, pp. 1291–1306.
- [25] M. Rotkowitz and S. Lall, “A characterization of convex problems in decentralized control,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 2, pp. 274–286, 2006.
- [26] B. Bamieh, F. Paganini, and M. A. Dahleh, “Distributed control of spatially invariant systems,” *Automatic Control, IEEE Transactions on*, vol. 47, no. 7, pp. 1091–1107, 2002.
- [27] B. Bamieh and P. G. Voulgaris, “A convex characterization of distributed control problems in spatially invariant systems with communication constraints,” *Systems & Control Letters*, vol. 54, no. 6, pp. 575–583, 2005.
- [28] A. Nayyar, A. Mahajan, and D. Teneketzis, “Decentralized stochastic control with partial history sharing: A common information approach,” *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1644–1658, July 2013.
- [29] H. S. Witsenhausen, “A counterexample in stochastic optimum control,” *SIAM Journal of Control*, vol. 6, no. 1, 1968.
- [30] J. N. Tsitsiklis and M. Athans, “On the complexity of decentralized decision making and detection problems,” in *IEEE Conference on Decision and Control (CDC)*, 1984.
- [31] G. E. Dullerud and R. D’Andrea, “Distributed control of heterogeneous systems,” *Automatic Control, IEEE Transactions on*, vol. 49, no. 12, pp. 2113–2128, 2004.
- [32] M. Rotkowitz, R. Cogill, and S. Lall, “Convexity of optimal control over networks with delays and arbitrary topology,” *Int. J. Syst., Control Commun.*, vol. 2, no. 1/2/3, pp. 30–54, Jan. 2010.
- [33] L. Lessard and S. Lall, “Convexity of decentralized controller synthesis,” *IEEE Transactions on Automatic Control*, To appear, 2016. [Online]. Available: <http://arxiv.org/pdf/1305.5859v2.pdf>
- [34] —, “Optimal controller synthesis for the decentralized two-player problem with output feedback,” in *2012 IEEE American Control Conference*, June 2012.
- [35] P. Shah and P. 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.
- [36] A. Lamperski and J. C. Doyle, “Output feedback  $\mathcal{H}_2$  model matching for decentralized systems with delays,” in *2013 IEEE American Control Conference*, June 2013.

- [37] L. Lessard, M. Krystalny, and A. Rantzer, “On structured realizability and stabilizability of linear systems,” in *American Control Conference*, 2013, June 2013, pp. 5784–5790.
- [38] C. W. Scherer, “Structured  $\mathcal{H}_\infty$ -optimal control for nested interconnections: A state-space solution,” *Systems and Control Letters*, vol. 62, pp. 1105–1113, 2013.
- [39] L. Lessard, “State-space solution to a minimum-entropy  $\mathcal{H}_\infty$ -optimal control problem with a nested information constraint,” in *2014 IEEE Conference on Decision and Control (CDC)*, 2014.
- [40] N. Matni, “Distributed control subject to delays satisfying an  $\mathcal{H}_\infty$  norm bound,” in *2014 IEEE Conference on Decision and Control (CDC)*, 2014.
- [41] T. Tanaka and P. A. Parrilo, “Optimal output feedback architecture for triangular LQG problems,” in *2014 IEEE American Control Conference*, June 2014.
- [42] A. Lamperski and L. Lessard, “Optimal decentralized state-feedback control with sparsity and delays,” *Automatica*, vol. 58, pp. 143–151, 2015.
- [43] Y.-S. Wang, N. Matni, S. You, and J. C. Doyle, “Localized distributed state feedback control with communication delays,” in *2014 IEEE American Control Conference*, June 2014.
- [44] Y.-S. Wang, N. Matni, and J. C. Doyle, “Localized LQR optimal control,” in *2014 53rd IEEE Conference on Decision and Control (CDC)*, 2014.
- [45] Y.-S. Wang and N. Matni, “Localized distributed optimal control with output feedback and communication delays,” in *IEEE 52nd Annual Allerton Conference on Communication, Control, and Computing*, 2014.
- [46] —, “Localized LQG optimal control for large-scale systems,” in *2016 IEEE American Control Conference*, 2016.
- [47] Y.-S. Wang, N. Matni, and J. C. Doyle, “Localized LQR control with actuator regularization,” in *2016 IEEE American Control Conference*, 2016.
- [48] Y.-S. Wang, S. You, and N. Matni, “Localized distributed Kalman filters for large-scale systems,” in *5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2015.
- [49] Y. S. Wang, N. Matni, and J. C. Doyle, “System level parameterizations, constraints and synthesis,” in *2017 American Control Conference*, May 2017, pp. 1308–1315.
- [50] N. Matni, Y.-S. Wang, and J. Anderson, “Scalable system level synthesis for virtually localizable systems,” in *2017 56th IEEE Conf. Decision Control*, 2017.
- [51] J. C. Doyle, N. Matni, Y.-S. Wang, J. Anderson, and S. Low, “System level synthesis: A tutorial,” in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 2856–2867.
- [52] Y.-S. Wang, “A system level approach to optimal controller design for large-scale distributed systems,” Ph.D. dissertation, California Institute of Technology, 2016.

- [53] J. Anderson and N. Matni, “Structured State Space Realizations for SLS Distributed Controllers,” in *Allerton*, 2017.
- [54] Y.-S. Wang, N. Matni, and J. C. Doyle, “A system level approach to controller synthesis,” *IEEE Transactions on Automatic Control*, 2019.
- [55] —, “Separable and localized system level synthesis for large-scale systems,” *IEEE Transactions on Automatic Control*, 2018.
- [56] Y. Chen and J. Anderson, “System level synthesis with state and input constraints,” *arXiv:1903.07174*, 2019.
- [57] S. Dean, S. Tu, N. Matni, and B. Recht, “Safely learning to control the constrained linear quadratic regulator,” *arXiv:1809.10121*, 2018.
- [58] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Prentice Hall New Jersey, 1996.
- [59] 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.
- [60] Ş. Sabău and N. C. Martins, “Youla-like parametrizations subject to QI subspace constraints,” *Automatic Control, IEEE Transactions on*, vol. 59, no. 6, pp. 1411–1422, 2014.
- [61] Y.-S. Wang, “A system level approach to optimal controller design for large-scale distributed systems,” Ph.D. dissertation, California Institute of Technology, 2017.
- [62] A. Rantzer, “Control synthesis with localizability and passivity constraints,” *arXiv:1812.07748*, 2018.
- [63] E. Jensen and B. Bamieh, “Optimal spatially-invariant controllers with locality constraints: A system level approach,” in *Proceedings of the 2018 Annual American Control Conference*. IEEE, 2018, pp. 2053–2058.
- [64] J. Anderson, N. Matni, and Y. Chen, “Sparsity preserving discretization with error bounds,” *arXiv:1903.11267*, 2019.
- [65] D. E. Rivera, M. Morari, and S. Skogestad, “Internal model control: Pid controller design,” *Industrial & engineering chemistry process design and development*, vol. 25, no. 1, pp. 252–265, 1986.
- [66] C. E. Garcia and M. Morari, “Internal model control. a unifying review and some new results,” *Industrial & Engineering Chemistry Process Design and Development*, vol. 21, no. 2, pp. 308–323, 1982.
- [67] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [68] N. Matni and V. Chandrasekaran, “Regularization for design,” *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3991–4006, Dec 2016.



- [69] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.
- [70] M. Dahleh and J. Pearson, J., “ $l^1$ -optimal feedback controllers for MIMO discrete-time systems,” *Automatic Control, IEEE Transactions on*, vol. 32, no. 4, pp. 314–322, Apr 1987.
- [71] P. L. Combettes and J.-C. Pesquet, “Proximal splitting methods in signal processing,” *arXiv preprint*, vol. arXiv:0912.3522, 2010. [Online]. Available: <http://arxiv.org/abs/0912.3522v4.pdf>
- [72] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.
- [73] P. Tseng, “Applications of a splitting algorithm to decomposition in convex programming and variational inequalities,” *SIAM Journal on Control and Optimization*, vol. 29, no. 1, pp. 119–138, 1991.
- [74] A. Lamperski and J. C. Doyle, “The  $\mathcal{H}_2$  control problem for quadratically invariant systems with delays,” *Automatic Control, IEEE Transactions on*, vol. 60, no. 7, pp. 1945–1950, 2015.