2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

System Level Synthesis: A Tutorial

John C. Doyle, Nikolai Matni, Yuh-Shyang Wang, James Anderson, and Steven Low

***Abstract*— This tutorial paper provides an overview of the System Level Approach to control synthesis; a scalable framework for large-scale distributed control. The system level approach is composed of three central components: System Level Parameterizations (SLPs), System Level Constraints (SLCs) and System Level Synthesis (SLP) problems. We describe how the combination of these elements parameterize the largest known class of constrained controllers that admit a convex formulation.**

I. About This Tutorial

In this tutorial paper we will present the *system level approach* (SLA) for synthesizing distributed controllers. There is far more material than we can hope to cover in a 12-page tutorial. The idea of this paper is that it should provide a snapshot of what the SLS framework is, how it can be implemented, and provide some very simple examples. We have created a website `www.cds.caltech.edu/syslevelsyn` which contains a largely extended version of this paper and will also include code, case studies, and links to recent papers. The outline of the paper is given below, any subjects in *italics* only appear in the extended online version of this paper.

- **Section II:** Introduction.
- **Section III:** Problem statement, *Youla parameterization, quadratic invariance (QI), motivating non-QI example.*
- **Section IV:** System level parameterization, state feedback, robustness, *output feedback.*
- **Section V:** Case studies, trade-offs, chain examples, *power networks, regularization for design.*
- **Section VI:** Conclusion.

II. Introduction

The Youla parameterization is arguably one of the most important results in control theory. In the seminal paper [1], Youla showed that there exists an isomorphism between between a stabilizing controller and the closed loop response from sensors to actuators. From a practical perspective this result showed that all possible closed-loop system responses could be achieved by an affine expression of the Youla parameter (often denoted $\mathbf{Q}$), this in turn allowed for direct optimization of the closed loop response. Together with state-space methods, this contribution played a major role in shifting controller synthesis from an ad hoc, loop-at-a-time tuning process to a principled one with well defined notions of optimality. Indeed, this approach proved very powerful, and paved the way for the foundational results of robust and optimal control that would follow [2]. With the advent of modern optimization-based control, the interpretation of $\mathbf{Q}$ being a closed loop response was downplayed and attention was turned to developing tractable optimization methods.

The System Level Approach (SLA) that we present in this tutorial is inspired by the system level thinking pioneered by Youla: rather than directly designing only the feedback loop between sensors and actuators, we propose directly designing the *entire closed loop response of the system*, as captured by the maps from process and measurement disturbances to control actions and states. A distinction between the SLA approach and Youla's is that the SLA explicitly models the internal delay structure of the feedback system, whereas Youla (and contemporary state-space methods) hid the internal structure of the controller, and focused instead on its input-output behavior (Figure 1: Top). This focus on controller input-output behavior was natural for the problems of that era (often motivated by aerospace and process control applications), where systems had a single logically centralized controller with global access to sensor measurements and global control over actuators. However, in the network setting, a single controller with instantaneous access to all system measurements and the ability to broadcast control actions without delay to the actuators is unrealistic. Hence, the communication and information sharing structure must be considered (Figure 1: Bottom.)

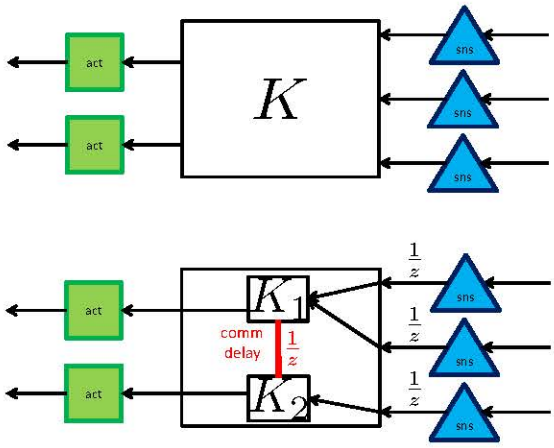

Fig. 1: Top: A classical centralized controller mapping sensor measurements to actuator commands. In this setting there are no communication constraints. Bottom: A distributed controller composed of two sub-controllers that can share information with a delay of 1 time-step.

J.C. Doyle, N. Matni, J. Anderson, and S. Low are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA. Y.-S. Wang is with GE Global Research. This research was supported by grants from the AFOSR and NSF, and by gifts from Huawei and Google. `doyle@caltech.edu`, `nmatni@caltech.edu`, `yuh-shyang.wang@ge.com`, `james@caltech.edu`, `slow@caltech.edu`.

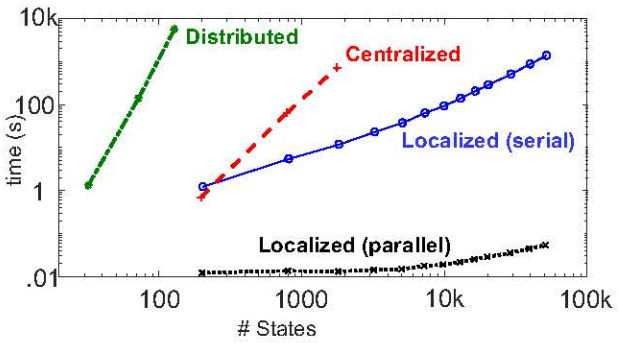

Fig. 2: Scalability of the state of the art distributed approach [3], classical centralized methods, and localized systems level approaches.

The resulting systems are composed of several sub-controllers, each equipped with their own sensors and actuators – sub-controllers then exchange locally available information via a communication network. It follows that the information exchanged between sub-controllers is constrained by the delay, bandwidth and reliability properties of this communication network, ultimately manifesting as information asymmetry among sub-controllers of the system. It is this information asymmetry, that makes distributed optimal controller synthesis challenging [4]–[9]. In fact, even for decentralized systems that have a very simple description, the resulting optimal control solution is still unknown [10].

Over the last 20 years the task of porting centralized control synthesis techniques to the distributed case has proven to be a formidable challenge. The most successful framework in this direction is the *quadratic invariance* (QI) methodology introduced by Rotkowitz and Lall [6]. The QI approach exactly characterizes the information constraints that can be imposed on the controller such that the resulting Youla parameterization of the closed-loop maintains convexity. In the extended version of this paper, we will describe quadratic invariance and draw comparisons with it when illustrating the SLA framework in Section V.

A fact that is not emphasized in the literature is that distributed controllers are actually more complex to synthesize and implement than their centralized counterparts. In Figure 2 the cost of synthesizing both distributed and centralized controllers is plotted as a function of the state dimension. One may be tempted to concede that this is simply an inherent limitation of the optimal control approach: achieving optimal performance in a large-scale system requires complex controllers, and the best that can be hoped for are approximations and/or principled heuristics. Our recent results prove otherwise, and in doing so, significantly generalize foundational concepts from classical and distributed optimal control such as the Youla parameterization and quadratic invariance.

With this tutorial paper we aim to present our framework for scalable distributed control which we term the systems level approach. We have tried to collect the salient points from our recent papers [11]–[22] on this topic and assemble them into one coherent paper and back this up with additional material, code, and case studies on the website. Our hope is that this will encourage others to find new and exciting application areas and take up the challenge of posing and answering new theoretical questions in this framework. We now provide a brief summary of the key points of this tutorial.

The SLA to controller synthesis is composed of three elements: System Level Parameterizations (SLPs), System Level Constraints (SLCs) and System Level Synthesis (SLS) problems. We highlight below some of the novel theoretical and computational tools applicable to the area of structured and constrained optimal controller synthesis that the SLA provides.

- Novel SLPs of all stabilizing controllers and the closed loop responses that they achieve [Section IV.A];
- 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 [Section IV.B];
- 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 [Section IV.A and IV.B];
- The set of constrained stabilizing controllers that can be efficiently parameterized using SLPs and SLCs is a strict superset of those that can be parameterized using quadratic invariance – hence the SLA provides a generalization of the QI framework, characterizing the broadest known class of constrained controllers that admit a convex parameterization [online];
- A catalog of SLCs that admit a convex representation will be given: highlights include general convex constraints on the Youla parameter (QI subspace constraints being a special case thereof), multi-objective performance constraints, spatiotemporal constraints on the system response, and constraints on the architectural complexity of the controller [23], [24] [Section IV.B and online];
- SLPs and SLCs can be combined to formulate a SLS problem, which defines the broadest known class of constrained optimal control problems that can be solved using convex programming [Section IV.C];
- The optimal control problems considered in the QI and localized optimal control literature are all special cases of SLS problems [online].
- Robust variants of SLPs allow for localized controller synthesis and implementation even if a system is not exactly localizable [Section IV.D];
- The SLA can naturally accommodate noisy communication between sub-controllers and other uncertainties which are "internal" to the control system, which will be increasingly important issues, especially in the context of biological and low-power systems [online].

- The system level approach can be used to quantify tradeoffs in large-scale systems, such as those that arise between controller performance and (i) implementation complexity, (ii) controller robustness to internal computation error, and (iii) architectural complexity [Section V].

Preliminaries & Notation: Lower and upper case boldface Latin letters such as $\mathbf{x}$ and $\mathbf{G}$ denote signals and transfer matrices, respectively, and calligraphic letters such as $\mathcal{S}$ to denote sets. We work with discrete linear time invariant (LTI) 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 $\mathcal{RH}_\infty$. Let $G[i]$ 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., $\mathcal{F}_T := \{\mathbf{G} \in \mathcal{RH}_\infty \,|\, \mathbf{G} = \sum_{i=0}^{T} \frac{1}{z^i} G[i]\}$.

III. Problem Statement

Throughout this paper we will be concerned with designing distributed controllers for discrete-time LTI systems that take the following form

$$x[t+1] = Ax[t] + B_1 w[t] + B_2 u[t] \tag{1a}$$
$$\bar{z}[t] = C_1 x[t] + D_{11} w[t] + D_{12} u[t] \tag{1b}$$
$$y[t] = C_2 x[t] + D_{21} w[t] + D_{22} u[t] \tag{1c}$$

where x, u, w, y, $\bar{z}$ are the state vector, control action, external disturbance, measurement, and regulated output, respectively. It is implicitly assumed that the dimensions of these signals are all of compatible dimensions. Equation (1) can be written in partitioned 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 $\mathbf{P}_{ij} = C_i(zI - A)^{-1}B_j + D_{ij}$ is a sub-transfer matrix. We refer to $\mathbf{P}$ as the open loop plant model.

It is further assumed that the system is sparse, specifically that it is composed of n coupled subsystems that interact with each other according to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The nodes $\mathcal{V} = \{1, \ldots, n\}$ represent the set of subsystems, and the edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ imply an interaction between two such subsystems. An edge $(i, j) \in \mathcal{E}$ means that the state x_j of subsystem j directly affects the states x_i of subsystem i.

Throughout this work we will deal with a dynamic output feedback control law of the form $\mathbf{u} = \mathbf{K}\mathbf{y}$, where $\mathbf{K}$ is assumed to have the state space realization

$$\xi[t+1] = A_k \xi[t] + B_k y[t], \tag{2a}$$
$$u[t] = C_k \xi[t] + D_k y[t], \tag{2b}$$

with internal state ξ. In the z-domain it follows that $\mathbf{K} = C_k(zI - A_k)^{-1}B_k + D_k$. A schematic diagram of the interconnection of the plant $\mathbf{P}$ and the controller $\mathbf{K}$ is shown in Figure 3. The following assumptions are made throughout the paper.

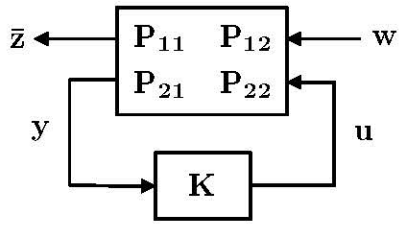

Fig. 3: Interconnection of the plant $\mathbf{P}$ and controller $\mathbf{K}$.

Assumption 1: The interconnection in Figure 3 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.

We now have all the pieces to be able to state the canonical distributed control problem [6], [25]–[27]. The goal is to minimize a suitable chosen norm of the closed loop map by selecting a controller $\mathbf{K}$ that has a predefined structure. Formally this can be stated as

$$\begin{array}{ll} \underset{\mathbf{K}}{\text{minimize}} & \|\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\| \\ \text{subject to} & \mathbf{K} \text{ internally stabilizes } \mathbf{P} \\ & \mathbf{K} \in \mathcal{C}, \end{array} \tag{3}$$

where $\mathcal{C}$ is a subspace that incorporates the *information constraint*. This subspace can enforce, for instance, the information sharing constraints imposed on the controller $\mathbf{K}$ by the underlying communication network. For an example of an information constraint, consider the controller structure depicted in the bottom of Figure 1. The controller consist of two sub-controllers that exchange information but incur a delay in doing so. The first controller receives the first two measurements directly and the third is received after a delay, via the second controller. Similarly, the second controller gets the third measurement directly, while the first two signals after received through the controller after a one-step delay. This mapping from sensor to actuation can be written as

$$\begin{bmatrix} u_1(z) \\ u_2(z) \end{bmatrix} = \Big(\frac{1}{z}\begin{bmatrix} * & * & 0 \\ 0 & 0 & * \end{bmatrix} \oplus \frac{1}{z^2}\begin{bmatrix} * & * & * \\ * & * & * \end{bmatrix} \oplus \ldots\Big) \begin{bmatrix} y_1(z) \\ y_2(z) \\ y_x(z) \end{bmatrix}$$

where a $*$ denotes a not specified, non-zero entry. Such a mapping can be encoded in the subspace constraint $\mathcal{C}$.

If the final constraint $\mathbf{K} \in \mathcal{C}$ is removed the standard centralized model matching problem is recovered. In this case the control action is simply $u = \mathbf{K}y$. In the extended version of this tutorial we include a detailed description of the Youla parameterization that convexifies (3) and the notion of *quadratic invariance* [6] which extends this approach to the distributed setting.

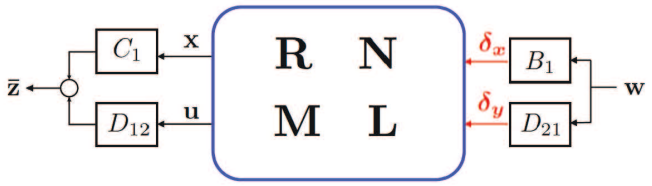

Fig. 4: The closed-loop response mapping $\mathbf{w}$ to $\bar{\mathbf{z}}$. The SLP that maps the part of the closed-loop response that we design is the map $(\delta_{\mathbf{x}}, \delta_{\mathbf{y}}) \to (\mathbf{x}, \mathbf{u})$.

IV. System Level Approach

In this section the three core components of the system level approach are introduced: the system level parameterization (SLP), system level constraints (SLCs), and system level synthesis (SLP) problems.

A. System Level Parameterization

For an LTI system with dynamics given by (1), we define the *system response* $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ to be the maps satisfying

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{\boldsymbol{x}} \\ \boldsymbol{\delta}_{\boldsymbol{y}} \end{bmatrix}, \tag{4}$$

where $\boldsymbol{\delta}_{\boldsymbol{x}} = B_1\mathbf{w}$ is the disturbance on the state vector, and $\boldsymbol{\delta}_{\boldsymbol{y}} = D_{21}\mathbf{w}$ is the disturbance on the measurement.

We say that a system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ is *stable and achievable* with respect to a plant $\mathbf{P}$ if there exists an internally stabilizing controller $\mathbf{K}$ such that the interconnection illustrated in Figure 3 leads to closed loop behavior consistent with equation (4). Figure 4 illustrates the system response described by equation (4) as a block diagram. Unlike the Youla approach, keeping the state in the mapping is absolutely central to the system level approach.

The following theorem is at the centre of the system level approach, it characterizes in an algebraic manner, all the achievable responses for a stabilizing controller.

Theorem 1: For the output feedback problem with $D_{22} = 0$ in (1) the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ transfer matrices from (4) are:

$$\begin{aligned} \mathbf{R} &= (zI - A - B_2\mathbf{K}C_2)^{-1} \\ \mathbf{M} &= \mathbf{K}C_2\mathbf{R} \\ \mathbf{N} &= \mathbf{R}B_2\mathbf{K} \\ \mathbf{L} &= \mathbf{K} + \mathbf{K}C_2\mathbf{R}B_2\mathbf{K}. \end{aligned} \tag{5}$$

and the following are true:

(a) The affine subspace described by:

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{6a}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{6b}$$

$$\mathbf{R}, \mathbf{M}, \mathbf{N} \in \frac{1}{z}\mathcal{RH}_\infty, \quad \mathbf{L} \in \mathcal{RH}_\infty \tag{6c}$$

parameterizes all system responses (5) achievable by an internally stabilizing controller $\mathbf{K}$.

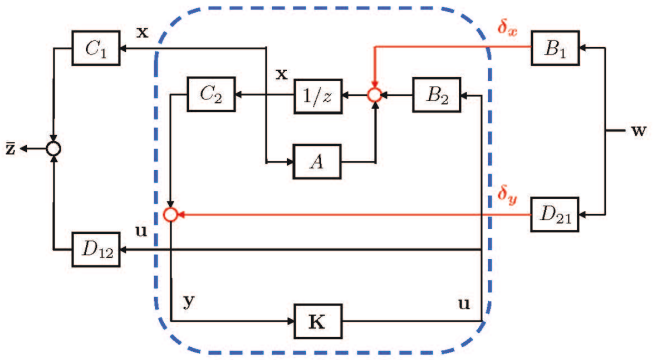

Fig. 5: Representation of the closed loop map in terms of the state space matrices from (1) and the controller $\mathbf{K}$. The blocks inside the dotted blue give the state space equivalent of the system response $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$.

(b) For any transfer matrices $\{\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}\}$ satisfying (6), the controller $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$ is internally stabilizing and achieves the desired response (5).

As this characterizes all the achievable system responses we refer to it as the system level parameterisation (SLP). In the same way that the Youla parameterization characterizes all maps from exogenous inputs to outputs, the SLP characterizes all closed loop responses from $(\boldsymbol{\delta}_x, \boldsymbol{\delta}_y)$ to $(\mathbf{x}, \mathbf{u})$. One of the key differences is that the SLP keeps the state in the parameterization. This is important because, intuitively, the state matrix A contains the bulk of the system structure. In Figure 5 the state space matrices are included in the closed-loop block diagram.

It should be noted that the SLP (6a)-(6c) is an affine subspace, thus it is a tractable constraint that can be incorporated into a convex optimization. This idea is central to the system level constraints (SLCs) that will be described in Section IV-B.

In this abridged tutorial paper we will only analyze Theorem 1 in the state feedback context. In this setting it admits a simpler characterization and allows us to provide intuition about the construction of a controller that achieves a desired system response. The output feedback controller is discussed in the extended version of this paper[1] as well as in [19].

*1) **State Feedback**:* Here we consider the problem of designing a state feedback controller when the plant model takes the form

$$\mathbf{P} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ I & 0 & 0 \end{array}\right]. \tag{7}$$

In this setting the controller $\mathbf{K}$ has full access to the state as $C_2 = I$ and the system response is simply given by the two transfer matrices $\{\mathbf{R}, \mathbf{M}\}$ which we now derive.

Taking the z-transform of the state dynamics from (1a) gives

$$(zI - A)\mathbf{x} = B_2\mathbf{u} + \boldsymbol{\delta}_{\boldsymbol{x}}, \tag{8}$$

[1]Available at `www.cds.caltech.edu/syslevelsyn`

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

By definition $\mathbf{R}$ is the system response mapping the external disturbance $\boldsymbol{\delta}_x$ to the state $\mathbf{x}$. Likewise, $\mathbf{M}$ is the system response mapping the disturbance $\boldsymbol{\delta}_x$ to the control action $\mathbf{u}$, i.e.

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix} = \begin{bmatrix}\mathbf{R}\\ \mathbf{M}\end{bmatrix}\boldsymbol{\delta}_x.$$

Substituting the dynamic state feedback control law $\mathbf{u} = \mathbf{K}\mathbf{x}$ into (8), the system response $\{\mathbf{R},\mathbf{M}\}$ as a function of the controller $\mathbf{K}$ can be written as

$$\begin{aligned}\mathbf{R} &= (zI - A - B_2\mathbf{K})^{-1}\\ \mathbf{M} &= \mathbf{K}(zI - A - B_2\mathbf{K})^{-1}.\end{aligned} \tag{9}$$

Using the above system response, we can now present the state feedback version of Theorem 1.

Theorem 2: For the state feedback system (7), the following are true:

(a) The affine subspace defined by

$$\begin{bmatrix}zI - A & -B_2\end{bmatrix}\begin{bmatrix}\mathbf{R}\\ \mathbf{M}\end{bmatrix} = I \tag{10a}$$

$$\mathbf{R},\mathbf{M} \in \frac{1}{z}\mathcal{RH}_\infty \tag{10b}$$

parameterizes all system responses from $\boldsymbol{\delta}_x$ to $(\mathbf{x},\mathbf{u})$, as defined in (9), achievable by an internally stabilizing state feedback controller $\mathbf{K}$.

(b) For any transfer matrices $\{\mathbf{R},\mathbf{M}\}$ satisfying (10), the controller $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$ is internally stabilizing and achieves the desired system response (9).

Proof: Necessity of Theorem 2 is provided in [19]. The sufficiency part of the theorem is discussed in the extended version of this paper as it provides a construction of the controller $\mathbf{K}$. ∎

The following lemma connects the affine subspace constraint (10) to classical notions of stabilizability:

Lemma 1: The pair (A, B_2) is stabilizable if and only if the affine subspace defined by (10) is non-empty.

Combining Theorem 2 and Lemma 1 we have the following implication:

$$(A, B_2) \text{ stabilizable} \iff \exists\ \{\mathbf{R},\mathbf{M}\} \text{ s.t. (10) feasible.}$$

Although the controller can be directly constructed according to $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$, the following disturbance-based implementation has several advantages as we shall see in Section IV-B. Let the system response $\{\mathbf{R},\mathbf{M}\}$ constrained to be finite impulse response (FIR) transfer matrices with horizon T. Now consider the disturbance-based controller implementation:

$$\hat{\delta}_x[t] = x[t] - \hat{x}[t] \tag{11a}$$

$$u[t] = \sum_{\tau=0}^{T-1} M[\tau+1]\hat{\delta}_x[t-\tau] \tag{11b}$$

$$\hat{x}[t+1] = \sum_{\tau=0}^{T-2} R[\tau+2]\hat{\delta}_x[t-\tau]. \tag{11c}$$

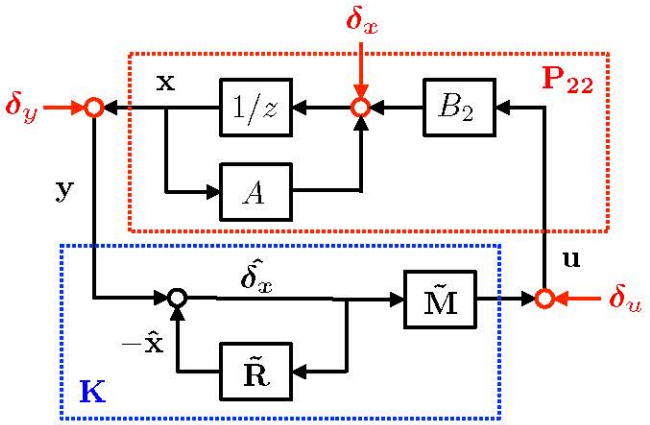

Fig. 6: Structure of the state feedback controller as described by implementation (12), where $\tilde{\mathbf{R}} := I - z\mathbf{R}$ and $\tilde{\mathbf{M}} := z\mathbf{M}$.

The internal states of the controller (11) should be interpreted as follows: $\hat{\delta}_x$ is the controller estimate of the state disturbance, and $\hat{x}$ is a desired or reference state trajectory. The estimated disturbance $\hat{\delta}_x[t]$ is computed by taking the difference between the current state measurement $x[t]$ and the current reference state value $\hat{x}[t]$. The control action $u[t]$ and the next reference state value $\hat{x}[t+1]$ are then computed using past estimated disturbances $\hat{\delta}_x[t-T+1],\ldots,\hat{\delta}_x[t]$.

Taking the z-transform of equations (11), we obtain their representation in the frequency domain

$$\hat{\boldsymbol{\delta}}_x = \mathbf{x} - \hat{\mathbf{x}} \tag{12a}$$

$$\mathbf{u} = z\mathbf{M}\hat{\boldsymbol{\delta}}_x \tag{12b}$$

$$\hat{\mathbf{x}} = (z\mathbf{R} - I)\hat{\boldsymbol{\delta}}_x. \tag{12c}$$

Combining equations (12) with (8) and (10), one can verify that the estimated disturbance $\hat{\delta}_x[t]$ indeed reconstructs the true disturbance $\delta_x[t-1]$ that perturbed the plant at time $t-1$; hence $\hat{\boldsymbol{\delta}}_x = z^{-1}\boldsymbol{\delta}_x$. It is then straightforward to show that the desired system response $\{\mathbf{R},\mathbf{M}\}$ satisfying $\mathbf{x} = \mathbf{R}\boldsymbol{\delta}_x$ and $\mathbf{u} = \mathbf{M}\boldsymbol{\delta}_x$ is achieved. Note that the previous argument holds for any FIR horizon T as well as for $T = \infty$. The controller architecture is surprisingly simple and is illustrated in Figure (12).

Lemma 2: Consider the state feedback system (7). Given any system response $\{\mathbf{R},\mathbf{M}\}$ lying in the affine subspace described by (10), the state feedback controller $\mathbf{K} = \mathbf{M}\mathbf{R}^{-1}$, with structure shown in Figure 6, internally stabilizes the plant. In addition, the desired system response, as specified by $\mathbf{x} = \mathbf{R}\boldsymbol{\delta}_x$ and $\mathbf{u} = \mathbf{M}\boldsymbol{\delta}_x$, is achieved.

Proof: See extended version. ∎

Remark 1: In stark contrast to the Youla approach, the system level approach does not place constraints on the controller. Instead constraints are placed on the entire system response, in the case of state feedback, this corresponds to placing constraints on $\{\mathbf{R},\mathbf{M}\}$.

Remark 2: Theorem 2 provides a necessary and sufficient condition for the system response $\{\mathbf{R},\mathbf{M}\}$ to be stable and achievable, in that elements of the affine subspace defined by

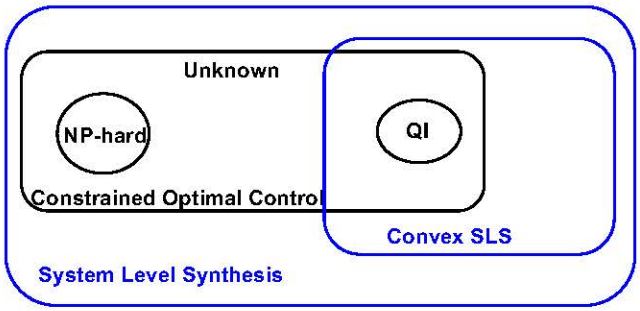

Fig. 7: Venn diagram illustrating how the various areas of distributed optimal control interlink. Of note here is that the class of QI problems are contained strictly inside the convex SLS class.

(10) parameterize all stable system responses achievable via state-feedback, as well as the internally stabilizing controllers that achieve them.

Remark 3: All controllers that are constructed from the SLA are dynamic, recall that $\mathbf{R}$ and $\mathbf{M}$ are transfer matrices. Note too that the Youla parameterization also only yields dynamic controllers.

B. System Level Constraints

In this section we will show that applying constraints to the SLPs developed in the previous section will allow for a truly scalable synthesis framework. As the constraints are imposed on the system response (as was traditionally attempted with Youla) we refer to them as system level constraints (SLCs). SLCs, together with the SLP and a suitably chosen cost functional, lead to the system level synthesis (SLS) problem which takes the form

$$\underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad g(\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}) \tag{13a}$$

$$\text{subject to} \quad \text{equations (6a)} - \text{(6b)} \tag{13b}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{S} \tag{13c}$$

where $\mathcal{S}$ is the SLC. The state feedback version of (13) follows in an analogous manner. In section IV-C suitable cost functionals and scalable synthesis procedures will be discussed. The goal of this section is to describe the design choices available for $\mathcal{S}$ and to examine how they can be used to generate a localized system response. The SLC set $\mathcal{S}$ can be viewed of as the intersection of three sets

$$\mathcal{S} = \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X},$$

where $\mathcal{L}$ is a *locality* constraint, $\mathcal{F}_T$ is is a spatiotemporal constraint, and $\mathcal{X}$ is an arbitrary convex constraint that can be used for miscellaneous constraints such as performance bounds for example. With this description we arrive at the *localized SLS* problem which is the same as optimization (13) but with constraint (13c) replaced with

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{L} \cap \mathcal{F}_T \cap \mathcal{X}.$$

Before describing various spatial and temporal constrains a remark about Youla and QI subspace constraints are in order. Although it won't be proved in this tutorial paper, it is important to note that any controller of the form (**??**), subject to constraints on $\mathbf{Q}$ can be equivalently expressed by $\mathbf{K} = \mathbf{L} - \mathbf{M}\mathbf{R}^{-1}\mathbf{N}$. As a corollary, any subspace constraint that is quadratically invariant can also be expressed in the SLS framework [19]. To make this more clear, the hierarchies of distributed control are plotted as a Venn diagram in Figure 7.

1) Locality Constraints $\mathcal{L}$: Locality constraints are used to enforce structure on a transfer matrix, specifically sparsity. Such constraints may be desirable when there is a large financial cost to install a controller, physically placing the controller is difficult, or when obtaining a measurement from certain area of the network is not possible.

Given a transfer matrix $\mathbf{G} \in \mathcal{RH}_\infty$, let $\mathbf{G}_{ij}$ denote the element at row i, column j of $\mathbf{G}$. In this case the sparsity constraint specifies which $\mathbf{G}_{ij} = 0$. Formally this called a subspace constraint and is written as

$$\begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{L} \quad \text{or} \quad \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{L} \tag{14}$$

depending on whether or not the problem is an output, or state feedback one.

Of particular interest are subspaces $\mathcal{L}$ that define transfer matrices of sparse support. An immediate benefit of enforcing such sparsity constraints on the system response is that implementing the resulting controller can be done in a *localized way*, i.e., each controller state and control action can be computed using a local subset (as defined by the support of the system response) of the global controller state and sensor measurements. For this reason, we refer to the constraint (14) as a *localized* SLC when it defines a subspace with sparse support. Further, as we highlight in the next section, such localized constraints also allow for the resulting system response to be computed in a localized way, i.e., the global computation decomposes naturally into decoupled subproblems that depend only on local submatrices of the state-space representation (1). Clearly, both of these features are extremely desirable when computing controllers for large-scale systems.

2) Temporal Constraints $\mathcal{F}_T$: Temporal constraints can be enforced by requiring that the system response is made to be a finite impulse response. In this case the system designer can select the horizon length T. Given the parameterization of stabilizing controllers of Theorem 1, it is trivial to enforce that a system response be FIR with horizon T via the following SLC

$$\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L} \in \mathcal{F}_T. \tag{15}$$

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

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

(b) The controller achieving the desired system response can be implemented using the FIR filter banks $\tilde{\mathbf{R}}, \tilde{\mathbf{M}}, \in \mathcal{F}_T$,

as illustrated in Figure 6. This simplicity of implementation is extremely helpful when applying these methods in practice.

(c) When a FIR SLC is imposed, the resulting set of stable achievable system responses and corresponding controllers admit a finite dimensional representation – specifically, the constraints specified in Theorem 1 only need to be applied to the impulse response elements $\{R[t], M[t], N[t], L[t]\}_{t=0}^{T}$.

Remark 4: It should be noted that the computational benefits claimed above hold only for discrete time systems. For continuous time systems, a FIR transfer matrix is still an infinite dimensional object, and hence the resulting parameterizations and constraints are in general infinite dimensional as well.

3) Other Constraints $\mathcal{X}$: Generic convex system performance bounds of the form

$$g(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L}) < \gamma$$

for fixed γ can be included here. A typical choice is th $\mathcal{H}_2$-norm. This will be elaborated on further in Section IV-C.

C. System Level Synthesis

In this section the SLP and SLCs that we have described previously are combined to produce a synthesis algorithm that we call the system level synthesis (SLS) problem. For the state feedback case, the SLS problem takes the form

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & g(\mathbf{R}, \mathbf{M}) \\ \text{subject to} \quad & \text{(10a)} - \text{(10b)} \\ & \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{S}. \end{aligned} \tag{16}$$

Provided that the cost function g is chosen to be convex, the resulting SLP (16) will be a convex program.

The main difference between the system level approach and the Youla/QI based based methods are that here, the emphasis is placed on feasibility. For Youla based methods, the hard work is in showing that the problem is QI and thus convex. In comparison, the SLA requires constraint satisfaction which may not always be possible. What happens when feasibility is not net is dealt with in Section IV-D.

Localized LQG Control

In [12], [15] the localized LQG optimal control problem was posed and solved. It can be recovered as a special case of the SLS problem (16) by selecting the system norm $\|\cdot\|$ to be the $\mathcal{H}_2$ norm, and selecting the constraint set $\mathcal{S}$ to be a spatiotemporal SLC. In the case of a state-feedback problem [12], the resulting SLS problem is of the form

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & \|C_1\mathbf{R} + D_{12}\mathbf{M}\|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \text{(10a)} - \text{(10b)} \\ & \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T, \end{aligned} \tag{17}$$

for $\mathcal{C}$ a QI subspace SLC, $\mathcal{L}$ a sparsity SLC, and $\mathcal{F}_T$ a FIR SLC.

The observation that we make in [12] (and extend to the output feedback setting in [15]), is that the localized SLS problem (17) can be decomposed into a set of independent sub-problems solving for the columns $\mathbf{R}_i$ and $\mathbf{M}_i$ of the transfer matrices $\mathbf{R}$ and $\mathbf{M}$ – as these problems are independent, they can be solved in parallel. Further, the sparsity constraint $\mathcal{L}$ restricts each sub-problem to a local subset of the system model and states, as specified by the nonzero components of the corresponding column of the transfer matrices $\mathbf{R}$ and $\mathbf{M}$, allowing each of these sub-problems to be expressed in terms of optimization variables (and corresponding sub-matrices of the state-space realization (6)) that are of significantly smaller dimension than the global system response $\{\mathbf{R}, \mathbf{M}\}$. Thus for a given feasible spatiotemporal SLC, the localized SLS problem (17) can be solved for arbitrarily large-scale systems, assuming that each sub-controller can solve its corresponding sub-problem in parallel.[2]

D. Virtually Localizable Systems

It was mentioned in Section IV-C that feasibility may not be possible in some circumstances. This may be because there is not enough actuation in the system to control it, or the time horizon in the FIR filters is too short. In this section we briefly present a robustness result that can be seen as providing *virtual localization* to system responses that would otherwise be infeasible. The idea is reminiscent of classical robust control, where we allow for a $\mathbf{\Delta}$ perturbation. This work is presented in [18], but we summarize the main result here and apply it to some simple case studies in the next section.

Theorem 3: Let $(\mathbf{R}_c, \mathbf{M}_c, \mathbf{\Delta})$ be a solution to

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_c \\ \mathbf{M}_c \end{bmatrix} = I + \mathbf{\Delta}$$

Then, the controller implementation

$$\begin{aligned} \hat{\boldsymbol{\delta}}_x &= \mathbf{x} - \hat{\mathbf{x}} \\ \mathbf{u} &= z\mathbf{M}_c\hat{\boldsymbol{\delta}}_x \\ \hat{\mathbf{x}} &= (z\mathbf{R}_c - I)\hat{\boldsymbol{\delta}}_x \end{aligned}$$

internally stabilizes the system (A, B_2) if and only if $(I + \mathbf{\Delta})^{-1}$ is stable.

Define $\|\mathbf{\Delta}\|_{\mathcal{E}_1} := \|\mathbf{\Delta}^\top\|_{\mathcal{L}_1}$. Theorem 3 can now be combined with small gain theorems to provide simple sufficient conditions for robust stability.

Corollary 1 (Robustness): Under the conditions of Theorem 3, the closed loop system is stable if one of $\|\mathbf{\Delta}\|_{\mathcal{H}_\infty}, \|\mathbf{\Delta}\|_{\mathcal{L}_1}, \|\mathbf{\Delta}\|_{\mathcal{E}_1} < 1$.

Proof: Classical, see [28] for $\mathcal{H}_\infty$ and [29] for the remainder. ∎

Remark 5: Note that $\|\mathbf{\Delta}\|_{\mathcal{L}_1}$ and $\|\mathbf{\Delta}\|_{\mathcal{E}_1}$ are the worst case $\ell_\infty \to \ell_\infty$ and $\ell_1 \to \ell_1$ gains of $\mathbf{\Delta}$, respectively.

[2]We also show how to co-design an actuation architecture and feasible corresponding spatiotemporal constraint in [16], and so the assumption of a feasible spatiotemporal constraint is a reasonable one.

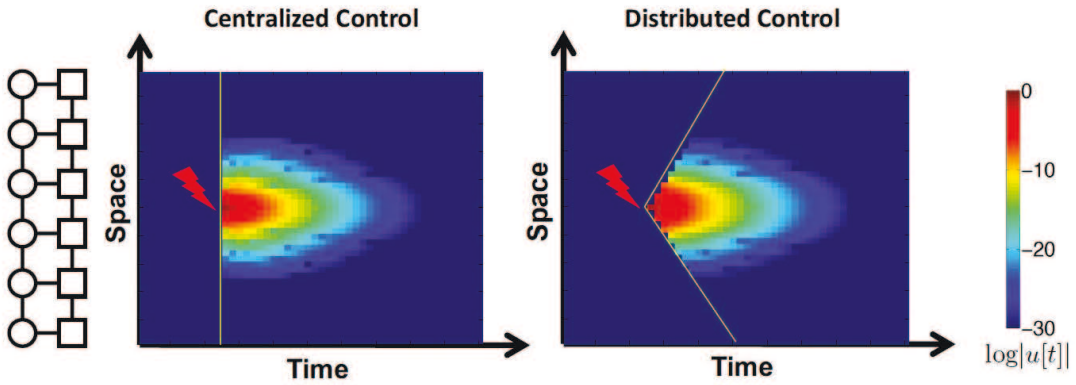

Fig. 8: Space time diagrams comparing the magnitude of the control action for two different control implementations. A single disturbance hits the centre of the chain. **Left:** In the centralized setting there is infinite communication speed and thus the controller can sense the disturbance, compute the control action, and broadcast it to the actuators instantly. The yellow line is the communication delay, in this instance zero delay corresponds to a vertical line. **Right:** In the distributed setting [3] we illustrate the optimal solution where the communication speed is twice as fast as the plant dynamics. Here the communication delay is clearly visible.

By combining Theorem 3 with corollary 1 a SLS problem for *virtually localizable systems* is given by

$$\begin{aligned} \underset{\mathbf{R}_c,\mathbf{M}_c,\mathbf{\Delta}}{\text{minimize}} \quad & \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R}_c \\ \mathbf{M}_c \end{bmatrix} (I+\mathbf{\Delta})^{-1} B_1 \right\| \\ \text{s.t.} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_c \\ \mathbf{M}_c \end{bmatrix} = I + \mathbf{\Delta} \\ & \begin{bmatrix} \mathbf{R}_c \\ \mathbf{M}_c \end{bmatrix} \in \mathcal{L} \cap \mathcal{F}_\mathcal{T} \cap \mathcal{X}, \ \|\mathbf{\Delta}\|_\star < 1 \end{aligned} \tag{18}$$

where $\|\cdot\|_\star$ is any of the norms described in corollary 1. Unfortunately the SLS problem (18) is not convex due to the objective function. To circumvent this problem, in [18] we introduce a quasi-convex upper bound to the objective function that can be efficiently optimized.

V. Case Studies

In this section we consider a simple chain model where each node in the chain corresponds to a single state. With this example we will illustrate clearly the concept of localization and explore the various trade offs between locality, communication speed, actuation density, and filter horizon. In the extended paper we examine power networks where nodes in the network have more complex dynamics (i.e. have multiple states).

A. Chain System

The first example considered is a bi-directional chain with $n = 100$ nodes, where each node corresponds to a scalar state. The node dynamics are given by

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

where α can be tuned to make the system more or less stable and κ tunes the coupling strength. The value b_i is given by 1 if there is an actuator at subsystem i, and 0 otherwise. We place 40 actuators in the chain network, with actuator location specified by $i = 5j - 4$ and $5j$ for $j = 1, \ldots, 20$. The objective function is the quadratic term $\|\mathbf{x}\|_2^2 + \gamma\|\mathbf{u}\|_2^2$. For this example we set $\kappa = 1$ and adjust the value of α until the system has a spectral radius of 1.1.

1) FIR, Locality, and Communication Speed: We are now able to specify the closed loop response (i.e. build the controller). The two design parameters we consider first are the FIR filter horizon and the size of the region we wish to localize to. The FIR horizon determines the settling time of the system, if this is chosen to be too short then the system will fail to localize in time and the SLS problem will fail to find a feasible solution. The spatial localization is achieved by imposing a SLC on $\{\mathbf{R}, \mathbf{M}\}$. In this case we impose a band structure with bands of width d. This corresponds to the controller obtaining information from d-hops of the physical system. The performance criteria is chosen to be the $\mathcal{H}_2$-norm of the closed loop response. In Figure 10 we plot the heat map of the control action for $(d, T) = (4, 10)$. Figure 10 should be compared with the centralized and distributed results shown in Figure 8.

The effects of varying the horizon length T are shown in Figure 9. The plot shows that these parameters lead to no degradation in performance with respect to a centralized optimal controller, while leading to significant improvements in synthesis and implementation complexity. If we choose $(d, T) = (5, 15)$, there is only 0.3% performance degradation compared to the centralized optimal controller, which corresponds to $(d, T) = (99, \infty)$. The result shown in Figure 9 (right) holds for a wide range of the parameters (κ, α) of the plant and controller.

We next study the effects of communication delay on the performance. We assume that the communication network has the same topology as the physical network, and it takes time t_c for a sub-controller to transmit information to its direct neighbors. The delay is normalized with respect to the sampling time of the discrete time system (1), and hence it may not necessarily be a whole integer. We adopt the following convention to handle fractional delays: if information is received by a sub-controller between two sampling times t and $t+1$, then it may be used by the sub-controller to compute its control action at time $t+1$. It is necessary to have $t_c < 1$ for the existence of a localized system response [16].

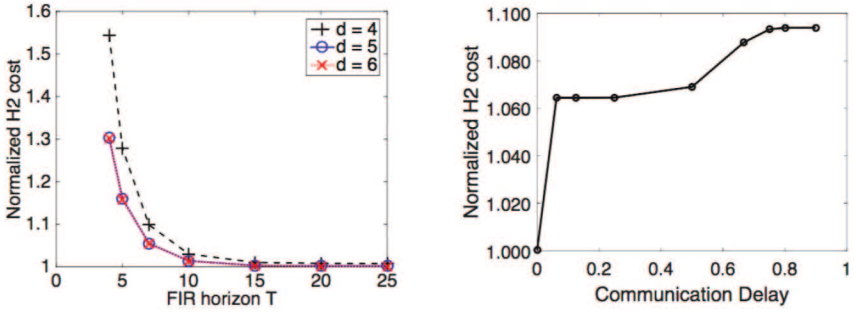

Fig. 9: Left: Closed loop performance vs. FIR horizon length T. **Right:** Closed loop performance vs. communication speed - $(d, T) = (13, 20)$ for $t_c = 0.9$, else $(8, 20)$.

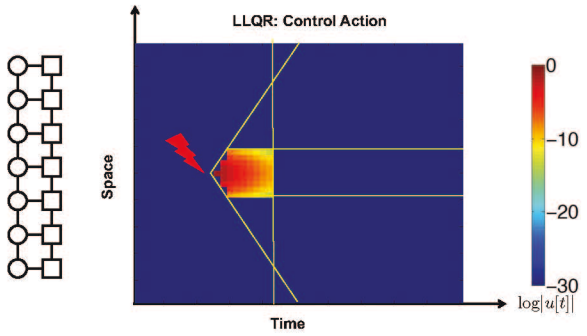

Fig. 10: The system has been successfully localized in both time and space. Moreover, it has been done more economically than in the centralized and distributed case.

We choose $(d, T) = (8, 20)$ for the previous example, and study the trade-off between communication delay t_c and the normalized $\mathcal{H}_2$ cost. As shown in Figure 9 (right), communication delay only leads to slight degradation in performance. Note that the degradation is mostly contributed by the delay constraint. To verify this claim, we compare our localized controller with the QI optimal controller on a 40-state chain example. Simulation shows that the localized FIR constraint $(d, T) = (8, 20)$ only leads to 0.03% degradation compared to a QI optimal controller with the same delay constraint.

In the next example we consider a 50 node chain with sparse actuation, specifically there is only actuation present at every other node in the network. The parameters α and κ are tuned to make the system marginally stable. We set $(d, T) = (4, 15)$, and study the effects of varying communication delays. In Figures 11–13 we show how the system localizes as the communication speed is reduced – here we define communication speed as the reciprocal of communication delay. The performance metric in this case was chosen to be the $\mathcal{E}_1$-norm of the closed loop. The virtual localization (c.f. Section IV-D) is needed in order to provide a feasible SLS problem when communication delay exceeds $2/3$. Without the robustness provided by Theorem 3, the resulting system level synthesis problems would have been infeasible. The *leakage* can be seen clearly in the heat maps. What is not shown (in the short version of this tutorial) is the striking result that even when the communication is much slower than the plant, performance is still considerably better than when running in open-loop. Looking at Figures 11–13, the vertical yellow lines represent the temporal constraints (i.e. the FIR horizon length) placed on the closed loop system. The horizontal lines correspond to the spatial constraint. Without the virtual localization result, any time there is *leakage* outside of the intersection of the horizontal and vertical lines as seen in Figures 12–13 then the synthesis problem would be infeasible and the interpretation would be that under this choice of design parameters the system could not be sufficiently localized.

2) Trade Offs: Finally we present a series of examples where the localization effects are clearly seen for varying design parameters. The open loop parameters remain unchanged from the previous example. The performance metric is the $\mathcal{H}_2$ norm for the closed loop (and the $\mathcal{E}_1$ norm for the Δ block where appropriate) - these values are provided in the figure captions. In Figure 14 we show how the localization of the closed loop changes as a function of the actuation density. At a value of 1 there is actuation at each node, at 0.5 there is actuation every other node. It is clear that the *virtual localization* allows for the design of controllers which require fewer actuators. In Figure 15 the sparsity measured in terms of the number of d-*hops* of the plant topology is varied. This has the effect of pushing the horizontal spatial constraints closer together as d decreases. In the final example, the FIR filter horizon is decreased causing the horizontal temporal constraint to move to the left, limiting the time that the disturbance is allowed to have any impact on the state.

VI. Conclusion

In this abridged tutorial paper we have presented the key aspects of the system level synthesis framework. We have created a wiki `www.cds.caltech.edu/syslevelsyn` that contains a much extended version of this paper, links to recent papers, case studies, and code to run the simulations from this paper.

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.

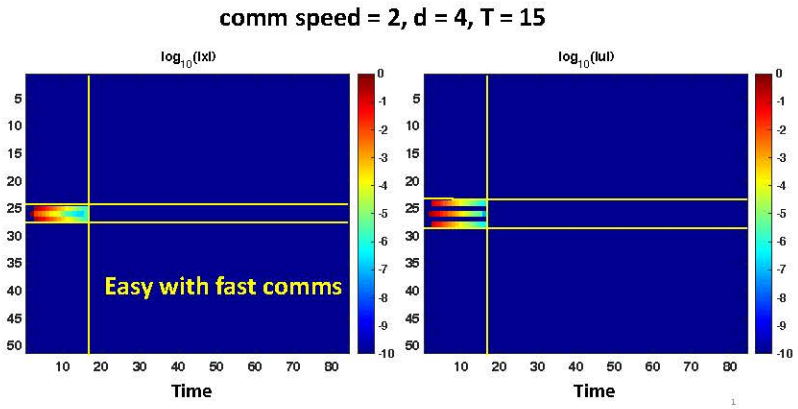

Fig. 11: When the communication speed is twice as fast as the dynamics, the system is easily localized. The state deviation is shown on the left and the control action on the right. The bands in the control signal appear because the system is sparsely actuated.

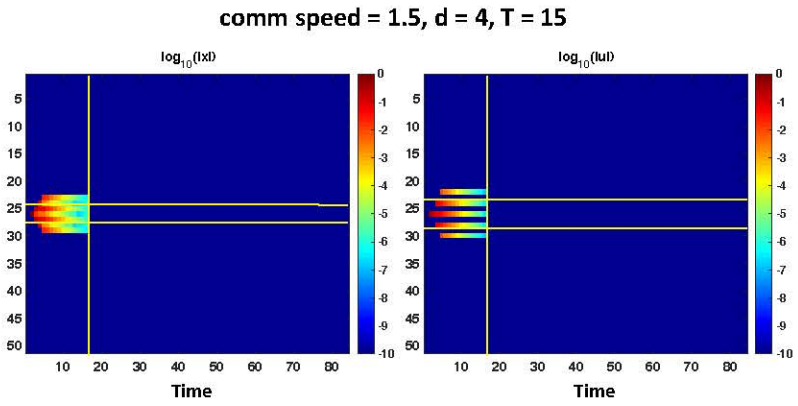

Fig. 12: As the communication speed decreases, the system fails to completely localize. Localization in time can be seen to be successful. However, in both the state and control action, there is a leak in the spatial dimension. Without the Δ term from the virtual localization the system level synthesis problem would have been infeasible.

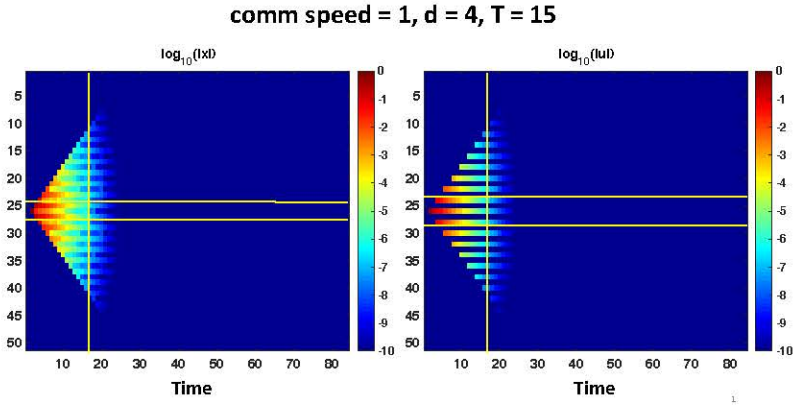

Fig. 13: With the dynamics propagating at the same speed as the controllers communicate and with such sparse actuation, we are now unable to localize in space and time as desired. However, the control action and state deviation is reasonably well contained.

[2] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, Aug 1989.

[3] A. Lamperski and J. C. Doyle, "Output feedback $\mathcal{H}_2$ model matching for decentralized systems with delays," in *2013 IEEE American Control Conference (ACC)*, June 2013.

[4] Y.-C. Ho and K.-C. Chu, "Team decision theory and information

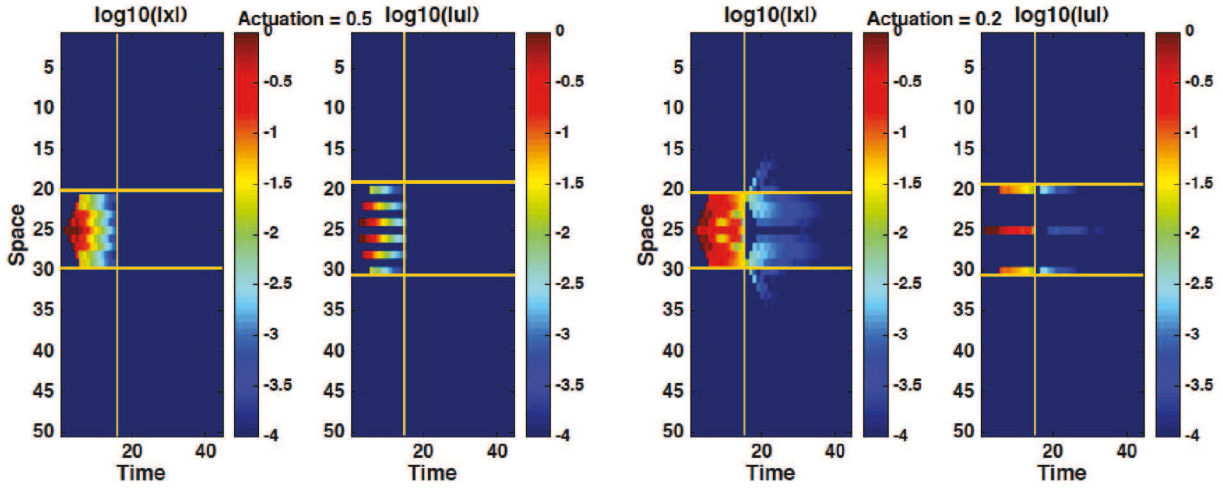

Fig. 14: As the number of actuators available for control decreases it becomes more difficult to obtain a localized response. The $\mathcal{H}_2$-norm for the actuation = 0.5 is 79.5, and for actuation = 0.2 it is 97.52 with, $\|\Delta\|_{\mathcal{E}_1} = 0.25$.

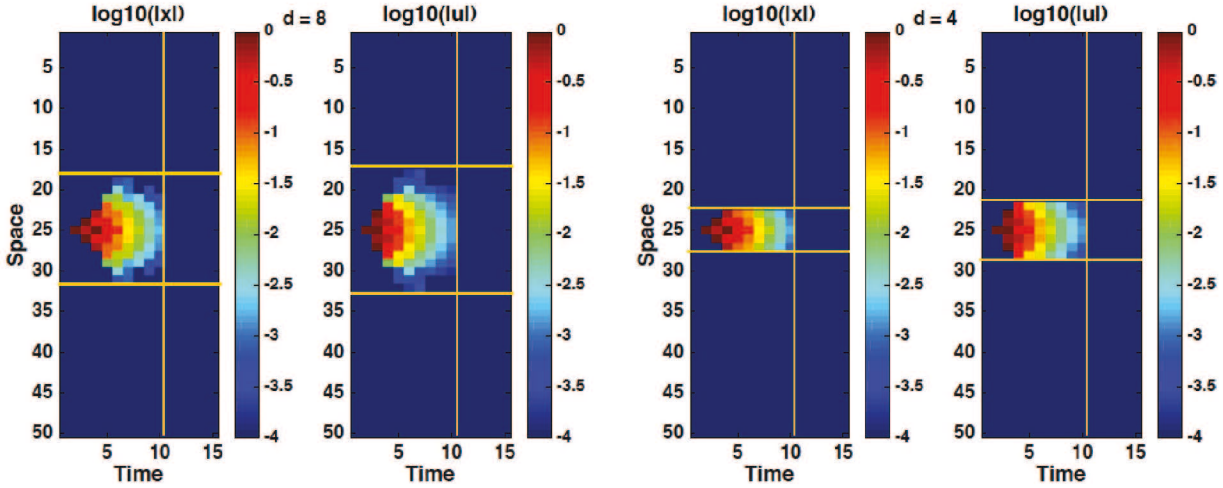

Fig. 15: The d-hop sparsity is decreased from $d = 8$ to $d = 4$. The closed loop $\mathcal{H}_2$-norm is 72.57 and 72.59 for $d = 8$ and $d = 4$ respectively. No virtual localization was required.

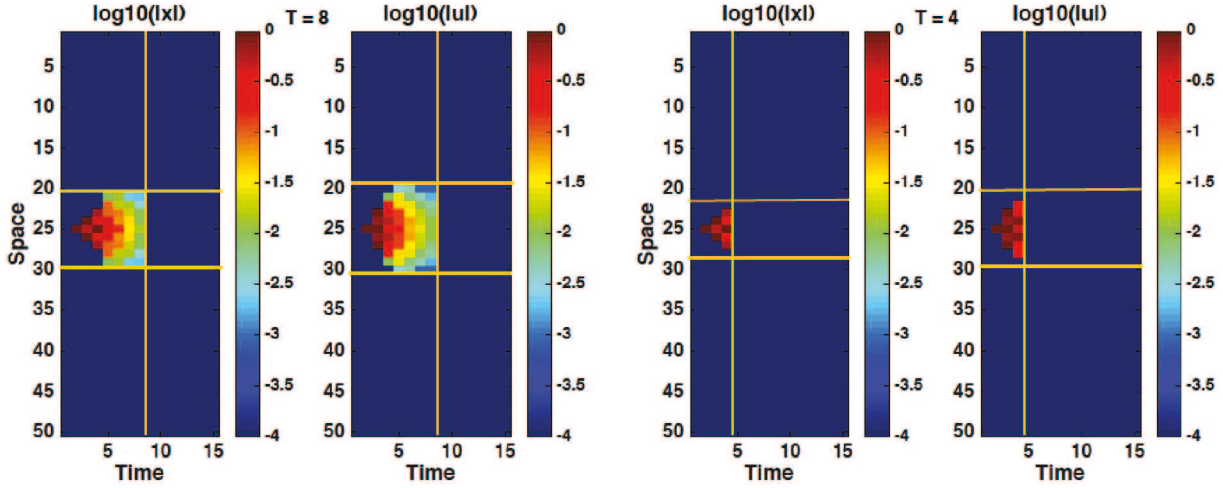

Fig. 16: The FIR Horizon length is decreased from $T = 8$ to $T = 4$. In neither case was virtual actuation required. The $\mathcal{H}_2$-norm was 72.57 for $T = 8$ and 72.96 for $T = 4$.

structures in optimal control problems–part i," *Automatic Control, IEEE Transactions on*, vol. 17, no. 1, pp. 15–22, 1972.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] H. S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM Journal of Control*, vol. 6, no. 1, 1968.

[11] 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 (ACC)*, June 2014.

[12] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized LQR optimal control," in *2014 53rd IEEE Conference on Decision and Control (CDC)*, 2014.

[13] 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.

[14] 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.

[15] Y.-S. Wang and N. Matni, "Localized LQG optimal control for large-scale systems," in *2016 IEEE American Control Conference (ACC)*, 2016.

[16] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized LQR control with actuator regularization," in *2016 IEEE American Control Conference (ACC)*, 2016.

[17] Y. S. Wang, "Localized lqr with adaptive constraint and performance guarantee," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 2769–2776.

[18] N. Matni, Y.-S. Wang, and J. Anderson, "Scalable system level synthesis for virtually localizable systems," in *2017 56th IEEE Conference on Decision and Control (CDC), submitted.*, 2017.

[19] Y.-S. Wang, N. Matni, and J. C. Doyle, "A system level approach to controller synthesis," *submitted to IEEE Transactions on Automatic Control*, 2017. [Online]. Available: \url{https://arxiv.org/abs/1610.04815}

[20] ——, "Separable and localized system level synthesis for large-scale systems," *IEEE Transactions on Automatic Control, Conditionally Accepted*, 2017. [Online]. Available: \url{https://arxiv.org/abs/1701.05880}

[21] ——, "System level parameterizations, constraints and synthesis," *IEEE American Control Conference, to appear*, 2017.

[22] J. Anderson and N. Matni, "Structured state space realizations for sls controllers," in *IEEE 55th Annual Allerton Conference on Communication, Control, and Computing*, 2017.

[23] N. Matni and V. Chandrasekaran, "Regularization for design," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2016.

[24] N. Matni, "Communication delay co-design in h2 distributed control using atomic norm minimization," *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2015.

[25] 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.

[26] ——, "Convexity of decentralized controller synthesis," *IEEE Transactions on Automatic Control, To appear*, 2016. [Online]. Available: http://arxiv.org/pdf/1305.5859v2.pdf

[27] 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.

[28] G. E. Dullerud and F. Paganini, *A Course In Robust Control Theory: A Convex Approach*. Springer-Verlag, 2000.

[29] M. A. Dahleh and Y. Ohta, "A necessary and sufficient condition for robust bibo stability," *Syst. Control Lett.*, vol. 11, no. 4, pp. 271–275, Oct. 1988. [Online]. Available: http://dx.doi.org/10.1016/0167-6911(88)90070-9