Localized Learning of Robust Controllers for Networked Systems with Unknown Parameters and Dynamic Topology

SooJean Han*

April 28, 2020

Abstract

This manuscript addresses the problem of controller synthesis for large-scale networked systems with dynamic and unknown topology. Such networked systems arise in applications such as the power grid when lines between consumers are downed due to natural causes, or in multi-agent robotic networks and embedded sensor networks, both of which tend to be structurally complex and prone to high faults. We develop a robust and adaptive system-level controller synthesis approach to this problem by making two extensions to an existing system-level controller synthesis approach: 1) each power system estimates uncertainties for its own local region independently of other systems in the network, allowing for a distributed implementation that can be mostly parallelized, and 2) we allow for uncertainties in the topological structure of the network to account for dynamically-changing topologies; in the case study of the power grid, this may correspond to new cable installations or downed power lines. The proposed topological algorithm will generate a control law that is robust to all possible link failures, while adaptively deducing which links were not from the original topology by iteratively solving an L_1-regularization-based convex optimization problem. It is described sequentially in three settings: centralized, localized, and iterative localized for multiple modifications over time. In particular, the iterative localized scheme is designed around networks which switch between topological configurations arranged in a finite-state Markov Chain; the subsystems additionally perform consensus to estimate its transition probabilities. The advantages of this method are twofold. First, as a result of the localized implementation of the system-level approach, our controller can be extended to large-scale networks. Second, achieving exact consensus or learning precisely where the network structure has changed is not necessary to stabilize the overall network due to the robust nature of our controller. We simulate our algorithms on various network topologies to illustrate.

1 Introduction

Maintaining stability in a large-scale network, even when the parameters or topology of a network are not totally known, is an important issue which arises in a variety of applications. In the specific

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

case of the power grid, one may think of lines which have fallen due to severe weather conditions or the installation of new lines between consumers. Furthermore, power fluctuations should be monitored and controlled in order to regulate flows and reduce energy waste. More importantly, for any possible abnormal yet massive disruptions in the network, the sensors and controllers should mitigate the problem as quickly and as close to source as possible so that the failure does not cascade throughout the global network, as it did in the Northeast blackout of 2003 [1]. Alternatively stated, the effects of any disturbance should be *localized*. Other applications where some of the same issues arise are in multi-agent robot systems which work towards a collaborative goal (e.g., formation flying, information gathering); because each agent functions independently of each other, they need to communicate wirelessly to achieve this goal, corresponding to a highly dynamic communication network topology. There is a subsequent need to upgrade the control scheme in an efficient way that does not involve redesigning from scratch simply because the topology has changed.

One additional obstacle that is faced with in the deployment of control algorithms to a distributed, decentralized architecture is that the plant network is usually of immense scale. Realistic constraints such as system-to-system communication delay cannot be ignored. This motivates a distributed, decentralized implementation of a network of multiple sensors and controllers, and the recently-developed *System Level Synthesis* (SLS) framework [11, 25] facilitates our design. SLS leverages a Youla parameterization-like design to construct controllers that ensure these constraints are obeyed, along with additional desired design constraints. Imposing spatial and temporal locality also makes the control scheme scalable to large networks like the grid and disturbances are rejected within a local region about the point of origin. The work of [15] extended the SLS approach to integrate adaptive learning of bounded parameter uncertainties in the system dynamics while alternately designing controllers that are robust to these uncertainties.

In this paper, we develop robust and adaptive system-level approaches to address the mentioned disturbance-rejection problem in a decentralized way subject to various uncertainties (particularly parametric and topological) that may arise as a result of natural disruptions to the grid. Leveraging the localized implementation of SLS, we first introduce a method of local adaptation, where each system of the network maintains estimates of a local subset of uncertainties from the whole network. As each system's set is updated independently of the other systems, with minimal required communication with other systems, this allows for a parallel implementation of the adaptation process, which couples well with the decentralized control law deployment enabled by the SLS approach. We then show how this method can be extended to not only handle parametric uncertainties, but also uncertainties in the topological graph of the network, and take the first step towards extending the system-level synthesis (SLS) approach to designing localized robust controllers for potentially large-scale networked systems which adapts to topological changes over time. We first describe the core mechanics of the algorithm in the centralized formulation. Then the progression to the localized formulation follows easily, as its main distinction from the centralized formulation is that each subsystem in the network only accounts for structural changes in a local region of neighboring subsystems. Finally, we manipulate the localized formulation into an iterative scheme for fault-tolerance against successive modifications between links in the topological structure of the network. In particular, we consider the control of networks which switch between configurations according to a finite-state Markov Chain. An additional averaging consensus algorithm is performed atop the scheme to learn the transition probabilities of the chain, which can then be used to predict the next state and help speed up the adaptive learning process.

Due to the robust nature of the proposed controller, we emphasize that learning the probabilities or the locations of topological change are priorities which are second to system stabilization. Adaptive control problems come with a fundamental trade-off between safely stabilizing the system and adaptively learning the uncertainties; simultaneous exact achievement of both tasks is nearly impossible, even for systems which are not large-scale. The property of robustness allows us to optimize this trade-off, and uncertainty sets will continue to be reduced for as long as there does not exist a controller which will stabilize the system. This is especially useful for large-scale networks, where maintaining a complete picture of the entire system is nearly impossible due to a large number of parameters and a dense connection of subsystems.

1.1 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 [27] introduced the Youla parameterization approach, and [22] 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 [7]. 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 [10]. 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 [24, 25, 11, 17]. 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 [2] 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.

Adaptive control and dynamic network structures: Traditional adaptive control approaches like [16, 21] tend to overlook the issue of maintaining stability while learning uncertainties. More promising developments such as [8, 9] combine elements from data-driven control and machine learning to manage safety, but have not been shown to be scalable. Furthermore, most adaptive techniques focus on handling parametric uncertainties rather than topological ones; a method of designing scalable robust adaptive controllers for parametric uncertainties was proposed in [15]. On the other hand, there exists a rich literature of work on the treatment of distributed networked control systems with dynamic structures; see [14], Table 2 for a comprehensive survey. [13] 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 [12, 4, 18] and criterion for convergence are also provided. Such approaches have a lot of practical applicability like cooperative robot control [5, 6], oscillator synchronization [20], as well as parameter estimation and sensor fusion [26, 19]. 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 robustness property mentioned before. 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 subsystems 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 $\mathcal{V} = \{1, 2, \cdots N_s\}$.

3 A Brief Review of the Systems Level Approach

Before we present the main schemes, we will briefly review the necessary concepts from system level synthesis (SLS). SLS is appealing because it provides important advantages such as the incorporation of controller design specifications as convex constraints, and scalability to extremely large networks. For more details, we refer the interested reader to [25, 2].

For plants of the following generic linear, discrete-time form $\mathbf{x}[t+1] = A\mathbf{x}[t] + B\mathbf{u}[t] + \mathbf{w}[t]$, and assuming the disturbance is bounded $\|\mathbf{w}\|_\infty < \eta$ for some $\eta > 0$, the state-feedback controller is implemented as follows:

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

$$\hat{\mathbf{w}}[t] = \mathbf{x}[t] - \hat{\mathbf{x}}[t] \tag{1b}$$

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

with the controller's internal state $\hat{\mathbf{w}}$ and *system responses* $\{\Phi_x, \Phi_u\}$, which are closed-loop transfer function maps defined as $\mathbf{x} = \Phi_x\mathbf{w}$ and $\mathbf{u} = \Phi_u\mathbf{w}$. These transfer function maps are constrained to finite time horizon T, for which we will denote $\{\Phi_x, \Phi_u\} \in \mathcal{F}_T$. It was shown in [17] that even when this relationship is approximately satisfied, the implementation (1) produces a stable closed-loop response.

Definition 1. We associate a local d-hop set $\mathcal{L}_d(i)$ with each system $i \in \mathcal{V}$ to be the set of systems j for which the (i, j)th entry of G^d is nonzero. The system response is said to be d-localizable iff for every $i \in \mathcal{V}, j \notin \mathcal{L}_d(i)$, we have $\Phi_{x,ij} = 0$, and analogously for Φ_u. We denote this as $\{\Phi_x, \Phi_u\} \in \mathcal{L}_d$.

The desired behavior can then be achieved by constraining $\{\Phi_x, \Phi_u\}$ to lie in an appropriate convex set $\mathcal{S}$, and solving an optimization problem of the form:

$$\min_{\{\Phi_x, \Phi_u\}} f(\Phi_x, \Phi_u, Q, R) \tag{2a}$$

$$\text{s.t. } \{\Phi_x, \Phi_u\} \in \mathcal{S} \tag{2b}$$

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. We typically have $\mathcal{S} \subseteq \mathcal{L}_d \cap \mathcal{F}_T$, and it also includes system-to-system communication delay constraints as well as the necessary robustness constraints to keep the closed-loop response stable during the process of learning the uncertainties.

4 Robust Parameter Adaptation

4.1 Centralized Implementation

We will restrict our attention to the case of fully-known B, and assume any uncertain parameter values are in A. Let ξ_l be the uncertain parameters in A, where $l = 1, \cdots, p$. We are given initial bounds for each ξ_l, from which we can construct the initial polytope of uncertainty as $\mathcal{P}_0$. Create $N_x \times N_x$ basis matrices $\mathcal{A}_k$ corresponding to each parameter, plus a possible additional basis matrix $\mathcal{A}_0$ for the known part. A can then be decomposed

$$A = \mathcal{A}_0 + \sum_{l=1}^{p} \mathcal{A}_l \xi_l$$

By the system dynamics, $\boldsymbol{\xi} := \begin{bmatrix}\xi_1 \cdots \xi_p\end{bmatrix}^T$ must obey

$$\left\| \mathbf{x}[t+1] - \left(\mathcal{A}_0 + \sum_{l=1}^{p} \mathcal{A}_l \xi_l \right) \mathbf{x}[t] - B\mathbf{u}[t] \right\| \leq \eta$$

where we assume bounded disturbance $\|\mathbf{w}\|_\infty \leq \eta$. The set of all $\boldsymbol{\xi}$ that obeys this constraint forms our constraint set $\mathcal{C}_{t+1}$, and we update the uncertainty set as $\mathcal{P}_{t+1} = \mathcal{P}_t \cap \mathcal{C}_{t+1}$. The hope is that these polytopes shrink with each timestep as we accumulate more observations $(\mathbf{x}[t+1], \mathbf{x}[t], \mathbf{u}[t])$ from the system's behavior.

For controller design, start with the implementation (1). Because we are updating our estimates of the values at each time step as we adapt, the system responses $\Phi_x^{(t)}, \Phi_u^{(t)}$ are denoted with superscript (t) to make explicit that they are time-varying. Define

$$\Delta_t(A, B, \Phi_x, \Phi_u) = \Phi_x[t+1] - A\Phi_x[t] - B\Phi_u[t] \; \forall t < T$$
$$\Delta_T(A, B, \Phi_x, \Phi_u) = -A\Phi_x[T] - B\Phi_u[T]$$

Then from (1b):

$$
\begin{aligned}
\hat{\mathbf{w}}[t] &= \mathbf{x}[t] - \sum_{k=2}^{T} \Phi_x^{(t)}[k]\hat{\mathbf{w}}[t+1-k] \\
&= (A\mathbf{x}[t-1] + Bu[t-1] + \mathbf{w}[t-1]) - \sum_{k=2}^{T} \Phi_x^{(t)}[k]\hat{\mathbf{w}}[t+1-k] \\
&= A\left(\sum_{k=1}^{T} \Phi_x^{(t-1)}[k]\hat{\mathbf{w}}[t-k]\right) + B\left(\sum_{k=1}^{T} \Phi_u^{(t-1)}[k]\hat{\mathbf{w}}[t-k]\right) + \mathbf{w}[t-1] - \sum_{k=2}^{T} \Phi_x^{(t)}[k]\hat{\mathbf{w}}[t+1-k] \\
&= -\sum_{k=1}^{T} \Delta_k(A, B, \Phi_x^{(t-1)}, \Phi_u^{(t-1)})\hat{\mathbf{w}}[t-k] + \sum_{k=1}^{T-1} \left(\Phi_x^{(t-1)} - \Phi_x^{(t)}\right)[k+1]\hat{\mathbf{w}}[t-k] + \mathbf{w}[t-1]
\end{aligned}
$$

where the second equality comes from substituting in the expression for $\mathbf{u}[t]$ and using the fact that $\Phi_x^{(t)}[1] = I$ for all t. By the triangle inequality and the submultiplicativity property of norms:

$$
\|\hat{\mathbf{w}}[t]\| \leq \sum_{k=1}^{T} \left\|\Delta_k(A, B, \Phi_x^{(t-1)}, \Phi_u^{(t-1)})\right\| \|\hat{\mathbf{w}}[t-k]\| + \left\|\sum_{k=1}^{T} \left(\Phi_x^{(t-1)} - \Phi_x^{(t)}\right)[k+1]\hat{\mathbf{w}}[t-k]\right\| + \eta
$$

Thus, we can constrain

$$
\sum_{k=1}^{T} \left\|\Delta_k(A', B, \Phi_x^{(t-1)}, \Phi_u^{(t-1)})\right\| \leq \lambda_t \tag{3a}
$$

$$
\left\|\sum_{k=1}^{T} \left(\Phi_x^{(t-1)} - \Phi_x^{(t)}\right)[k+1]\hat{\mathbf{w}}[t-k]\right\| \leq \gamma \tag{3b}
$$

for every possible matrix $A' = \mathcal{A}_0 + \sum \mathcal{A}_l \xi_l$ constructed by choosing uncertain parameters $\boldsymbol{\xi} \in \mathcal{P}_{t+1}$. To ensure that it holds for all $\boldsymbol{\xi} \in \mathcal{P}_{t+1}$, it suffices to ensure that is holds just for the extreme points of $\mathcal{P}_{t+1}$. These constraints are substituted as $\mathcal{S}$ into the optimization problem (2).

We call λ_t the *robustness margin* and for each timestep t it determines whether the controller is stabilizable with the tth polytope of uncertainties. We initially solve (2) with $f(\Phi_x, \Phi_u, Q, R, \lambda_t) = \lambda_t$, and if $\lambda_t \leq \lambda^*$ for a chosen value λ^*, we resolve (2) with

$$
f(\Phi_x, \Phi_u, Q, R, \lambda_t) = \sum_{k=1}^{T} \|Q\Phi_x[k] + R\Phi_u[k]\|_1
$$

Two steps are taken because optimizing for a performance objective is only reasonable if robust stability is feasible with uncertainty $\mathcal{P}_t$.

γ is the *adaptation margin* and ensures that the system response Φ_x doesn't fluctuate wildly with varying $\mathbf{w}$. One can observe that if $\mathbf{w}$ is very large, $\Phi_x^{(t)}$ is forced to be close to the previous iteration's $\Phi_x^{(t-1)}$ in order to compensate and obey the constraint. However, if γ is chosen too small, it may learn the incorrect system response and subsequently remain close to this response for all future time, resulting in an unstable controller. For this reason, the addition of this constraint is typically made optional.

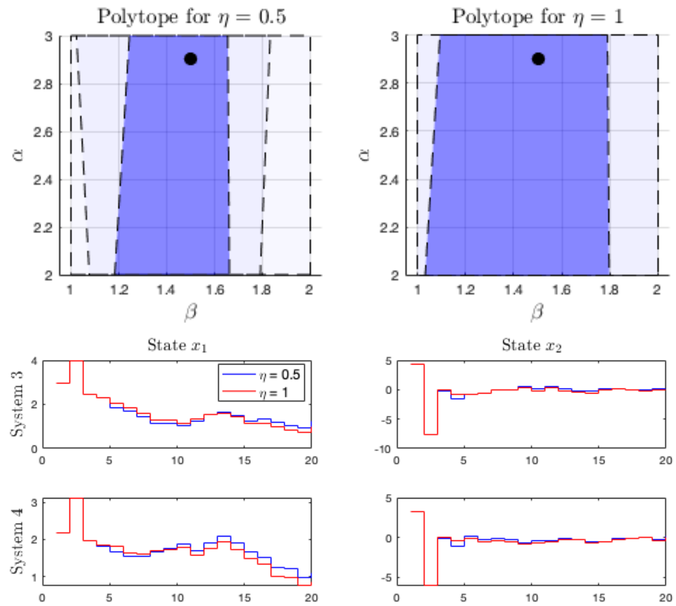

Figure 1: In the top figure, the resulting polytope. The black dot indicates the true parameter value. Multiple shades are overlapped so that the shape with the darkest color represents the final polytope. On the bottom figure, the state trajectories for systems 3 and 4 are shown. The control law does a better job stabilizing the system when true parameter is learned less accurately.

As illustration, we consider an adaptive implementation of a rearranged chain network of 4 systems with adjacency matrix:

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

The dynamics for the ith system are expressed with two unknown parameters α and β as

$$\mathbf{x}^{(i)}[k+1] = \begin{bmatrix} 1 & \Delta t \\ -\sum_j \alpha \Delta t & 1 - \beta_i \Delta t \end{bmatrix} \mathbf{x}^{(i)}[k] + \sum_{j \in \mathcal{N}_i} \begin{bmatrix} 0 & 0 \\ \alpha \Delta t & 0 \end{bmatrix} \mathbf{x}^{(j)}[k] + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}^{(i)}[k] + \mathbf{w}^{(i)}[k]$$

The true parameter values are chosen $\alpha = 2.9, \beta = 1.5$ to make the system open-loop unstable $(\max(|\lambda(A)|) = 1.0492)$, and the initial uncertainty ranges are $\alpha \in [2, 3], \beta \in [1, 2]$.

In Figure 1, we simulate this system for 20 time steps and two different values of η, both cases driven by the same noise process $\mathbf{w}$ such that $\|\mathbf{w}[t]\|_\infty \leq 0.3$. Initial conditions are made small: for each system, $\mathbf{x}_1^{(i)}[0] \in \text{Unif}[0, 2]$ and $\mathbf{x}_2^{(i)}[0] \in \text{Unif}[3, 5]$. As expected, the algorithm for $\eta = 1$ adapts less towards the true parameters than when $\eta = 0.5$. Furthermore, the control law for $\eta = 0.5$ does not stabilize the states of the systems to zero as closely as the control law for $\eta = 1$ does, which is illustrative of the fact that the system adapts better with perturbations that make yield nonzero values even at the expense of driving the system unstable. Finally, because the system is robustly stabilizable for both polytopes, learning precisely the true parameter value is not necessary to drive the states stably to 0.

4.2 Decentralized and Localized Implementation

We now extend the centralized implementation to a localized version that employs a decomposition of the full problem into multiple independent pieces, to achieve computational scalability to systems of larger size. Define local set $\mathcal{L}(i)$ for each system $i = 1, \cdots, N_s$ to be the d-hop set of i. Further, let d_c be the communication delay matrix between systems of the network, with the (i, j)th entry defined as

$$d_c(i,j) = \begin{cases} |j-i| & \text{if } j \in \mathcal{L}(i) \\ \infty & \text{else} \end{cases}$$

Let $\xi_l^{(i)}$, $l = 1, \cdots, p_i$ be the uncertain parameters associated with the ith system. Create the local set of basis matrices $\mathscr{A}_i := \left\{\mathcal{A}_l^{(i)}\right\}, l = 0, \cdots, p_i$ for each i in a fashion similar to the centralized case. Rather than maintaining estimates for all parameters in the entire network, each system i only keeps track of parameters $\left\{\boldsymbol{\xi}^{(j)} : j \in \mathcal{L}(i)\right\}$ for systems in $\mathcal{L}(i)$. Moreover, it updates its corresponding A submatrix A_i using the collection of basis matrices $\{\mathscr{A}_j : j \in \mathcal{L}(i)\}$. For example, for a chain or ring topology where each system has two consecutive neighbors, we have

$$A_i = \left[\begin{array}{c} \sum_{l=1}^{p_{i-1}} \xi_l^{(i-1)} \mathcal{A}_l^{(i-1)} \\ \hline \sum_{l=1}^{p_i} \xi_l^{(i)} \mathcal{A}_l^{(i)} \\ \hline \sum_{l=1}^{p_{i+1}} \xi_l^{(i+1)} \mathcal{A}_l^{(i+1)} \end{array}\right] \in \mathbb{R}^{n(2d+1)\times N_x}$$

Each system i has an initial uncertain polytope $\mathcal{P}_0^{(i)}$, which is updated iteratively using local measurements $\mathcal{P}_t^{(i)} = \mathcal{P}_{t-1}^{(i)} \cap \mathcal{C}_t^{(i)}$, where $\mathcal{C}_t^{(i)}$ is the consistent set constructed from local state observations similarly to the centralized algorithm.

Now we solve a local optimization problem of the form (2) for the ith columns of the system response matrices $\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}$. The constraints follow analogously to (3). Define the submatrix

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

where $i, j = 1, \cdots, N_s$ regardless of whether they are neighbors or not, $k = 1, \cdots, T$, and t iterates over the simulation time. This allows us to define our robustness margin constraints

$$\left\|\sum_{j\in\mathcal{L}(i)} \Delta_k^j\left(A, B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right)\right\| \leq c_i\rho^{k-1}\ \forall k \leq T-1$$

$$c_i \sum_{k=1}^{T} \rho^{k-1} \leq \lambda_t^{(i)} + \epsilon$$

and in place 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 slack variable ϵ helps us reduce the two-step process (checking for robust stabilizability prior to determining a legitimate control law) of solving (2) into a single step optimization problem. The objective function to minimize becomes

$$f(\Phi_x, \Phi_u, Q, R) = \sum_{k=1}^{T} \left\| Q\Phi_x[k] + R\Phi_u[k] \right\|_1 + r_c \epsilon$$

where $r_c > 0$ is a *relaxation constant* hyperparameter and we impose the additional constraint $\epsilon \geq 0$ within $\mathcal{S}$ in (2). The reasoning is that if the solution causes ϵ to be very large, this implies that the system is unable to determine a control law that is robustly stabilizable to all uncertainties in the current $\mathcal{P}_t$. We choose a very large relaxation constant so that even the slightest deviation of ϵ from 0 causes the cost objective to become large. When this happens, the optimization problem focuses on decreasing the uncertainty in the polytope. On the other hand, if the cost is small, it indicates that there exists a solution Φ_x, Φ_u that is robustly stabilizable with respect to $\mathcal{P}_t$.

Additionally, the method used in [15] ensured that these constraints were satisfied for all extreme points of $\mathcal{P}_t$, just as in the centralized version. However the number of extreme points scaled immensely , leading to much burden in the computation. To bypass this, we employ an alternative strategy using smart random selection of a point in the polytope. Note that the ith polytope of parameters can be described by a set of equations $M_i \boldsymbol{\xi}^{(i)} \leq q_i$ for parameter vector of dimension p_i. We sample a Gaussian random vector $\mathbf{c}_i \in \mathbb{R}^{p_i}$, normalized to be uniform, and solve the following minimization problem to obtain our selection $\boldsymbol{\xi}_i$ in the direction of the sampled random vector

$$\begin{aligned} &\min_{\boldsymbol{\xi}^{(i)}} \mathbf{c}_i^T \boldsymbol{\xi}^{(i)} \\ &\text{s.t. } M_i \boldsymbol{\xi}^{(i)} \leq q_i \end{aligned}$$

Note that because we are minimizing over an affine subspace, this inevitably leads to one of the extreme points to be chosen. Once the selection is made, we construct a local A_i submatrix and solve the local optimal control problem (2) with constraints and objective function set as above.

Because of the decomposition of the full problem into independent subproblems solved separately by each system, most of the algorithm outlined above can be implemented in a parallel manner. The final step of the algorithm generates the control law $\mathbf{u}[t]$ to be input into the system dynamics at timestep $t+1$. To achieve this, each system i aggregates all system responses from its local set, taking into account the communication delay, and computes a new local control law $\mathbf{u}_i[t]$ and its own estimated disturbance $\hat{\mathbf{w}}^{(i)}$ according to an implementation like (1):

$$\hat{\mathbf{w}}^{(i)}[t] = \mathbf{x}^{(i)}[t] - \sum_{j \in \mathcal{L}(i)} \sum_{k=2}^{T} \Phi_{x,ij}^{(t-d_c(i,j))}[k] \hat{\mathbf{w}}^{(j)}[t-k+1]$$

$$\mathbf{u}^{(i)}[t] = \sum_{j \in \mathcal{L}(i)} \sum_{k=1}^{T} \Phi_{u,ij}^{(t-d_c(i,j))}[k] \hat{\mathbf{w}}^{(j)}[t-k+1]$$

In Figure 2 , we simulate a ring network with number of systems $N_s = 8$. We have the same uncertain parameter arrangement as in the previous single-system case, but the β_i are assumed different for each system i. $\mathcal{L}(i)$ is chosen to be the 1-hop region around system i so that each

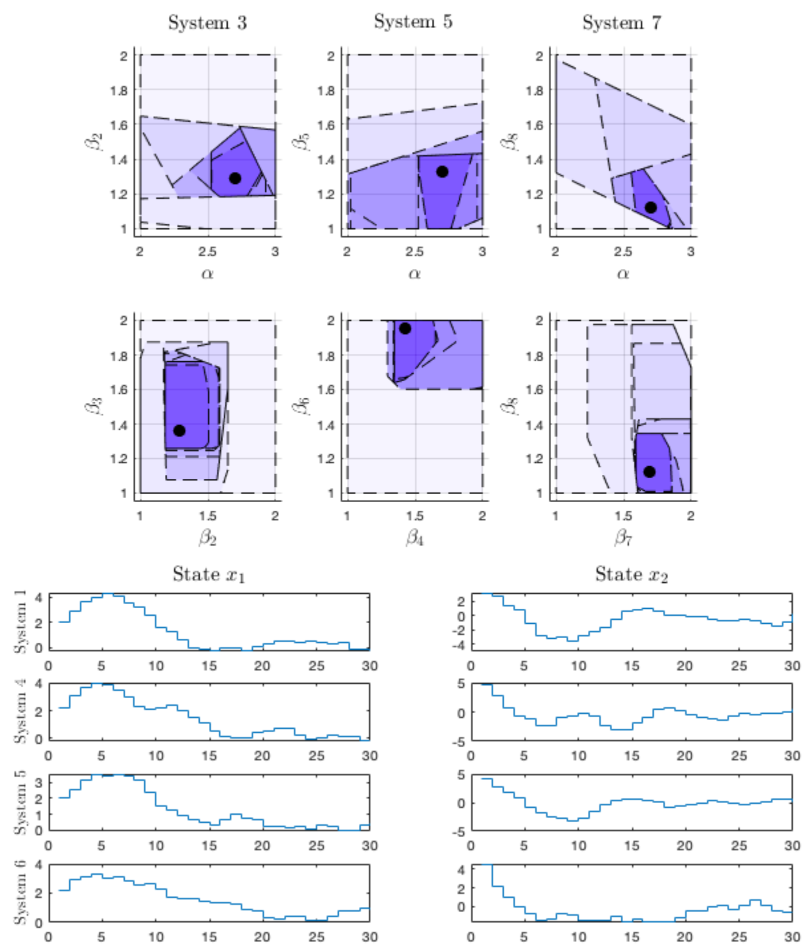

Figure 2: In the first figure, pairwise polytopes for systems 2 and 7 from the ring topology of 8. In the second figure, state trajectories for systems 1,4,5, and 6 in the ring network.

system i only maintains estimates of its own parameters and the parameters of its immediate neighbors $\alpha, \beta_{i-1}, \beta_i, \beta_{i+1}$. Each system response is generated with a finite time horizon of $T = 8$, and we choose hyperparameters $\rho = 0.7, r_c = 1000$. Simulating for 30 timesteps, we see all the states stabilize relatively well around 0 and any oscillations eventually become confined within a bounded interval. There is not much significant degradation in performance from the centralized algorithm.

5 Robust Topology Adaptation

The previous section focused on methods to stably learn parameter uncertainties. Here, we develop similar approaches to handle uncertainties in the topology of the network, and propose a control law design 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. The proposed centralized and localized algorithms are similar to the parameter adaptation algorithm discussed in the previous section. In the case of the smart 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.

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

$$\mathcal{P}_0 := \left\{ \sum_{l=1}^{M} \xi_l \mathcal{A}_l \ : \ \xi_l \in \{-1, 0, 1\} \right\} \tag{4}$$

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 $(\mathbf{x}[t+1], \mathbf{x}[t], \mathbf{u}[t])$:

$$\mathcal{P}_{t+1} := \left\{ D \in \mathcal{P}_t \ : \ \left\| \mathbf{x}[t+1] - \left(A^* + \sum_{l=1}^{M} \xi_l \mathcal{A}_l \right) \mathbf{x}[t] - B\mathbf{u}[t] \right\|_\infty \leq \eta \right\} \tag{5}$$

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:

$$\sum_{k=1}^{T} \left\| \Delta_k(A', B, \Phi_x^{(t-1)}, \Phi_u^{(t-1)}) \right\| \leq \lambda_t \ \ \forall A' = A^* + D, \ D \in \mathcal{P}_t \tag{6a}$$

$$\left\| \sum_{k=1}^{T} \left(\Phi_x^{(t-1)} - \Phi_x^{(t)} \right) [k+1] \hat{\mathbf{w}}[t-k] \right\| \leq \gamma \tag{6b}$$

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

$$\begin{aligned} \min_{\left\{ \Phi_x^{(t)}[k], \Phi_u^{(t)}[k] \right\}_{k=1}^{T}, \lambda_t} \quad & f\left(\Phi_x^{(t)}, \Phi_u^{(t)}, Q, R \right) = \begin{cases} \lambda_t & \text{if } \lambda_t \leq \lambda^* \\ \sum_{k=1}^{T} \left\| Q\Phi_x^{(t)}[k] + R\Phi_u^{(t)}[k] \right\|_1 & \text{else} \end{cases} \\ \text{s.t. } & \{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{F}_T \text{ and } (6) \end{aligned} \tag{7}$$

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 (6b) 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 (7) 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 (4) 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 (5).

To design local controllers, we solve a local optimization problem of the form (2) for the ith columns of the system response matrices $\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}$. The constraints follow analogously to (6). Define the submatrix

$$\Delta_k^j\left(A, B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right) := \Phi_{x,ji}^{(t)}[k+1] - \sum_{l\in\mathcal{N}(j)} A_{jl}\Phi_{x,li}^{(t)}[k] - B_j\Phi_{u,ji}^{(t)}[k] \tag{8}$$

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

$$\left\|\sum_{j\in\mathcal{L}_d(i)} \Delta_k^j\left(A', B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right)\right\| \leq c_i\rho^{k-1} \quad \forall A' = A^* + D, D \in \mathcal{P}_t \quad \forall k \leq T-1 \tag{9a}$$

$$c_i\sum_{k=1}^{T}\rho^{k-1} \leq \lambda_t^{(i)} + \epsilon,\ \epsilon \geq 0 \tag{9b}$$

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:

$$\min_{\left\{\Phi_x^{(t)}[k], \Phi_u^{(t)}[k]\right\}_{k=1}^{T}, \lambda_t, \epsilon} f\left(\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}, Q_i, R_i\right) = \sum_{k=1}^{T}\left\|Q\Phi_{x,i}^{(t)}[k] + R\Phi_{u,i}^{(t)}[k]\right\|_1 + r_c\epsilon \tag{10}$$

$$\text{s.t. } \{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{L}_d \cap \mathcal{F}_T \text{ and } (9)$$

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:

$$\mathbf{x}_i[t+1] = A_{ii}(\alpha(t))\mathbf{x}_i[t] + \sum_{j\in\mathcal{N}_i(\alpha(t))} A_{ij}\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t] \quad \forall i \in \mathcal{V} \tag{11}$$

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 [19, 26], 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 [26]. Since the methodology is the same across all subsystems $i \in \mathcal{V}$, the subscript i is henceforth removed for notational simplicity.

Similar to (4), 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 (11). 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 (5) 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 (10) 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 [23], 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 3. 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.

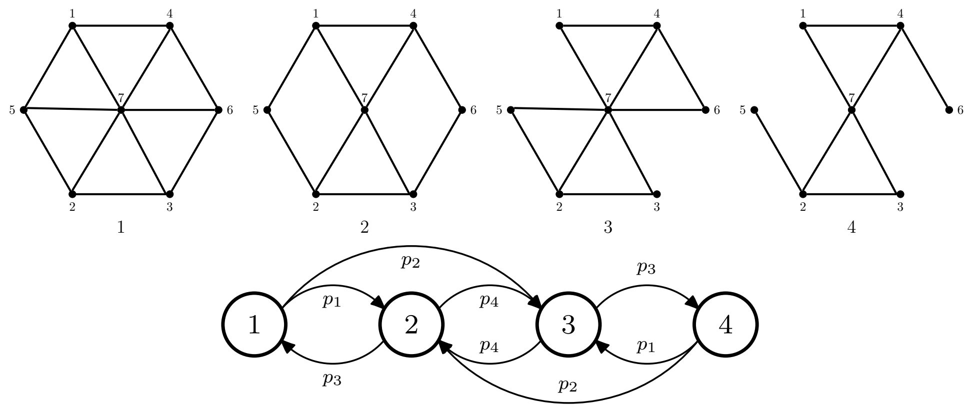

Figure 3: [Top] Hexagon topologies arranged in a Markov chain with four states. [Bottom] The state transition diagram for the four states.

In Figure 4, 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 5 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:

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

Acknowledgements

The author would like to thank Dimitar Ho and John Doyle for insightful discussions and input.

References

[1] The 2003 northeast blackout : Five years later. `https://www.scientificamerican.com/article/2003-blackout-five-years-later/`. Accessed: 2019-09-24.

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

[3] Boyd, S. and Barratt, C. *Linear controller design: limits of performance*. Prentice Hall Information and System Sciences Series. Prentice Hall.

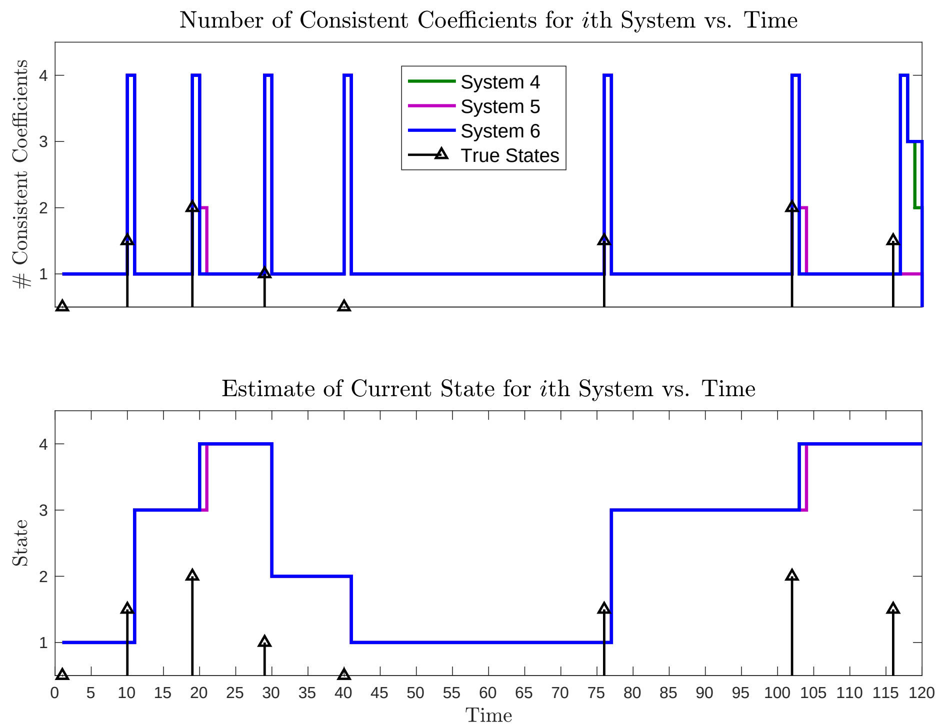

Figure 4: [Top] The number of consistent coefficients and [Bottom] the estimate of the current state for systems $4, 5$, and 6.

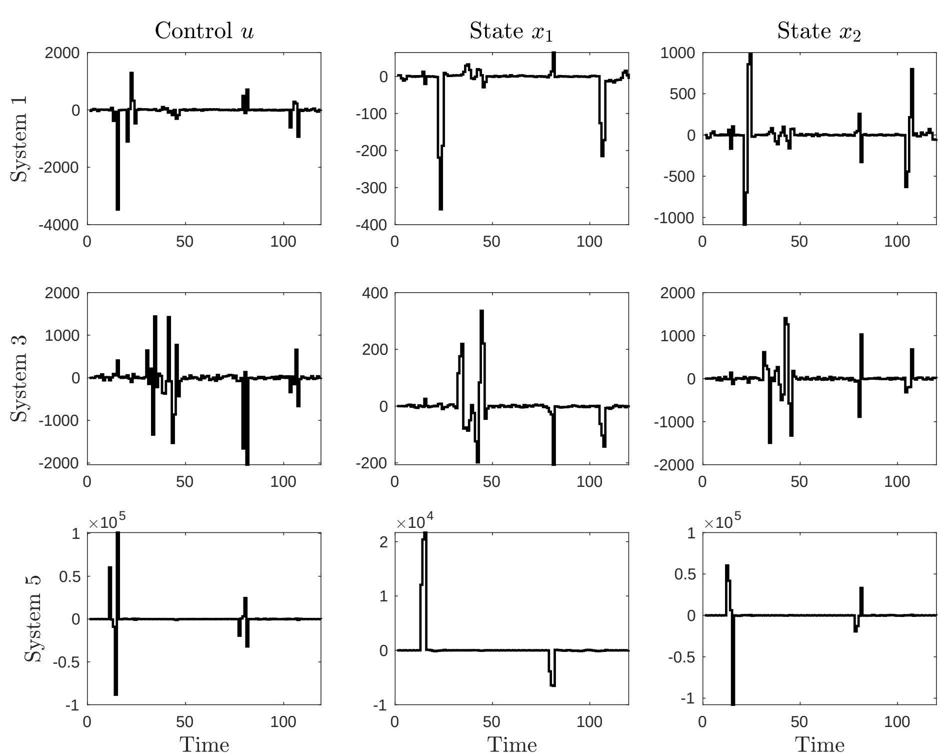

Figure 5: Control law and state evolutions for systems $1, 3$, and 5.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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*.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

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

[23] 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.

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

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

[26] 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.

[27] 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.