A Scalable Localized Approach to Filtering and Robust, Adaptive Controller Synthesis for Distributed Networked Systems

SooJean Han and John C. Doyle*

February 13, 2020

Abstract

This manuscript is intended to thoroughly survey a handful important applications and extensions to the recently proposed System Level Synthesis (SLS) framework for synthesizing localized and distributed controllers for large-scale networks of systems. First, the closely-related filtering problem is studied in the context of SLS, giving rise to the localized and distributed Kalman filtering (LDKF) method, which is essentially a scalable implementation of the celebrated Kalman filter. Second, an extension is made to robust and adaptive SLS, which designs controllers which are capable of stabilizing the network despite parametric uncertainties and/or dynamically-changing network topologies is discussed. Moreover, due to the robust nature of the control design scheme, learning these uncertainties precisely (i.e., the exact value of a parametric uncertainty or the exact locations of network connectivity changes) is not necessary to stabilize the overall network. Rather, the uncertainty sets are reduced over time through an adaptation scheme, but they are reduced only until there exists a robust controller which can stabilize the system for all values within the current uncertainty set.

1   Introduction

System Level Synthesis (SLS) is a unified framework for the controller synthesis of linear, discrete-time networked systems subject to realistic communication delay constraints. What makes the SLS approach appealing is its shift of the synthesis task from designing controllers to instead designing the entire closed-loop response, which provides important advantages such as scalability to larger networks.

Many extensions to SLS and its applications have arisen since [28]. Localized and distributed Kalman filtering (LDKF) was presented in [30], as an extension of the traditional centralized Kalman filtering

---

*SJ. Han and J. C. Doyle are with the Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA. Corresponding email: `soojean@caltech.edu`

method to networked systems of large scale. Employing the localized implementation of SLS, the LDKF method is able to design filters for large-scale networks of systems, a task which may not even be tractable using the traditional Kalman filtering method. Furthermore, LDKF is able to design such filters with performance comparable to that of the centralized Kalman filter (i.e., the accuracy of the estimate is the same between the two approaches when tested on networks where both methods are computationally tractable). SLS has additionally been developed into a localized, distributed robust control scheme for large-scale networks which additionally adapts to two separate types of uncertainties. First, each system estimates parametric uncertainties for its own local region independently of other systems in the network, allowing for a distributed implementation that can be mostly parallelized. Second, each system accounts for uncertainties in the topological structure of the network; this subsequently leads to an iterative scheme in which the system is able to handle multiple topological modifications over time.

In (the second half of) this manuscript, we focus on surveying the work on these two major extensions of the SLS framework and incorporate new, recent results for each respective topic. The first half of the paper will be dedicated to surveying relevant results from systems level synthesis, distributed control, graph theory, and consensus algorithms; past work related to each of the two extensions will also be discussed.

1.1   Previous and Related Work

System Level Synthesis: A parameterization of the set of internally-stabilizing controllers served as te basis for many controller synthesis methods. To achieve this in the case of linear time-invariant systems, the seminal work of [32] introduced the Youla parameterization approach, and [26] introduced the related stable factorization approach. Parameterizations such as these allow us to formulate the synthesis procedure as convex optimization problems [3] or linear programming problems [8]. This further allows us to incorporate miscellaneous controller design specifications by imposing them as additional constraints on the optimization problem (as long as they are convex). This ability to synthesize controllers through a principled optimization problem rather than a loop-at-a-time tuning process ultimately made way for foundational work on robust control and optimal control [12].

As present-day systems grow larger in scale with a rising amount of computational power, the degree of complexity in system dynamics and their interactions which the environment are also increasing. As such, the mentioned optimization problem for controller synthesis, which were typically implemented as a centralized procedure, may no longer be tractable to solve. This motivates the system-level parameterization, which essentially performs synthesis for the entire closed-loop response instead of solely the controller. The subsequent systems level approach was proposed and demonstrated in [28, 29, 13, 19]. The primary advantage of the systems level approach is it applicability to large-scale networks. This arises from the system level parameterization, which allows for decomposition of the synthesis optimization problem into multiple smaller subproblems, localized around each subsystem of the large-scale network. The recent survey paper [1] summarizes the systems level technique as well as a robust systems level design strategy for systems which may not satisfy some stringent constraints that the usual systems level approach imposes. Here, we will focus on the extensions which have been made

Distributed filtering and sensor fusion: Localized distributed Kalman filtering (LDKF) was a Kalman-like filtering method proposed by [30] as a way to extend the traditional Kalman filtering method to systems of large scale. Many distributed filtering methods have been proposed in the past, including the work of [21], the Kalman consensus filter [22], and the diffusion-based filtering approach of [4]. A distributed Bayesian filtering approach using maximum (or log) likelihood consensus was proposed in [2] as well. However, such approaches typically require each subsystem to store the entire plant network instead of just the local subset relevant to it, which leads to a major computational bottleneck. Instead, LDKF leverages the localized implementation of SLS, and is thus able to synthesis filters for large-scale networks which may not even be tractable for many of the aforementioned approaches. Furthermore, it was shown that LDKF is able to design such filters with performance comparable to that of the centralized Kalman filter (i.e., the accuracy of the estimate is the same between the two approaches when tested on networks where both methods are computationally tractable).

Adaptive control and dynamic network structures: Traditional adaptive control approaches [18, 25] and subsequent adaptive control algorithms which combine elements from data-driven control and machine learning [9, 10, 11] have been studied in a wide variety of settings. However, scalability to larger systems is not one of those settings. To this end, a systems-level approach of designing robust controllers which simultaneously adapts to uncertainties in parameters is proposed in [17]. Here, we develop a systems-level approach to handle uncertainties in the topology of the network, and propose a control law design scheme that is robust to changes in the network topology while locally and adaptively learning the locations at which the true topology differs from the assumed nominal. In the specific case of the power grid, we are motivated by natural failures such as power lines downed by severe weather conditions, and the subsequent need to upgrade the control scheme in an efficient way that does not involve redesigning from scratch. The interests of this study and the applications of the proposed scheme are not limited to just the grid. Collaborative tasks using distributed robotic systems that communicate with each other wirelessly (such as formation flying or information gathering) clearly necessitates a robust method of control to handle the highly dynamic communication network as the agents move about independently. In this direction, previous consensus-based approaches have been considered in cooperative robot control [6, 7] and oscillator synchronization [24], to name a few examples. Consensus-based methods of parameter estimation and sensor fusion have also been proposed [31, 23].

There exists a rich literature of work on the treatment of distributed networked control systems with dynamic structures; see [16], Table 2 for a comprehensive survey. [15] considers consensus for fault-detection in sensor networks whose topological structure switches according to a Markov chain. Event-triggered and sample-based consensus approaches for collections of systems arranged in a dynamic network have been studied in [14, 5, 20] and criterion for convergence are also provided. Such approaches have a lot of practical applicability, e.g., multiagent systems, which are not physically interconnected, perform their tasks independently of each other unless they meet and aggregate their observations. However, a primary difference between this branch of literature and the work presented here is that the true topology and/or uncertain parameters do not necessarily need to be fully determined prior to synthesizing a controller for it due to the ability of the systems level approach to designing robust localized controllers [19]. Hence, the performance of the consensus algorithm is less important than the performance of the stabilizing controller.

2 Notation

We use N_s to denote the number of systems in network, N_x (a multiple of N_s) to denote the total number of states, and $N_u \leq N_x$ as the number of control inputs. Denote the network of systems by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with adjacency matrix $G \in \mathbb{R}^{N_s \times N_s}$ with vertex set $\mathcal{V} = \{1, 2, \cdots N_s\}$. Note that an undirected graph implies that G is a symmetric matrix. For two systems $i, j \in \mathcal{V}$, we have that j belongs in the set $\mathcal{N}(i)$, called the *neighbor set* of i, iff $(i, j) \in \mathcal{E}$. We say that the graph $\mathcal{G}$ is *connected* if there exists a path (i.e. a sequence of edges in $\mathcal{E}$) between every pair of nodes $i, j \in \mathcal{V}$.

3 A Brief Review of the Systems Level Approach

3.1 The Centralized Formulation

We will consider specific plants of the following linear, discrete-time form:

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

where $\mathbf{x} \in \mathbb{R}^{N_x}$ denotes the state, $\mathbf{u} \in \mathbb{R}^{N_u}$ the control action, $\mathbf{w} \in \mathbb{R}^{N_w}$ the external disturbance, $\mathbf{y} \in \mathbb{R}^{N_y}$ the measurement, and $\bar{\mathbf{z}}$ the regulated output.

We desire to find a controller K which stabilizes the open-loop system when arranged in the feedback interconnection shown in Figure 1 such that the closed-loop system behavior minimizes some cost or error functional.
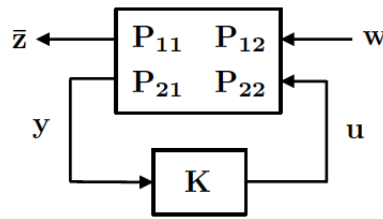

Figure 1: basic plant model

We can obtain a direct relationship between the disturbance $\mathbf{w}$ and the regulated output $\bar{\mathbf{z}}$. Apply the z-transform to the dynamics (1):

$$\mathbf{x}[t+1] = A\mathbf{x}[t] + B_1 w[t] + B_2 u[t] \implies z\mathbf{x} = A\mathbf{x} + B_1 w + B_2 u \implies \mathbf{x} = (zI - A)^{-1}(B_1 w + B_2 u)$$

$$\begin{cases} \bar{\mathbf{z}}[t] = C_1\mathbf{x}[t] + D_{11}\mathbf{w}[t] + D_{12}\mathbf{u}[t] \\ y[t] = C_2\mathbf{x}[t] + D_{21}\mathbf{w}[t] + D_{22}\mathbf{u}[t] \end{cases} \implies \begin{cases} \bar{\mathbf{z}} = (C_1(zI - A)^{-1}B_1 + D_{11})\mathbf{w} + (C_1(zI - A)^{-1}B_2 + D_{12})\mathbf{u} \\ \mathbf{y} = (C_2(zI - A)^{-1}B_1 + D_{21})w + (C_2(zI - A)^{-1}B_2 + D_{22})\mathbf{u} \end{cases}$$

Let $P_{ij} = C_i(zI - A)^{-1}B_j + D_{ij}$ so that we can write:

$$\begin{bmatrix} \bar{\mathbf{z}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ K\mathbf{y} \end{bmatrix}$$
$$\implies (I - P_{22}K)\mathbf{y} = P_{21}\mathbf{w}$$
$$\implies \bar{\mathbf{z}} = P_{11}\mathbf{w} + P_{12}K\mathbf{y} = (P_{11} + P_{12}K(I - P_{22})^{-1}P_{21})\mathbf{w}$$

The goal is to determine a K that will stabilize plant P and the cost/error we seek to minimize is a suitably chosen norm of the closed-loop transfer function from $\mathbf{w}$ to $\bar{\mathbf{z}}$. The **centralized optimal control formulation** is then as follows:

$$\min_K \left\| P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \right\| \tag{2}$$
$$\text{s.t. } K \text{ internally stabilizes } P$$

Two key assumptions are made about the system so that the above optimization problem is solvable:

1. The interconnection in Figure 1 is well-posed, meaning that all closed-loop transfer functions from each input to each output are proper. This further implies that the matrix $I - D_{22}D_k$ must be invertible.

2. Both the plant and controller realizations are stabilizable and detectable. This means that:

   - (A, B_2) and (A_k, B_k) are stabilizable: even though we cannot drive every state to their desired values, the components that are uncontrollable can at least be guaranteed to be stable and not problematic – a slightly weaker notion of controllability.

   - (A, C_2) and (A_k, C_k) are detectable: even though we cannot observe the value of every state, the ones that are unobservable can at least be guaranteed to be stable, hence not problematic – a slightly weaker notion of observability

3.2 System Level Parameterization

There are two main issues with the centralized formulation. First, the optimization problem (2) is nonconvex in terms of K, which makes it difficult to solve. Second, it may be intractable to solve in the case of large-scale systems (i.e., N_x is very large). This motivates the *system-level parameterization*, which essentially computes the entire closed-loop response instead of just the controller. In order to achieve this, K is allowed to have internal dynamics as well, instead of being simply constant:

$$\xi[t + 1] = A_k\xi[t] + B_k\mathbf{y}[t] \tag{3a}$$
$$\mathbf{u}[t] = C_k\xi[t] + D_k\mathbf{y}[t] \tag{3b}$$

where ξ represents the internal state of the controller. Since the control law is given by $\mathbf{u} = K\mathbf{y}$, we can express $K = C_k(zI - A_k)^{-1}B_k + D_k$ by again taking the z-transform and determining the transfer function between $\mathbf{u}$ and $\mathbf{y}$.
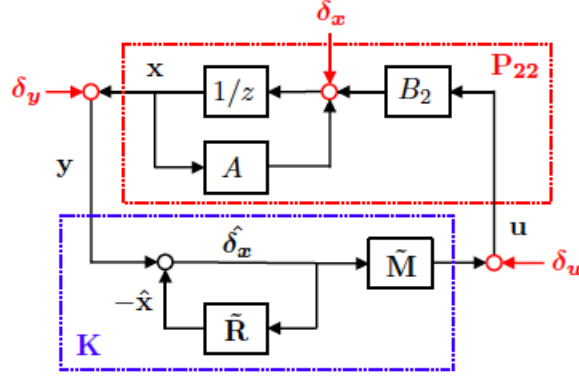
Fig. 3. The proposed state feedback controller structure, with $\tilde{\mathbf{R}} = I - z\mathbf{R}$ and $\tilde{\mathbf{M}} = z\mathbf{M}$.

Figure 2: Basic plant model for state feedback controller synthesis ([29]). Placeholder figure.

We will first look at the state feedback case, for the system (1) with $C_2 = I$ and $D_{21} = D_{22} = 0$:

$$\mathbf{x}[t+1] = A\mathbf{x}[t] + \mathbf{w}[t] + B\mathbf{u}[t] \tag{4a}$$

$$\bar{\mathbf{z}}[t] = C_1\mathbf{x}[t] + D_{11}\mathbf{w}[t] + D_{12}\mathbf{u}[t] \tag{4b}$$

$$\mathbf{y}[t] = \mathbf{x}[t] \tag{4c}$$

With feedback control law written as $\mathbf{u} = K\mathbf{x}$ and expressing the z-transform, we get:

$$(zI - A)\mathbf{x} = BK\mathbf{x} + \mathbf{w} \implies \mathbf{x} = (zI - A - BK)^{-1}\mathbf{w}$$
$$u = K\mathbf{x} = K(zI - A - BK)^{-1}\mathbf{w}$$

We then define the system response to be the transfer function matrices $\{\Phi_x, \Phi_u\}$ such that $\mathbf{x} = \Phi_x\mathbf{w}, \mathbf{u} = \Phi_u\mathbf{w}$, where $\Phi_x = (zI - A - BK)^{-1}$ and $\Phi_u = K(zI - A - BK)^{-1}$. This allows us to express the controller as $K = \Phi_u\Phi_x^{-1}$. However, computing the inverse Φ_x^{-1} will be difficult for large-scale systems, so instead of directly evaluating K, we arrange the blocks $\{\Phi_x, \Phi_u\}$ in the feedback interconnection shown in Figure 2. This arrangement further gives us the nice property that any sparsity structure imposed on the matrices $\{\Phi_x, \Phi_u\}$ will also be reflected in the structure of the controller K, a main advantage that will allow us to move from centralized controllers to distributed ones for systems of large scale.

We have the following algebraic characterization of the set of state-feedback system responses $\{\Phi_x, \Phi_u\}$ that are achievable by an internally stabilizing controller K.

Theorem 1 (State-Feedback System Level Parameterization)**.** The following results hold:

a. The affine subspace defined by

$$\begin{bmatrix} zI - A & -B \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \tag{5a}$$

$$\Phi_x, \Phi_u \in \frac{1}{z}\mathcal{RH}_\infty \tag{5b}$$

6

parametrizes all system responses from state disturbance $\mathbf{w}$ to (x, u) that give rise to achievable and internally stabilizing controllers.

b. Such controllers K are determined by $K = \Phi_u \Phi_x^{-1}$ and the system responses $\Phi_x = (zI - A - BK)^{-1}, \Phi_u = K(zI - A - BK)^{-1}$ are achieved.

Hence, any $\{\Phi_x, \Phi_u\}$ that satisfies (5) will yield an internally stabilizing controller $K = \Phi_u \Phi_x^{-1}$ for (4) that also gives us the desired system response. The types of desired system response include disturbance rejection (i.e, maintaining consensus among a network of subsystems despite a sudden entrance of $\mathbf{w}$ into one of the subsystems) or trajectory following/reference tracking.

To implement the closed-loop system, recall that signals in discrete-time can be represented by a linear combination of impulse signals. We solve a cost optimization problem similar to (2) and minimize over the *spectral component matrices* $\{\Phi_x[k], \Phi_u[k]\}$ corresponding to the system responses:

$$\Phi_x(z) = \sum_{k=1}^{T} \Phi_x[k] \frac{1}{z^i}, \;\; \Phi_u(z) = \sum_{k=1}^{T} \Phi_u[k] \frac{1}{z^i}$$

From the block diagram Figure 2, we can derive the following formulas for our signals of interest. First, the control law:

$$\mathbf{u}(z) = \tilde{\Phi}_u(z)\hat{\mathbf{w}}(z) = z\Phi_u(z)\hat{\mathbf{w}}(z) \implies \mathbf{u}[t] = (\tilde{\Phi}_u * \mathbf{w})[t] = \sum_{k=0}^{T} \Phi_u[k]\hat{\mathbf{w}}[t-i]$$

because converting from z-domain to the time domain turns multiplication into convolution.

Next, the state vector can be written:

$$\hat{\mathbf{w}}(z) = \mathbf{x}(z) + \tilde{\Phi}_{\mathbf{x}}\hat{\mathbf{w}}(z) = \mathbf{x}(z) + (I - \Phi_x(z))\hat{\mathbf{w}}(z)$$

$$\implies \mathbf{x}(z) = \Phi_x(z)\hat{\mathbf{w}}(z) \implies \mathbf{x}[t] = (\Phi_x * \mathbf{w})[t] = \sum_{k=0}^{T} \Phi_x[k]\hat{\mathbf{w}}[t-i]$$

$$\implies \mathbf{x}[t+1] = (\Phi_x * \mathbf{w})[t+1] = \sum_{k=0}^{T} \Phi_x[k]\hat{\mathbf{w}}[t+1-i]$$

Overall, the state-feedback controller is implemented as follows:

$$\hat{\mathbf{x}}[t] = \sum_{k=2}^{T} \Phi_x[k]\hat{\mathbf{w}}[t+1-k] \tag{6a}$$

$$\hat{\mathbf{w}}[t] = \mathbf{x}[t] - \hat{\mathbf{x}}[t] \tag{6b}$$

$$\mathbf{u}[t] = \sum_{k=1}^{T} \Phi_u[k]\hat{\mathbf{w}}[t+1-k] \tag{6c}$$

It was shown in [19] that even when this relationship is approximately satisfied, the implementation as in (6) produces a stable closed-loop response.
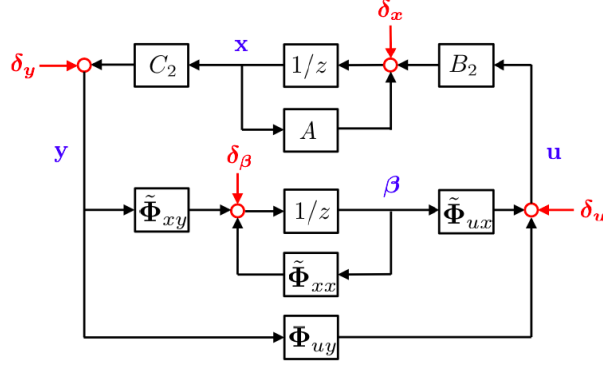
7

Figure 3: Output feedback controller structure with $\tilde{\Phi}_{xx} := z(I - z\Phi_{xx})$, $\tilde{\Phi}_{ux} := z\Phi_{ux}$, and $\tilde{\Phi}_{xy} := -z\Phi_{xy}$ (adapted from [29]). Placeholder figure.

Now we will consider the output-feedback case. We will restrict our attention to when $D_{22} = 0$ in (1), so that the output dynamic is not affected by the control input u. The overall plant dynamics become:

$$\mathbf{x}[t+1] = A\mathbf{x}[t] + Bu[t] + \mathbf{w}[t]$$
$$y[t] = C\mathbf{x}[t] + \mathbf{v}[t]$$

Substituting the output-feedback control law $\mathbf{u} = K\mathbf{y}$ into the z-transform of the dynamics yields $(zI - A - BKC)\mathbf{x} = \mathbf{w} + BK\mathbf{v}$.

Similarly to the state-feedback case, we shall define the system response to be the matrices Φ_{xx}, Φ_{xy}, Φ_{ux}, Φ_{uy} such that:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix}$$

They are achieved by the controller K as follows:

$$\Phi_{xx} = (zI - A - BKC)^{-1}$$
$$\Phi_{xy} = KC\Phi_{xx}$$
$$\Phi_{ux} = \Phi_{xx}BK$$
$$\Phi_{uy} = K + KC\Phi_{xx}BK$$

and the full closed-loop architecture is given by Figure 3.

Like the state-feedback case, we have a main result for the algebraic characterization of all achievable system responses $\{\Phi_{xx}, \Phi_{xy}, \Phi_{ux}, \Phi_{uy}\}$.

Theorem 2 (Output-Feedback System Level Parameterization)**.** The following results hold:

a. The affine subspace defined by

$$\begin{bmatrix} zI - A & -B \end{bmatrix} \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix}$$

$$\begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} zI - A \\ -C \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix}$$

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

parametrizes all system responses from state disturbance that give rise to achievable and internally stabilizing controllers.

b. Such controllers K are determined by $K = \Phi_{uy} - \Phi_{ux}\Phi_{xx}^{-1}\Phi_{xy}$ and the system responses $\Phi_{xx} = (zI - A - BKC)^{-1}, \Phi_{ux} = KC\Phi_{xx}, \Phi_{xy} = \Phi_{xx}BK, \Phi_{uy} = K + KC\Phi_{xx}BK$ are achieved.

3.3 Spatial/Temporal Locality and Delays

Let the network be denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ the set of vertices (subsystems) and $\mathcal{E}$ the set of unweighted, directed edges. The distance metric $d(v_i \to v_j)$ on $\mathcal{G}$ is the shortest path from v_i to v_j where $v_i, v_j \in \mathcal{V}$. The direction of the arrow must be noted because the edges are directed.

Locality is measured in two ways: spatially and temporally.

1. **Spatial locality** ensures that the disturbance is only felt within a neighborhood of the entrance. To model this as a constraint, we introduce a few definitions.

   Definition 1. Denote $\Phi_{x,ij}$ to be the transfer matrix describing the perturbation from disturbance component w_j (which directly affects system j) to the state estimate x_i of system i. Then the map Φ_x is *d-localizable* iff for every pair of systems $i, j \in \mathcal{V}$, $\Phi_{x,ij} = 0$ for all i such that $\text{dist}(i, j) > d$. Analogously, the definition for d-localizability of the map Φ_u to the control law u_i follows the same way. Accordingly, we will associate a local *d-hop set* with each system i as the set of systems j where the (i, j)th entry of G^d is nonzero.

   Informally, a system is d-localizable if the ith system in the network can construct a controller using a plant model which is limited to the neighborhood of systems within distance d of i and implemented using information signals that are gathered only from this same neighborhood.

   We denote the subspace $\mathcal{L}_d$ as the d-localized constraint set. It constrains Φ_x to be d-localized and Φ_u to be $(d+1)$-localized. This forces any disturbance to be rejected in finite space(i.e., so that it does not propagate throughout the entire network). This is implemented by imposing sparsity patterns on each spectral factor of Φ_x and Φ_u:

   $$\text{supp}\left(\Phi_x[t]\right) = \bigcup_{s=1}^{t} \frac{1}{z^s}\text{supp}\left(A^{\min(d, \lfloor t_c(s-1)\rfloor)}\right)$$

   $$\text{supp}\left(\Phi_u[t]\right) = \bigcup_{s=1}^{t} \frac{1}{z^s}\text{supp}(B^T)\text{supp}\left(A^{\min(d+1, \lfloor t_c(s-1)\rfloor)}\right)$$

   where t_c is the communication speed of the controller network relative to the plant network, and $\text{supp}(\cdot)$ denotes the support operator. Clearly, if $t_c < 1$, it would be impossible to localize the system because the controller is not reacting quickly enough to prevent the disturbance effects from propagating throughout the plant.

2. **Temporal locality** ensures that the system response Φ_x, Φ_u has a finite impulse response of horizon length T. This forces the disturbance to be rejected in finite time. We enforce the

constraint as $\{\Phi_x, \Phi_u\} \in \mathcal{F}_T$, where

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

The z-transform affine constraint can then be expressed in the time-domain as follows:

$$\begin{bmatrix} I & -A & & & \\ & I & -A & & \\ & & \ddots & \ddots & \\ & & & & -A \\ & & & & I \end{bmatrix} \begin{bmatrix} \Phi_x[T+1] \\ \Phi_x[T] \\ \vdots \\ \Phi_x[2] \\ \Phi_x[1] \end{bmatrix} = \begin{bmatrix} B\Phi_u[T] \\ B\Phi_u[T-1] \\ \vdots \\ B\Phi_u[1] \\ I \end{bmatrix}$$

where the fact that the matrix sizes and vector lengths are limited by T is a result of the finite time-horizon constraint.

Thus, in each of the subproblems described in the previous subsection, we can express the constraint as membership in the set $\mathcal{S} = \mathcal{L}_d \cap \mathcal{F}_T$. Systems that satisfy both spatial and time constraints simultaneously with d-hops and horizon T are said to be (d, T)-*localizable*.

3.4   System Level Synthesis

Now, instead of solving (2), the desired closed-loop behavior can be achieved by enforcing appropriate convex constraints on $\{\Phi_x, \Phi_u\}$ and solving the optimal control problem of the form:

$$\min_{\{\Phi_x, \Phi_u\}} f(\Phi_x, \Phi_u, Q, R) \tag{8a}$$

$$\text{s.t. } \{\Phi_x, \Phi_u\} \in \mathcal{S} \tag{8b}$$

where $Q \in \mathbb{R}^{N_x \times N_x}, R \in \mathbb{R}^{N_u \times N_u}$ are cost matrices which assign weight to Φ_x, Φ_u respectively, f is a convex function of both Φ_x and Φ_u (typically a weighted norm or LQR cost), and $\mathcal{S}$ is a convex set which encodes the constraints. Of particular interest, especially in distributed, decentralized implementations, is when $\mathcal{S}$ includes the spatial and temporal constraints described in 3.3.

4 Localized, Distributed Kalman Filters

When the standard LQR problem is formulated in terms of (8), it is known as the **localized linear quadratic regulator** (LLQR), and is expressed as follows:

$$\min_{\{\Phi_x[k],\Phi_u[k]\}_{k=1}^T} \sum_{k=1}^T \text{tr}\left(\Phi_x[k]^T Q \Phi_x[k] + \Phi_u[k]^T R \Phi_u[k]\right)$$

$$\text{s.t.} \begin{bmatrix} I & -A & & & \\ & I & -A & & \\ & & \ddots & \ddots & \\ & & & & -A \\ & & & & I \end{bmatrix} \begin{bmatrix} \Phi_x[T+1] \\ \Phi_x[T] \\ \vdots \\ \Phi_x[2] \\ \Phi_x[1] \end{bmatrix} = \begin{bmatrix} B\Phi_u[T] \\ B\Phi_u[T-1] \\ \vdots \\ B\Phi_u[1] \\ I \end{bmatrix}$$

$$\text{supp}\left(\Phi_x[k]\right) = \bigcup_{j=1}^k \frac{1}{z^j} \text{supp}\left(A^{\min(d,\lfloor t_c(j-1)\rfloor)}\right)$$

$$\text{supp}\left(\Phi_u[k]\right) = \bigcup_{j=2}^k \frac{1}{z^j} \text{supp}(B^T)\text{supp}\left(A^{\min(d+1,\lfloor t_c(j-1)\rfloor)}\right)$$

(d,T)-localizability allows us to decompose the constraints into smaller subconstraints. Furthermore, the objective function may also be decomposed, as the following algebraic manipulation shows:

$$\sum_{k=1}^T \text{tr}\left(\Phi_x[k]^T Q \Phi_x[k] + \Phi_u[k]^T R \Phi_u[k]\right) = \sum_{i=1}^{N_x}\sum_{k=1}^T \text{tr}\left(\Phi_x[k]^T Q \Phi_x[k] e_i e_i^T\right) + \text{tr}\left(\Phi_u[k]^T R \Phi_u[k] e_i e_i^T\right)$$

$$= \sum_{i=1}^{N_x}\sum_{k=1}^T \text{tr}\left(e_i^T \Phi_x[k]^T Q \Phi_x[k] e_i\right) + \text{tr}\left(e_i^T \Phi_u[k]^T R \Phi_u[k] e_i\right)$$

$$= \sum_{i=1}^{N_x}\sum_{k=1}^T e_i^T \Phi_x[k]^T Q \Phi_x[k] e_i + e_i^T \Phi_u[k]^T R \Phi_u[k] e_i$$

$$= \sum_{i=1}^{N_x}\sum_{k=1}^T (\Phi_x[k]^T)_i Q (\Phi_x[k])_i + (\Phi_u[k]^T)_i R (\Phi_u[k])_i$$

Here, we will formulate the LLQR for Kalman filtering. The system of interest is of the form

$$\mathbf{x}[k+1] = A\mathbf{x}[k] + \mathbf{w}[k] \tag{9a}$$
$$\mathbf{y}[k] = C\mathbf{x}[k] + \mathbf{v}[k] \tag{9b}$$

where $\mathbf{w}[k] \in \mathbb{R}^{N_x}, \mathbf{v}[k] \in \mathbb{R}^{N_y}$ are now Gaussian vectors with mean 0 and respective covariance matrices $Q = q * I_{N_x}$ and $R = r * I_{N_y}$ with $q, r > 0$.

Recall the Kalman filter state estimate update equation:

$$\hat{\mathbf{x}}[k+1|k] = A[k](\hat{\mathbf{x}}[k|k-1] + L[k](\mathbf{y}[k] - C[k]\hat{\mathbf{x}}[k|k-1]))$$

where $\mathbf{x}[k+1|k]$ denotes the state estimate at time $k+1$ given information obtained until time k.

For SLS to apply, we can derive an equivalent version of the affine constraints in Theorem 1. Taking the z-transform of the Kalman filter equation yields:

$$\hat{\mathbf{x}}[k+1|k] = A(\hat{\mathbf{x}}[k|k-1] + L[k]\tilde{\mathbf{y}}[k]) \implies (zI - A)\hat{\mathbf{x}} = L(y - C\hat{\mathbf{x}})$$
$$\implies (zI - A + LC)\hat{\mathbf{x}} = L\mathbf{y} = L(C\mathbf{x} + \mathbf{w})$$

Now take the z-transform of the system dynamics and add a term of $LC\mathbf{x}$ to both sides.

$$(zI - A + LC)\mathbf{x} = LC\mathbf{x} + \mathbf{w}$$

Subtracting the two z-transformed equations yields:

$$(zI - A + LC)\tilde{\mathbf{x}} = \mathbf{w} - L\mathbf{v}$$

where $\tilde{\mathbf{x}} := \mathbf{x} - \hat{\mathbf{x}}$ is the estimation error. Rewritten:

$$\tilde{\mathbf{x}} = \underbrace{(zI - A + LC)^{-1}}_{=:\Phi_w}\mathbf{w} + \underbrace{\left(-(zI - A + LC)^{-1}L\right)}_{=:\Phi_v}\mathbf{v}$$

and we can now treat Φ_w, Φ_v like our system response maps Φ_x, Φ_u. More specifically, $\Phi_w \in \mathbb{R}^{N_x \times N_x}$ represents the closed-loop transfer matrix from $\mathbf{w}$ to $\tilde{\mathbf{x}}$ and $\Phi_v \in \mathbb{R}^{N_y \times N_x}$ represents the closed-loop map from $\mathbf{v}$ to $\tilde{\mathbf{x}}$. By Theorem 2, the achievable set of observers can be parameterized by the affine subspace

$$\begin{bmatrix} \Phi_w & \Phi_v \end{bmatrix} \begin{bmatrix} zI - A \\ -C \end{bmatrix} = I$$
$$\Phi_w, \Phi_v \in \frac{1}{z}\mathcal{RH}_\infty$$

and we can implement the gain matrix $L = -\Phi_w^{-1}\Phi_v$.

To relate better to the characterization in Theorem 1, we can take the transpose across:

$$\begin{bmatrix} zI - A^T & -C^T \end{bmatrix} \begin{bmatrix} \Phi_w \\ \Phi_v \end{bmatrix} = I$$

After solving for internally stabilizing Φ_v and Φ_w, we can recover an estimate $\hat{\mathbf{x}}$ using the relationship $\hat{\mathbf{x}} = \mathbf{x} - \tilde{\mathbf{x}} = \mathbf{x} - \Phi_w\mathbf{w} - \Phi_v\mathbf{v}$. Alternatively, because one would not know what $\mathbf{x}, \mathbf{w}, \mathbf{v}$ are in real implementation, we write instead $\hat{\mathbf{x}} = \Phi_w B\mathbf{u} + \Phi_v\mathbf{y} = \Phi_v\mathbf{y}$, which is obtained directly from the z-transform of the Kalman filter equation and using $L = -\Phi_w^{-1}\Phi_v$. Note that because there is no control term in (9), we set $\mathbf{u} = 0$.

The overall LLQR problem that we must solve in order to obtain $\Phi_w[k], \Phi_v[k]$ for $k = 1, 2, \cdots, T$

such that any disturbances can be localized within d hops and time horizon T is:

$$\min_{\{\Phi_w[k],\Phi_v[k]\}_{k=1}^T} \sum_{k=1}^T \operatorname{tr}\left(\Phi_w[k]Q\Phi_w[k]^T + \Phi_v[k]R\Phi_v[k]^T\right) \tag{10a}$$

$$\text{s.t.} \quad \begin{bmatrix} I & -A^T & & & \\ & I & -A^T & & \\ & & \ddots & \ddots & \\ & & & -A^T \\ & & & & I \end{bmatrix} \begin{bmatrix} \Phi_w^T[T+1] \\ \Phi_w^T[T] \\ \vdots \\ \Phi_w^T[2] \\ \Phi_w^T[1] \end{bmatrix} = \begin{bmatrix} C^T\Phi_v^T[T] \\ C^T\Phi_v^T[T-1] \\ \vdots \\ C^T\Phi_v^T[1] \\ I \end{bmatrix} \tag{10b}$$

$$\operatorname{supp}\left(\Phi_w^T[k]\right) = \bigcup_{j=1}^k \frac{1}{z^j}\operatorname{supp}\left(A^{\min(d,\lfloor t_c(j-1)\rfloor)}\right) \tag{10c}$$

$$\operatorname{supp}\left(\Phi_v[k]\right) = \bigcup_{j=2}^k \frac{1}{z^j}\operatorname{supp}(C)\operatorname{supp}\left(A^{\min(d+1,\lfloor t_c(j-1)\rfloor)}\right) \tag{10d}$$

Simulation results with $N_s = N_y = 625, N_x = 625 * 2 = 1250$, and $d = 1, t_c = 2$, and $T = 5$ are presented in Figure 4 and Figure 5.
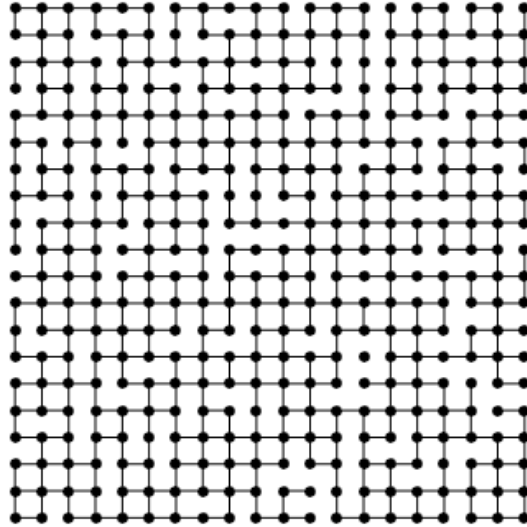

Figure 4: 20×20 mesh network of subsystems.

Compared against the centralized Kalman filter (which is implemented by stacking the matrices and estimating a $N_x \times 1$ state vector), the error of estimates for the first eight states (i.e, error of state estimate for the first four subsystems) are plotted below. There is a huge spike in the first few seconds in the distributed SLS implementation, but in steady-state, the performance against the Kalman filter is nearly identical. The real benefit of distributed SLS filtering over a centralized approach is the computation time; indeed, it takes the Kalman filter 19.3891 seconds to compute the estimate state trajectory, whereas the distributed SLS filter finishes in 3.2585 seconds. Thus, we are trading off a slight degradation in performance (though nearly similar) for a large boost in computation time.
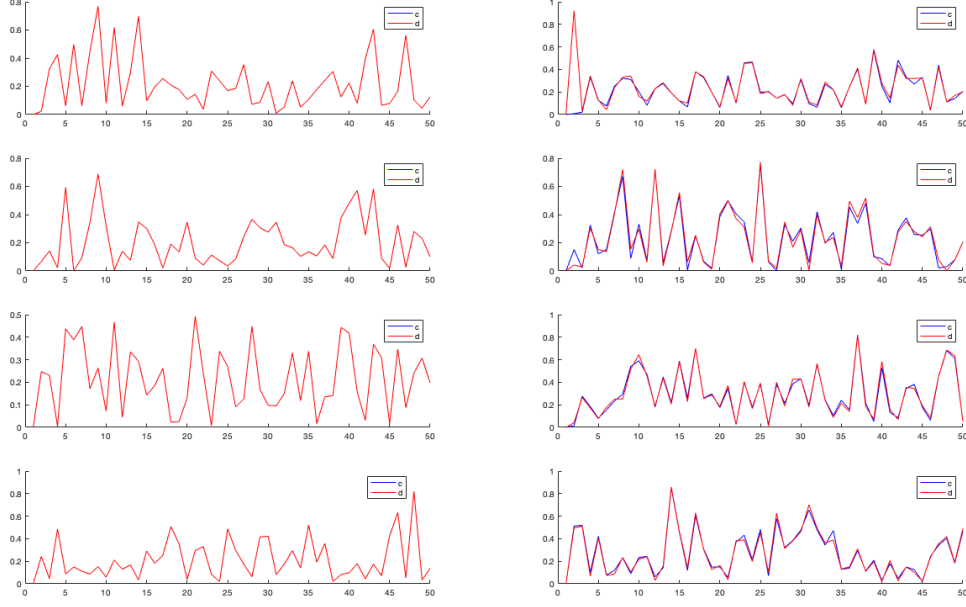
Figure 5: Error estimate for the first eight state components of the system.

5 Localized Robust and Adaptive Controllers

In this section, we will first present the centralized scheme for maintaining robust control over a topologically-varying network, then develop a localized extension which allows for computational scalability to networks of larger size. We will restrict our attention to the case where only the topological structure of the plant network varies according to link modifications while the controller network remains static. That is, the adjacency matrix G for A is varied while the B matrix remains the same.

5.1 Centralized Implementation

We begin with a nominal topological structure A^* of the network. We are aware that at least one link has been disconnected, and although we do not know which one(s), we are given a finite collection of K candidate link failure matrices D, one of which gives us the true topology $A = A^* + D$. This setup is consistent with real-world scenarios where we are oftentimes able to vaguely identify the local region in which a potential link failure has occurred. We assume that none of the candidate matrices causes the graph to become disconnected.

With this premise, the system dynamics are given by $\mathbf{x}[t+1] = (A^* + D)\mathbf{x}[t] + B\mathbf{u}[t] + \mathbf{w}[t]$. The matrix A^* denotes the known nominal system and link failures D enter in the form of perturbations to A^*.

To characterize the set of D, we introduce basis matrices $\mathcal{A}_l$ to encode all possible single-link modifications so that linear combinations can be used to model a general number of failures corresponding

14

to each candidate D. We will denote this set as $\mathcal{P}_0$, and formally refer to it as the initial *consistent set*.

$$\mathcal{P}_0 := \left\{ \sum_{l=1}^{M} \xi_l \mathcal{A}_l \; : \; \xi_l \in \{-1, 0, 1\} \right\} \tag{11}$$

where coefficient $\xi_l = 1$ is for when a link is added, $\xi_l = -1$ for when a link is deleted, $\xi_l = 0$ for when a link is unchanged. Because it is a discrete combinatorial set, we will impose $K << 2^M$ to make the problem tractable.

At each timestep, the consistent set is updated using new observations of $(\mathbf{x}[t+1], \mathbf{x}[t], \mathbf{u}[t])$:

$$\mathcal{P}_{t+1} := \left\{ D \in \mathcal{P}_t \; : \; \left\| \mathbf{x}[t+1] - \left(A^* + \sum_{l=1}^{M} \xi_l \mathcal{A}_l \right) \mathbf{x}[t] - B\mathbf{u}[t] \right\|_\infty \leq \eta \right\} \tag{12}$$

We will now use SLS to design the controller $\left\{ \Phi_x^{(t)}, \Phi_u^{(t)} \right\}$, where the superscript (t) is included to show that the control laws may change over time as more of the topology is learned. In the context of our topology adaptation problem, the following inequalities should be satisfied:

$$\sum_{k=1}^{T} \left\| \Delta_k(A', B, \Phi_x^{(t-1)}, \Phi_u^{(t-1)}) \right\| \leq \lambda_t \;\; \forall A' = A^* + D, \; D \in \mathcal{P}_t \tag{13a}$$

$$\left\| \sum_{k=1}^{T} \left(\Phi_x^{(t-1)} - \Phi_x^{(t)} \right)[k+1]\hat{\mathbf{w}}[t-k] \right\| \leq \gamma \tag{13b}$$

λ_t is referred to as the *robustness margin* and for each timestep t it determines whether the controller is stabilizable with the tth polytope of uncertainties. γ is the *adaptation margin* and ensures that the system response Φ_x doesn't fluctuate wildly with largely-varying $\mathbf{w}$.

The full optimization problem for centralized robust control which adapts to topological changes is hence presented:

$$\min_{\left\{ \Phi_x^{(t)}[k], \Phi_u^{(t)}[k] \right\}_{k=1}^{T}, \lambda_t} f\left(\Phi_x^{(t)}, \Phi_u^{(t)}, Q, R \right) = \begin{cases} \lambda_t & \text{if } \lambda_t \leq \lambda^* \\ \sum_{k=1}^{T} \left\| Q\Phi_x^{(t)}[k] + R\Phi_u^{(t)}[k] \right\|_1 & \text{else} \end{cases} \tag{14}$$
$$\text{s.t. } \{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{F}_T \text{ and } (13)$$

The two separate steps expressed in the objective function above are taken because optimizing for a performance objective is only reasonable if robust stability is feasible with uncertainty $\mathcal{P}_t$.

Remark 1. In implementation, the inclusion of (13b) to $\mathcal{S}$ is made optional. This is because the incorrect system response may be learned and closely adhered to for the rest of time if γ is chosen too small, resulting in an unstable controller. This is problematic in the case of topological uncertainties, where the sparsity patterns of all the candidate topologies D may be different.

5.2 Localized Implementation

A localized version of the algorithm essentially decomposes (14) into multiple independent subproblems. For system i, the submatrix A_i of consideration only includes the rows of A corresponding to the systems in $\mathcal{L}_d(i)$ (Definition 1). This means each system only keeps track of link modifications within its own local subset. Further, let d_c be the communication delay matrix between systems of the network, defined as $d_c(i, j) = |j - i|$ if $j \in \mathcal{L}_d(i)$, and ∞ otherwise.

Each system i begins with a local initial consistent set $\mathcal{P}_0^{(i)}$, defined the same way as in (11) but instead with M_i basis matrices $\mathcal{A}_l^{(i)}$ which have dimensions equal to A_i. Each consistent set is locally updated from $\mathcal{P}_t^{(i)}$ to $\mathcal{P}_{t+1}^{(i)}$ in a fashion similar to (12).

To design local controllers, we solve a local optimization problem of the form (8) for the ith columns of the system response matrices $\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}$. The constraints follow analogously to (13). Define the submatrix

$$\Delta_k^j\left(A, B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right) := \Phi_{x,ji}^{(t)}[k+1] - \sum_{l \in \mathcal{N}(j)} A_{jl}\Phi_{x,li}^{(t)}[k] - B_j\Phi_{u,ji}^{(t)}[k] \tag{15}$$

where $i, j \in \mathcal{V}$, $k = 1, \cdots, T$, and t iterates over the simulation time. This allows us to define our robustness margin constraints

$$\left\| \sum_{j \in \mathcal{L}_d(i)} \Delta_k^j\left(A', B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right) \right\| \le c_i\rho^{k-1} \quad \forall A' = A^* + D, D \in \mathcal{P}_t \quad \forall k \le T - 1 \tag{16a}$$

$$c_i \sum_{k=1}^{T} \rho^{k-1} \le \lambda_t^{(i)} + \epsilon, \ \epsilon \ge 0 \tag{16b}$$

Unlike the centralized formulation, instead of a constant, we introduce $\rho > 0$ and $c_i > 0$ to ensure faster exponential convergence to zero, which is motivated by the possibility of local disturbances propagating throughout the network in a cascading manner if it is not killed quickly enough within the local region.

The full optimization problem for localized robust, topologically-adaptive control is hence presented:

$$\min_{\left\{\Phi_x^{(t)}[k], \Phi_u^{(t)}[k]\right\}_{k=1}^{T}, \lambda_t, \epsilon} f\left(\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}, Q_i, R_i\right) = \sum_{k=1}^{T} \left\| Q\Phi_{x,i}^{(t)}[k] + R\Phi_{u,i}^{(t)}[k] \right\|_1 + r_c\epsilon \tag{17}$$

$$\text{s.t. } \{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{L}_d \cap \mathcal{F}_T \text{ and } (16)$$

Remark 2. The objective function described here is equivalent to the two-part objective function in (14). The slack variable ϵ (scaled by fixed $r_c > 0$) helps reduce the two-part process into a single step optimization problem; when ϵ is very large, the optimization problem effectively focuses on shrinking the polytope until ϵ is sufficiently reduced.

5.3 Iterative Multi-Stage Implementation

In this section, we present the iterative localized robust controller design for networks which switch between topological configurations arranged in a finite-state ergodic Markov Chain. The proposed

algorithm is a simple variation of the localized implementation from Section 5.2, namely with time-varying local sets $\mathcal{L}_d(i,t)$. The dynamics are now modeled as a time-varying hybrid system:

$$\mathbf{x}_i[t+1] = A_{ii}(\alpha(t))\mathbf{x}_i[t] + \sum_{j \in \mathcal{N}_i(\alpha(t))} A_{ij}\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t] \quad \forall i \in \mathcal{V} \tag{18}$$

where $\alpha(t) \in \mathbb{N}$ is a discrete-valued signal which switches with time and B is kept constant across all possible states of A. According to [23, 31], consensus among distributed systems is achievable with time-varying topologies, under conditions such as joint-connectedness among topologies that are visited infinitely many times. We will restrict our attention in this paper to hybrid dynamics where the signal α switches according to an ergodic Markov chain with a finite number K of states. Link modifications occur for a sequence of unknown times $\{T_1, T_2, \cdots\}$. The initial true system topology $A(\alpha(0))$ (with adjacency matrix $G(\alpha(0))$) is known.

Each subsystem keeps a nominal topology estimate $A^{(i)}(\alpha^*(t))$ and updates it whenever it detects that a switch has been made. The transition probability matrix P of the chain is unknown to the system, and each subsystem maintains an estimate $\hat{P}^{(i)}$, which it updates both locally and via simple averaging with the values of its other neighbors [31]. Since the methodology is the same across all subsystems $i \in \mathcal{V}$, the subscript i is henceforth removed for notational simplicity.

Similar to (11), the initial consistent sets are formed from M_k basis matrices $\mathcal{A}_\ell^{(i,k)}$ where $k = 1, \cdots, K$, $i \in \mathcal{V}$, and the collective modification is expressed as a linear combination of these bases. At each timestep t, an observation $\mathbf{x}[t]$ is made from the system (18). We identify which coefficients remain consistent with the system dynamics $(\mathbf{x}[t], \mathbf{x}[t-1], \mathbf{u}[t-1])$ by updating the consistent set in a fashion similar to (12) for each i. Because identification for each system i was only done using information local to i, additional consensus may be performed to further narrow down the consistent set in order to estimate the state of the Markov chain more precisely.

As before, it is most important to maintain system stability while this identification and consensus process is being done. To construct a topologically-robust controller $\{\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}\}$ for each system i, we simply solve the optimization problem (17) with the same communication delay matrix d_c defined in Section 5.2 and time horizon T.

5.4   Numerical Simulation

We simulate our iterative, localized scheme on a power grid network from [27], where the ith subsystems obeys the dynamics below. c_i is its inertia, b_i is a damping factor, w_i is the external disturbance, u_i is the control action, and Δt is the sampling time. The states are the relative phase angle $x_1^{(i)}$ between its rotor axis and external field, and its derivative, the frequency $x_2^{(i)}$.

$$\begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \end{bmatrix}[t+1] = \left(\begin{bmatrix} 1 & \Delta t \\ -\frac{a_i}{c_i}\Delta t & 1 - \frac{b_i}{c_i}\Delta t \end{bmatrix} + \sum_{j \in \mathcal{N}(i)} \begin{bmatrix} 0 & 0 \\ \frac{a_{ij}}{c_i}\Delta t & 0 \end{bmatrix} \right) \begin{bmatrix} x_1^{(j)} \\ x_2^{(j)} \end{bmatrix}[t] + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix}(w_i[t] + u_i[t])$$

We choose $\eta = 0.3$ and initial conditions $\mathbf{x}_i[0] = (3,0)^T$, $\mathbf{x}_i[0] = (0,0)^T$ $\forall i > 1$. Our hexagon network switches topologically according to a Markov chain with four states, as shown in Figure 6. The true values of the transition probabilities are $p_1 = 1 - p_2 = 0.4$ and $p_3 = 1 - p_4 = 0.8$, and are initially

unknown to each subsystem in the network. We assume that the entire set of states are known to each system, so that there are at most four consistent coefficients in $\mathcal{P}_t$ per system over all t. For the control law, we take $\rho = 0.7$, time horizon $T = 5$, and information from 1-hop regions about each system.
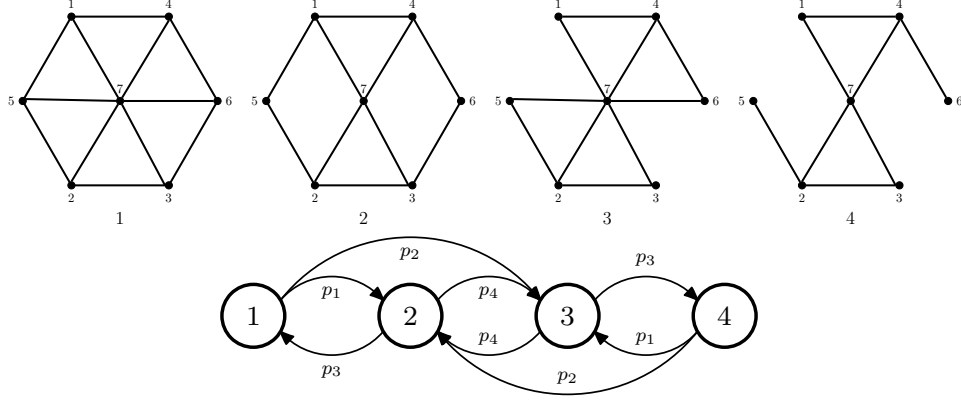

Figure 6: [Top] Hexagon topologies arranged in a Markov chain with four states. [Bottom] The state transition diagram for the four states.

In Figure 7, the black stems indicate the time of switching, and a height of $\frac{1}{2}s$ indicates a switch to state s. In the top figure, the number of consistent coefficients versus time is shown for subsystems $4, 5$, and 6. Each time a switch to a different configuration occurs, each subsystem resets with the entire set Markov chain states. In the bottom figure, each system estimates the actual topological configuration at the current time. Note that although each system manages to ultimately estimate the correct state, there is a slight time delay in when it achieves this. Figure 8 shows the control law u and two state values x_1, x_2 for each of systems $1, 3$, and 5. We see that it stabilizes the system and is only a little rough during the switching phases when the system is uncertain of the current topology. The state values for system 5 are more unstable than those of systems 1 and 3 because its local links vary the most across all four topological configurations. Finally, using standard average consensus, all systems converge to the same transition matrix:

$$\hat{P} = \begin{bmatrix} 0 & 0.3333 & 0.6667 & 0 \\ 0.75 & 0 & 0.25 & 0 \\ 0 & 0.1429 & 0 & 0.8571 \\ 0 & 0.6 & 0.4 & 0 \end{bmatrix}$$

References

[1] Anderson, J., Doyle, J. C., Low, S., and Matni, N. (2019). System level synthesis. *ArXiv preprint, arXiv:1904.01634*.

[2] Bandyopadhyay, S. and Chung, S.-J. (2018). Distributed bayesian filtering using logarithmic opinion pool for dynamic sensor networks. *Automatica*, 97:7–17.

[3] Boyd, S. and Barratt, C. *Linear controller design: limits of performance*. Prentice Hall Information and System Sciences Series. Prentice Hall.
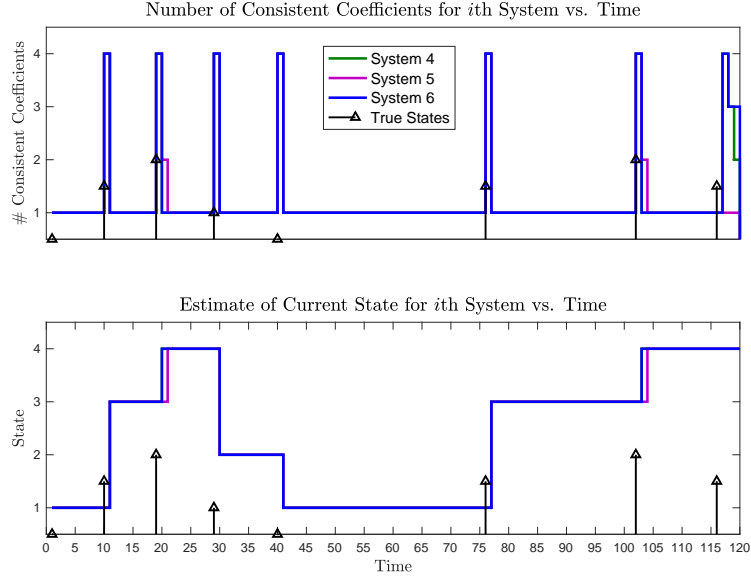
Figure 7: [Top] The number of consistent coefficients and [Bottom] the estimate of the current state for systems 4, 5, and 6.
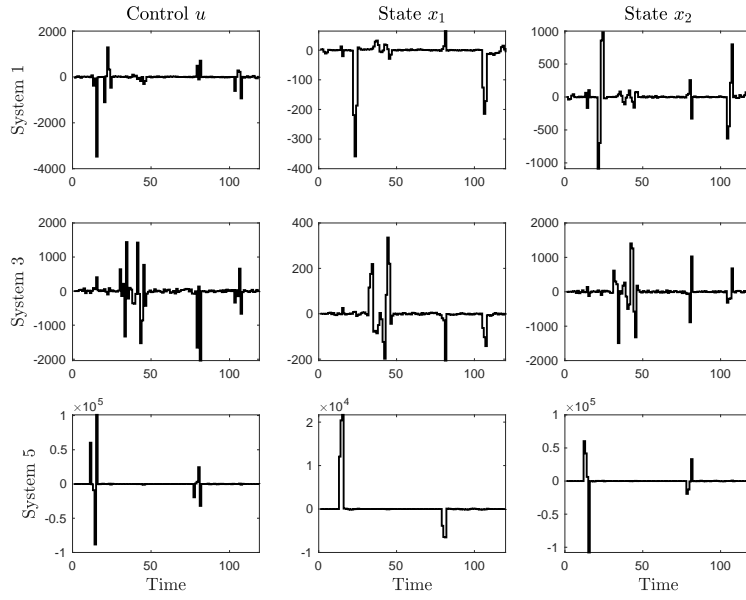

Figure 8: Control law and state evolutions for systems 1, 3, and 5.

[4] Cattivelli, F. S. and Sayed, A. H. (2010). Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084.

[5] Cheng, L., Wang, Y., Hou, Z.-G., Tan, M., and Cao, Z. (2013). Sampled-data based average consensus of second-order integral multi-agent systems: Switching topologies and communication noises. *Automatica*, 49(5):1458 – 1464.

[6] Chung, S.-J. and Slotine, J. E. (2009). Cooperative robot control and concurrent synchronization of Lagrangian systems. *IEEE Transactions on Robotics*, 25(3):686–700.

[7] Chung, S.-J. and Slotine, J.-E. (2010). On synchronization of coupled Hopf-Kuramoto oscillators with phase delays. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3181–3187.

[8] Dahleh, M. and Díaz-Bobillo, I. (1995). *Control of uncertain systems: a linear programming approach*. Prentice Hall.

[9] Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2017). On the sample complexity of the linear quadratic regulator. *ArXiv preprint, arXiv:1710.01688*.

[10] Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2018a). Regret bounds for robust adaptive control of the linear quadratic regulator. *ArXiv preprint, arXiv:1805.09388*.

[11] Dean, S., Tu, S., Matni, N., and Recht, B. (2018b). Safely learning to control the constrained linear quadratic regulator. *ArXiv preprint, arXiv:1809.10121*.

[12] Doyle, J., Glover, K., Khargonekar, P., and Francis, B. (1989). State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847.

[13] Doyle, J. C., Matni, N., Wang, Y., Anderson, J., and Low, S. (2017). System level synthesis: A tutorial. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2856–2867.

[14] Gao, Y. and Wang, L. (2011). Sampled-data based consensus of continuous-time multi-agent systems with time-varying topology. *IEEE Transactions on Automatic Control*, 56(5):1226–1231.

[15] Ge, X. and Han, Q.-L. (2014). Distributed fault detection over sensor networks with Markovian switching topologies. *International Journal of General Systems*, 43(3-4):305–318.

[16] Han, Q.-L., Yang, F., and Ge, J. (2017). Distributed networked control systems: A brief overview. *Information Sciences*, 380(20):117–131.

[17] Ho, D. and Doyle, J. C. (2019). Scalable robust adaptive control from the system level perspective.

[18] Ioannou, P. A. and Sun, J. (1995). *Robust Adaptive Control*. Prentice-Hall, Inc., USA.

[19] Matni, N., Wang, Y., and Anderson, J. (2017). Scalable system level synthesis for virtually localizable systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3473–3480.

[20] Nedic, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J. N. (2009). On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517.

[21] Olfati-Saber, R. (2007). Distributed Kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498.

[22] Olfati-Saber, R. (2009). Kalman-consensus filter : Optimality, stability, and performance. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 7036–7042.

[23] Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533.

[24] Slotine, J.-J., Wang, W., and Rifai, K. (2004). Contraction analysis of synchronization in networks of nonlinearly coupled oscillators. In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, pages 5–8.

[25] Tao, G. (2014). Multivariable adaptive control: A survey. *Automatica*, 50(11):2737 – 2764.

[26] Vidyasagar, M. (2011). *Control System Synthesis: A Factorization Approach*. Morgan & Claypool Publishers, 1st edition.

[27] Wang, Y.-S., Matni, N., and Doyle, J. (2018). Separable and localized system level synthesis for large-scale systems. *IEEE Transactions on Automatic Control*, 63(12):4234–4249.

[28] Wang, Y.-S., Matni, N., and Doyle, J. C. (2014). Localized LQR optimal control.

[29] Wang, Y.-S., Matni, N., and Doyle, J. C. (2016). A system level approach to controller synthesis.

[30] Wang, Y.-S., You, S., and Matni, N. (2015). Localized distributed Kalman filters for large-scale systems. volume 48.

[31] Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 63–70.

[32] Youla, D., Jabr, H., and Bongiorno, J. (1976). Modern Wiener-Hopf design of optimal controllers–Part II: The multivariable case. *IEEE Transactions on Automatic Control*, 21(3):319–338.