Proceedings of Machine Learning Research vol 120:1–10, 2020

Localized Learning of Robust Controllers for Networked Systems with Dynamic Topology

SooJean Han SOOJEAN@CALTECH.EDU
California Institute of Technology, Pasadena, CA 91125, USA.

Abstract

This paper addresses the problem of controller synthesis for 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. The algorithm 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 our 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. To demonstrate performance, we simulate the iterative localized algorithm for a simple centered hexagon network with 4 different topological states.

Keywords: Networked control systems, Robust control, Switched systems

1. Introduction

Maintaining stability in a large-scale network, even when the topology of the network is not totally known, is an important issue which arises in a variety of applications. For instance, in the 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. Multi-agent robots which operate independently of each other to work towards a collaborative goal (e.g., formation flying, information gathering) communicate wirelessly to do so, 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.

In this paper, we 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.

However, 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.

Related Work Traditional adaptive control approaches like Ioannou and Sun (1995); Tao (2014) tend to overlook the issue of maintaining stability while learning uncertainties. More promising developments such as Dean et al. (2017, 2018) 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 Ho and Doyle (2019). On the other hand, there exists a rich literature of work on the treatment of distributed networked control systems with dynamic structures; see Han et al. (2017), Table 2 for a comprehensive survey. Ge and Han (2014) 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 Gao and Wang (2011); Cheng et al. (2013); Nedic et al. (2009) and criterion for convergence are also provided. Such approaches have a lot of practical applicability like cooperative robot control Chung and Slotine (2009, 2010), oscillator synchronization Slotine et al. (2004), as well as parameter estimation and sensor fusion Xiao et al. (2005); Olfati-Saber and Murray (2004). 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. Background

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 $\mathcal{V} = \{1, 2, \cdots N_s\}$.

2.1. System Level Synthesis Review

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 Wang et al. (2016); Anderson et al. (2019).

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:

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

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 Matni et al. (2017) 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:

$$\min_{\{\Phi_x, \Phi_u\}} f(\Phi_x, \Phi_u, Q, R) \text{ s.t. } \{\Phi_x, \Phi_u\} \in \mathcal{S} \tag{2}$$

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.

3. Implementation

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.

3.1. 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*.

$$\mathcal{P}_0 := \left\{ \sum_{l=1}^{M} \xi_l \mathcal{A}_l \ : \ \xi_l \in \{-1, 0, 1\} \right\} \tag{3}$$

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

$$\mathcal{P}_{t+1} := \left\{ D \in \mathcal{P}_t \ : \ \left\| \mathbf{x}[t+1] - \left(A^* + \sum_{l=1}^{M} \xi_l \mathcal{A}_l \right) \mathbf{x}[t] - B\mathbf{u}[t] \right\|_\infty \leq \eta \right\} \tag{4}$$

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:

$$\sum_{k=1}^{T} \left\| \Delta_k(A', B, \Phi_x^{(t-1)}, \Phi_u^{(t-1)}) \right\| \leq \lambda_t \ \ \forall A' = A^* + D, \ D \in \mathcal{P}_t \tag{5a}$$

$$\left\| \sum_{k=1}^{T} \left(\Phi_x^{(t-1)} - \Phi_x^{(t)} \right) [k+1] \hat{\mathbf{w}}[t-k] \right\| \leq \gamma \tag{5b}$$

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:

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

$$\text{s.t. } \{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{F}_T \text{ and } (5)$$

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 (5b) 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.*

3.2. Localized Implementation

A localized version of the algorithm essentially decomposes (6) 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 (3) 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 (4).

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 (5). Define the submatrix

$$\Delta_k^j\left(A, B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right) := \Phi_{x,ji}^{(t)}[k+1] - \sum_{l \in \mathcal{N}(j)} A_{jl}\Phi_{x,li}^{(t)}[k] - B_j\Phi_{u,ji}^{(t)}[k] \tag{7}$$

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

$$\left\| \sum_{j \in \mathcal{L}_d(i)} \Delta_k^j\left(A', B, \Phi_{x,ji}^{(t)}, \Phi_{u,ji}^{(t)}\right) \right\| \leq c_i \rho^{k-1} \;\; \forall A' = A^* + D, D \in \mathcal{P}_t \;\; \forall k \leq T-1 \tag{8a}$$

$$c_i \sum_{k=1}^{T} \rho^{k-1} \leq \lambda_t^{(i)} + \epsilon, \; \epsilon \geq 0 \tag{8b}$$

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:

$$\min_{\left\{\Phi_x^{(t)}[k], \Phi_u^{(t)}[k]\right\}_{k=1}^{T}, \lambda_t, \epsilon} f\left(\Phi_{x,i}^{(t)}, \Phi_{u,i}^{(t)}, Q_i, R_i\right) = \sum_{k=1}^{T} \left\| Q\Phi_{x,i}^{(t)}[k] + R\Phi_{u,i}^{(t)}[k] \right\|_1 + r_c\epsilon \tag{9}$$

s.t. $\{\Phi_x^{(t)}, \Phi_u^{(t)}\} \in \mathcal{L}_d \cap \mathcal{F}_T$ and (8)

Remark 2 *The objective function described here is equivalent to the two-part objective function in (6). 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.*

4. 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 3.2, namely with time-varying local sets $\mathcal{L}_d(i,t)$. The dynamics are now modeled as a time-varying hybrid system:

$$\mathbf{x}_i[t+1] = A_{ii}(\alpha(t))\mathbf{x}_i[t] + \sum_{j\in\mathcal{N}_i(\alpha(t))} A_{ij}\mathbf{x}_j[t] + B_i\mathbf{u}[t] + \mathbf{w}_i[t] \ \ \forall i \in \mathcal{V} \tag{10}$$

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 Olfati-Saber and Murray (2004); Xiao et al. (2005), 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 Xiao et al. (2005). Since the methodology is the same across all subsystems $i \in \mathcal{V}$, the subscript i is henceforth removed for notational simplicity.

Similar to (3), 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 (10). 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 (4) 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 (9) with the same communication delay matrix d_c defined in Section 3.2 and time horizon T.

5. Numerical Simulation

We simulate our iterative, localized scheme on a power grid network from Wang et al. (2018), 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 1. 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.

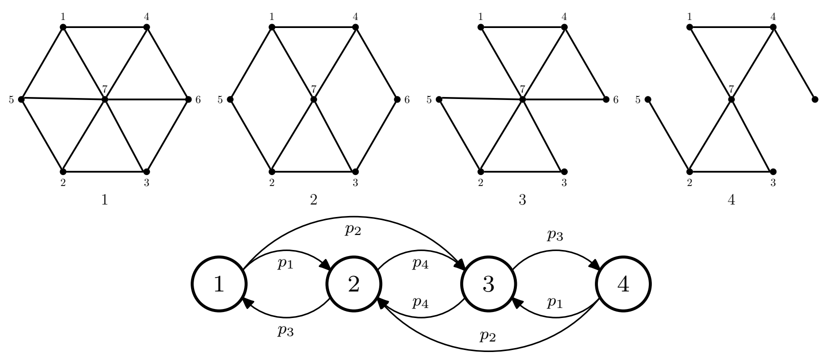

Figure 1: [Top] Hexagon topologies arranged in a Markov chain with four states. [Bottom] The state transition diagram for the four states.

In Figure 2, 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 3 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:

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

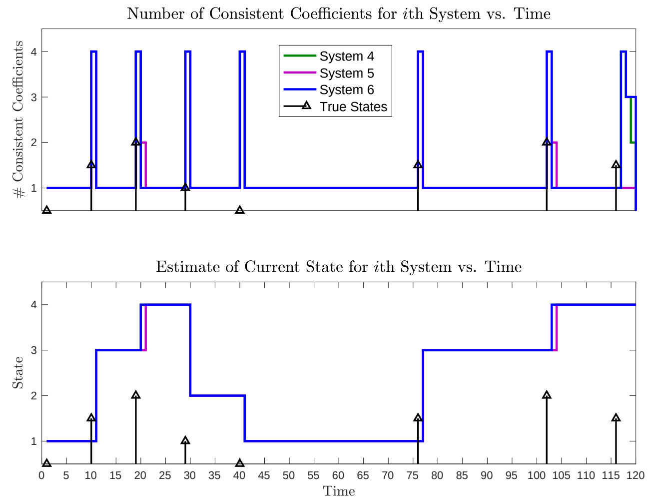

Figure 2: [Top] The number of consistent coefficients and [Bottom] the estimate of the current state for systems $4, 5$, and 6.

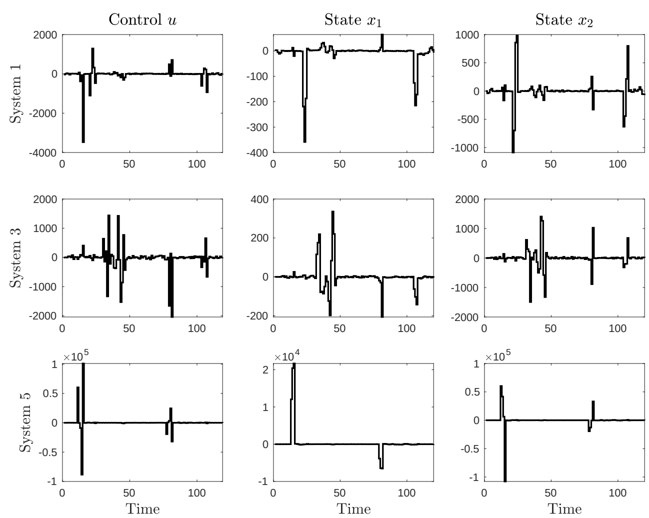

Figure 3: Control law and state evolutions for systems $1, 3$, and 5.

6. Conclusion

In conclusion, this paper developed a robust system-level controller synthesis approach to large-scale networked systems which adaptively learned topological changes over time. In particular, we considered systems which switched between topological configurations of a finite-state Markov Chain. We demonstrated via simulation that each subsystem managed to stabilize under a robust control law that used only local information, while adapting to the current topology within reasonable time after a switch is made. We emphasized throughout that learning precisely where the network structure has changed is unnecessary to stabilize the overall network due to the robust nature of the controller we design.

References

James Anderson, John C. Doyle, Steven Low, and Nikolai Matni. System level synthesis. *ArXiv preprint, arXiv:1904.01634*, 2019.

Long Cheng, Yunpeng Wang, Zeng-Guang Hou, Min Tan, and Zhiqiang Cao. Sampled-data based average consensus of second-order integral multi-agent systems: Switching topologies and communication noises. *Automatica*, 49(5):1458 – 1464, May 2013.

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

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

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *ArXiv preprint, arXiv:1710.01688*, 2017.

Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. Safely learning to control the constrained linear quadratic regulator. *ArXiv preprint, arXiv:1809.10121*, 2018.

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

Xiaohua Ge and Qing-Long Han. Distributed fault detection over sensor networks with Markovian switching topologies. *International Journal of General Systems*, 43(3-4):305–318, 2014.

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

Dimitar Ho and John C. Doyle. Scalable robust adaptive control from the system level perspective, 2019.

Petros A. Ioannou and Jing Sun. *Robust Adaptive Control*. Prentice-Hall, Inc., USA, 1995.

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

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

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

Jean-Jacques Slotine, Wei Wang, and Khalid Rifai. 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, Jan 2004.

Gang Tao. Multivariable adaptive control: A survey. *Automatica*, 50(11):2737 – 2764, 2014.

Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle. A system level approach to controller synthesis, 2016.

Yuh-Shyang Wang, Nikolai Matni, and John Doyle. Separable and localized system level synthesis for large-scale systems. *IEEE Transactions on Automatic Control*, 63(12):4234–4249, Dec 2018.

L. Xiao, S. Boyd, and S. Lall. 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, Apr 2005.