

# Distributed Linear Quadratic Regulator Robust to Communication Dropouts<sup>\*</sup>

C. Amo Alonso<sup>\*</sup> D. Ho<sup>\*</sup> J.M. Maestre<sup>\*\*</sup>

<sup>\*</sup> *Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA (e-mail: camoalon@caltech.edu, dho@caltech.edu).*

<sup>\*\*</sup> *Dep. of Systems and Automation Engineering, University of Seville, Spain (e-mail: pepemaestre@us.es)*

---

**Abstract:** We present a solution to deal with information package dropouts in distributed controllers for large-scale networks. We do this by leveraging the System Level Synthesis approach, a control framework particularly suitable for large-scale networks that addresses information exchange in a very transparent manner. To this end, we propose two different schemes for controller synthesis and implementation. The first one synthesizes a controller inherently robust to dropouts, which is later implemented in an offline fashion. For the second approach, we synthesize a collection of controllers offline and then switch between different controllers online depending on the current dropouts detected in the system. The two approaches are illustrated and compared by means of a simulation example.

*Keywords:* Distributed Control, System Level Synthesis, Modular Control, Communication Dropouts.

---

## 1. INTRODUCTION

In distributed control methods, local controllers coordinate their actions to deal with mutual interaction issues and improve overall system performance. The information exchanged is typically performed using a communication network, which is a potential source of vulnerability due to well-known problems as packet losses and cyber-aggressions. See for example (Lun et al., 2019) for a comprehensive review of cyber-physical systems security and (Sandberg et al., 2015), where attack and defense strategies are shown for Network Control Systems.

Unreliable networks where packet dropouts can occur have been explored in the past through many different control methods. For instance, in (Quevedo et al., 2015; Mishra et al., 2018) the features of stochastic model predictive control are exploited to buffer the input sequence and mitigate communication losses, and in Cetinkaya et al. (2015), where time-inhomogeneous Markov chains are used to model random packet losses in a control scheme where a feedback gain is used.

In this paper, we focus on a version of the linear quadratic regulator (LQG) problem that incorporates communication constraints as well as probabilistic communication dropouts between subcontrollers. To do this, we study this problem in the context of the recently proposed System Level Synthesis (SLS) framework (Anderson et al., 2019; Wang et al., 2017; Matni et al., 2017), which allows for the synthesis and implementation of distributed and localized controllers in a scalable way, making it a very suitable framework for large-scale networks. In particular, the SLS framework explicitly accounts for the information structure of the network, so the sub-controllers are sparsely connected, i.e., each of the sub-controllers only exchanges information with other sub-controllers within its communication range. Besides, it allows to localize the effects of the disturbances within that range, which ultimately allows the sub-

controllers to only access local information and solve for local subproblems of much smaller complexity than the complexity of the whole problem. Given this ability to localize, together with the explicit dependence on the information structure, it is important to analyze how an SLS approach can be affected by packet losses, and also to propose means to relieve this issue.

In this context, our main contribution is an algorithm for synthesis and implementation of SLS controllers that allows for a time-varying communication topology to deal with packet losses. To this end, two different approaches are proposed:

- Offline distributed controller synthesis and implementation of SLS controllers intrinsically robust to communication dropouts. This requires minimal online computations.
- Offline synthesis of SLS controllers together with an online implementation strategy that adapts the controllers to different communication topologies based on the sensed package losses. In contrast to the first approach, this procedure requires more online computations, but comes with increased control performance as it can adapt to changing communication topologies.

Since SLS is a natural framework to perform distributed control and the communication structure appears explicitly in the formulation, the presented derivations are carried out by leveraging the SLS framework. All the results are suitably distributed among the sub-controllers of the network. Finally, a simulation example is given to illustrate the proposed methods.

The remainder of this paper is organized as follows. In Section 2, we present the problem formulation. Section 3 introduces the offline distributed synthesis and implementation strategies using the robust SLS framework, and we provide a result for the distribution of the robust SLS synthesis. Section 4 presents an algorithm for offline distributed synthesis of SLS robust controllers together with an online distributed implementation that allows to tackle the communication dropouts. In Section 5,

---

<sup>\*</sup> This work was partially supported by the European Research Council (Advanced Research Grant 769051-OCNTSOLAR) and the MINECO-Spain project DPI2017-86918-R.

we illustrate the proposed approach via simulation. We end in Section 6 with conclusion, and directions for future work.

**Notation.** Lower-case and upper-case Latin and Greek letters such as  $x$  and  $A$  denote vectors and matrices respectively, although lower-case letters might also be used for scalars or functions (the distinction will be apparent from the context). We use bracketed indices to denote the time of the true system, i.e., the system is at state  $x(t)$  at time  $t$ . Subscripts denote time indices within a loop, i.e.,  $x_t$  denotes the  $t^{\text{th}}$  state within the loop. To denote subsystem variables, we use square bracket notation, i.e.  $[x]_i$  denotes the components of vector  $x$  that correspond to subsystem  $i$ . Boldface lower and upper case letters such as  $\mathbf{x}$  and  $\mathbf{K}$  denote finite horizon signals and lower block triangular operators, respectively:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \end{bmatrix}, \mathbf{K} = \begin{bmatrix} K^{0,1} & 0 \\ K^{0,2} & K^{1,1} \\ \vdots & \ddots & \ddots \end{bmatrix},$$

where each  $K^{i,j}$  is a matrix of compatible dimension. In this notation,  $\mathbf{K}$  is the matrix representation of the convolution operation induced by a time varying controller  $K_t(x_{0:t})$ , so that  $u_t = \mathbf{K}^{t,:} \mathbf{x}$ , where  $\mathbf{K}^{t,:}$  represents the  $t$ -th block-row of  $\mathbf{K}$ .

## 2. PROBLEM STATEMENT

Consider a discrete-time linear time invariant (LTI) dynamical system with dynamics:

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad (1)$$

where  $x(t) \in \mathbb{R}^n$  is the state,  $u(t) \in \mathbb{R}^p$  is the control input, and  $w(t) \in \mathbb{R}^n$  is an exogenous disturbance, which we assume to be additive-white Gaussian noise. The system is structured and can be described as a collection of  $N$  interconnected subsystems, with local state, control, and disturbance inputs given by  $[x]_i$ ,  $[u]_i$ , and  $[w]_i$  for each subsystem  $i$ . Accordingly if  $\mathcal{S}$  is a set of subsystems, we will understand  $[x]_{\mathcal{S}}$ ,  $[u]_{\mathcal{S}}$  to be the concatenation of  $[x]_i$ ,  $[u]_i$  for all  $i \in \mathcal{S}$ . The matrices  $A$  and  $B$  can also be partitioned into a compatible local block structure  $[A]_{ij}$ ,  $[B]_{ij}$  so the local dynamics for subsystem  $i$  are:

$$[x]_i(t+1) = \sum_j [A]_{ij} [x]_j(t) + \sum_j [B]_{ij} [u]_j(t) + [w]_i(t).$$

The interconnection topology of the system can be modeled as a time-invariant unweighted directed graph  $\mathcal{G}(E, V)$ , where each subsystem  $i$  is identified with a vertex  $v_i \in V$  and an edge  $(v_i, v_j) \in E$  exists whenever  $[A]_{ij} \neq 0$  or  $[B]_{ij} \neq 0$ .

Inspired by control applications in the large-scale system setting, we will assume that subcontroller  $[u]_i$  is only able to directly measure its own subsystem state  $[x]_i$ , but can receive information with no delay from a small set of neighboring sub-controllers  $\bar{\mathcal{V}}^{in}(i) \in V$  through some kind of communication network and can send information with no delay to a small set of neighboring sub-controllers  $\bar{\mathcal{V}}^{out}(i) \in V$  in the same manner. To model this interaction more precisely, assume that every controller  $i$  has an internal state  $[p_t]_i$  and at every time-step it performs the following two operations in order:

- (1) compute message  $[m_t]_i$  based on history of internal state and own measurement  $[x_t]_i$

$$[m_t]_i = [h_t]_i([x_t]_i, [p_t]_i, [p_{t-1}]_i, \dots), \quad (2)$$

- (2) exchange messages with neighbors and update internal state:

$$[p_t]_i = [m_t]_{\bar{\mathcal{V}}^{in}(i)}, \quad (3)$$

- (3) compute control action based on own measurement  $[x_t]_i$  and history of internal state  $[p_t]_i$ :

$$[u_t]_i = [k_t]_i([x_t]_i, [p_t]_i, [p_{t-1}]_i, \dots). \quad (4)$$

In real applications communication networks rarely offer a continuously reliable connection between subsystems. Thus, we will assume that some communication links in the network can occasionally drop packets. We model this by defining the constraint that subsystem  $i$  at time  $t$  can receive information only from the subset  $\mathcal{V}_t^{in}(i) \subset \bar{\mathcal{V}}^{in}(i)$  and can only send information to subset  $\mathcal{V}_t^{out}(i) \subset \bar{\mathcal{V}}^{out}(i)$ . We define  $\mathcal{D}_t^{out}(i) := \bar{\mathcal{V}}^{out}(i) / \mathcal{V}_t^{out}(i)$  as the subset of systems which dropped packages from system  $i$ , and assume that for each  $i$  there is a small subset of neighbors  $\mathcal{V}_t^{in}(i) \subset \bar{\mathcal{V}}^{in}(i)$  and  $\mathcal{V}_t^{out}(i) \subset \bar{\mathcal{V}}^{out}(i)$  that guarantee communication without dropouts.

If we consider dropouts, then the controller implementation gets perturbed by replacing update step (3) with

$$[p_t]_i = [m_t]_{\mathcal{V}_t^{in}(i)}, \quad (3')$$

which simply states that the internal state update step does not get all messages of its neighbors.

We will model the dropouts occurring for every subsystem  $i$  as a random process, in particular we will assume  $\mathcal{V}_t^{in}(i)$  to be independent random variables over  $t$ , that are identically distributed according to some probability mass function

$$f_i : 2^{\bar{\mathcal{V}}^{in}(i) / \mathcal{V}_t^{in}(i)} \rightarrow [0, 1]. \quad (5)$$

**Remark 1.**  $2^{\bar{\mathcal{V}}^{in}(i) / \mathcal{V}_t^{in}(i)}$  means the power set of all possible subsystems  $\bar{\mathcal{V}}^{in}(i) / \mathcal{V}_t^{in}(i)$  that could drop communication to subsystem  $i$ .

Furthermore, assume the random sets  $\mathcal{V}_t^{in}(i)$  are also independent for different  $i$ . We will refer to the joint-probability mass function as  $f$  and we will abbreviate

$$\mathcal{V}_t^{in} := \{\mathcal{V}_t^{in}(1), \mathcal{V}_t^{in}(2), \dots, \mathcal{V}_t^{in}(N)\}$$

as the collection of subsets of  $\{\mathcal{V}_t^{in}(i)\}$  at time  $t$ .

With these definitions at hand, we can formulate our goals as to minimize the expected average quadratic cost over the gaussian disturbance  $w$  and the distribution of the dropouts occurring, i.e.,

$$\begin{aligned} \min_{\{h_t, k_t\}} \quad & \mathbb{E}_{\mathcal{V}_t^{in} \sim f} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \mathbb{E}[x_t^T Q x_t + u_t^T R u_t] \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t + w_t \quad \forall t = 0, 1, \dots, \\ & (2), (4), (3'), \end{aligned} \quad (6)$$

where  $\mathbb{E}$  is the mathematical expectation and  $(Q, R)$  are positive definite cost matrices.

In this work, we propose two ways to tackle this problem, both formulated in the SLS framework since it provides a tractable way to deal with the communication constraint (3). The first strategy is to tackle the problem with an offline controller. To do this, we will synthesize a controller offline inherently robust to a specified set of communication dropout patterns. This will penalize performance but will guarantee stability, as we will show in section 4. We also present a different approach where different controllers for different information exchange topologies are synthesized offline, which then are implemented by an online controller that senses dropouts and changes the communication strategy while guaranteeing stability. Performance of these two strategies is compared through simulation.

### 3. PRELIMINARIES: SYSTEM LEVEL SYNTHESIS

In this section we present an abridged version of the SLS framework (Anderson et al. (2019); Matni et al. (2017); Wang (2017) and references therein). The SLS framework will be used in the derivations presented in this paper due to its ability to impose locality constraints, as well as its ability to distribute both controller synthesis and implementation. Recent work has extended the SLS approach to the time-varying system (Anderson et al. (2019); Ho and Doyle (2019)) and even general nonlinear systems (Ho (2020)).

#### 3.1 Time domain System Level Synthesis

Consider the system with dynamics (1). Let  $u_t$  be a causal time-varying state-feedback control law, i.e.,  $u_t = K_t(x_0, x_1, \dots, x_t)$  where  $K_t$  is some linear map. We will denote  $\mathbf{K}$  the general bounded causal linear operator mapping state sequences  $\mathbf{x}$  to input sequences  $\mathbf{u} = \mathbf{K}\mathbf{x}$ . Although rather informal, for ease of exposition we will think of linear bounded operators as infinite dimensional lower triangular matrices. To this end, define  $\mathbf{A} := \text{blkdiag}(A, A, \dots)$ ,  $\mathbf{B} := \text{blkdiag}(B, B, \dots)$  as infinite dimensional block-diagonal matrices with  $A$  and  $B$  on the diagonal and let  $Z$  be the block-downshift operator<sup>1</sup>. Using this notation, the dynamics (1) impose the following relationship between the state  $\mathbf{x}$ , input  $\mathbf{u}$  and disturbance  $\mathbf{w}$ :

$$\mathbf{x} = Z(\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{w}.$$

Overall, the closed loop behavior of the system in (1) under the feedback law  $\mathbf{K}$  can be entirely characterized by

$$\mathbf{x} = (I - Z(\mathbf{A} + \mathbf{B}\mathbf{K}))^{-1}\mathbf{w} =: \Phi_x \mathbf{w} \quad (7a)$$

$$\mathbf{u} = \mathbf{K}(I - Z(\mathbf{A} + \mathbf{B}\mathbf{K}))^{-1}\mathbf{w} =: \Phi_u \mathbf{w}, \quad (7b)$$

where the operators  $\Phi_x$  and  $\Phi_u$  are the closed loop maps from the disturbance  $\mathbf{w}$  to the state  $\mathbf{x}$  and control input  $\mathbf{u}$ , respectively.

The SLS framework relies on this parametrization for the optimal controller synthesis to be performed by directly optimizing over the system responses  $\Phi_x$  and  $\Phi_u$ , as opposed to the controller map  $\mathbf{K}$  itself.

**Theorem 3.1.** *For the dynamics (1) state feedback law  $\mathbf{K}$  defining the control action as  $\mathbf{u} = \mathbf{K}\mathbf{x}$ , the following are true*

(1) *the affine subspace defined by*

$$[I - ZA \quad -ZB] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \quad (8)$$

*parameterizes all possible system responses.*

(2) *for any transfer matrices  $\{\Phi_x, \Phi_u\}$  satisfying (8), the controller  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  achieves the desired response.*

Hence, by using the SLS framework optimal control problems can be reformulated into convex optimization problems. A detailed description on how to do this is provided in (Anderson et al., 2019). One of the main advantages of formulating an optimal control problem with the system response parametrization is that it allows to impose (2), (3) and (4) in a convex manner by imposing the system responses to be localized. As shown in (Anderson et al., 2019), communication constraints on controllers of the form  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  can be easily incorporated into optimal control problems, by requiring that the closed loop maps  $\{\Phi_x, \Phi_u\}$  lie in a suitably chosen subspace  $\mathcal{S}$ . Overall, many traditionally non-convex optimal control problems can

<sup>1</sup> A matrix with identity matrices along its first block sub-diagonal and zeros elsewhere.

be recast into their equivalent convex SLS representation of the form

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \quad & g(\Phi_x, \Phi_u) \\ \text{s.t.} \quad & [I - ZA \quad -ZB] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = I \\ & \{\Phi_x, \Phi_u\} \in \mathcal{S}, \end{aligned} \quad (9)$$

where the optimization is phrased over the feasible closed loop maps  $\{\Phi_x, \Phi_u\}$ . Moreover,  $\mathcal{S}$  can incorporate other convex constraints imposed on the system responses, i.e., finite impulse response (FIR), performance bounds, etc.

#### 3.2 Virtually localizable System Level Synthesis

In the previous subsection the SLS framework was presented. Oftentimes the constraint  $\{\Phi_x, \Phi_u\} \in \mathcal{S}$  can be too restrictive to impose on the closed loop maps, while it is only used for certifying that certain communication constraints on the controller  $\mathbf{K} = \Phi_u \Phi_x^{-1}$  are enforced. A robust-variant of the SLS parametrization described above was introduced by Matni et al. (2017), and later generalized in (Ho and Doyle, 2019) for the general setting of time-varying systems, which addresses this issue. The following general approach can be found in Anderson et al. (2019) and is based on the following result:

**Theorem 3.2.** *Let  $(\tilde{\Phi}_x, \tilde{\Phi}_u, \Delta)$  be a solution to*

$$[I - ZA \quad -ZB] \begin{bmatrix} \tilde{\Phi}_x \\ \tilde{\Phi}_u \end{bmatrix} = (I + \Delta).$$

*Then, the controller implementation*

$$\mathbf{u} = \tilde{\Phi}_u \hat{\mathbf{w}}, \quad \hat{\mathbf{x}} = (I - \tilde{\Phi}_x) \hat{\mathbf{w}}, \quad \hat{\mathbf{w}} = \mathbf{x} - \hat{\mathbf{x}}, \quad (10)$$

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

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \tilde{\Phi}_x \\ \tilde{\Phi}_u \end{bmatrix} (I + \Delta)^{-1} \mathbf{w}. \quad (11)$$

This theorem allows for a reformulation of problem (9), which allows to search over controller implementations  $\mathbf{K} = \tilde{\Phi}_u \tilde{\Phi}_x^{-1}$  where  $\tilde{\Phi}_x, \tilde{\Phi}_u$  do not have to be necessarily closed loop maps. As stated in (Anderson et al., 2019), a sufficient condition for  $(I + \Delta)^{-1}$  to be stable is that any of the induced norms  $\|\Delta\|_{\mathcal{H}_\infty}, \|\Delta\|_{\mathcal{L}_1}, \|\Delta\|_{\varepsilon_1}$  are smaller than  $< 1$ . In particular, the optimal control problem (9) can be relaxed to:

$$\begin{aligned} \min_{\tilde{\Phi}_x, \tilde{\Phi}_u, \Delta} \quad & g\left(\begin{bmatrix} \tilde{\Phi}_x \\ \tilde{\Phi}_u \end{bmatrix} (I + \Delta)^{-1}\right) \\ \text{s.t.} \quad & [I - ZA \quad -ZB] \begin{bmatrix} \tilde{\Phi}_x \\ \tilde{\Phi}_u \end{bmatrix} = I + \Delta \\ & \{\tilde{\Phi}_x, \tilde{\Phi}_u\} \in \mathcal{L}_d \cup \mathcal{S}, \quad \|\Delta\|_* < 1, \end{aligned} \quad (12)$$

where  $*$  in  $\|\cdot\|_*$  is one of  $\mathcal{H}_\infty, \mathcal{L}_1, \varepsilon_1$ .

## 4. CONTROLLER STRUCTURE AND DISTRIBUTED SYNTHESIS

Here we introduce the reformulation of problem (6) into the SLS framework, and illustrate the controller structure and its implementation and the communication model. This will provide us with a set of tools that will be useful in the next sections.

In what follows, we use the notation  $\Phi_x^{t,k}$ ,  $\Phi_u^{t,k}$  and  $[\Phi_x^{t,k}]_{i,j}$ ,  $[\Phi_u^{t,k}]_{i,j}$  to index sub-matrices of  $\Phi_x$  and  $\Phi_u$

$$\Phi_x = \begin{bmatrix} \Phi_x^{0,1} & 0 & 0 & \cdots \\ \Phi_x^{0,2} & \Phi_x^{1,1} & 0 & \cdots \\ \Phi_x^{0,3} & \Phi_x^{1,2} & \Phi_x^{2,1} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (13)$$

$$\Phi_x^{t,k} = \begin{bmatrix} [\Phi_x^{t,k}]_{1,1} & [\Phi_x^{t,k}]_{1,2} & \cdots & [\Phi_x^{t,k}]_{1,n} \\ [\Phi_x^{t,k}]_{2,1} & [\Phi_x^{t,k}]_{2,2} & \cdots & [\Phi_x^{t,k}]_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ [\Phi_x^{t,k}]_{n,1} & [\Phi_x^{t,k}]_{n,2} & \cdots & [\Phi_x^{t,k}]_{n,n} \end{bmatrix} \quad (14)$$

and the same convention is understood for  $\Phi_u$  and  $\Delta$ . Notice that we are abusing notation, and although we are referring to the system responses in (11), we will drop the tilde for convenience. Also, we will use the abbreviation for the columns

$$[\phi_x^t]_i := [\Phi_x^{t,:}]_{:,i} \quad [\phi_u^t]_i := [\Phi_u^{t,:}]_{:,i}. \quad (15)$$

With this notation, we can write the controller implementation (10) into its distributed form:

$$[\hat{w}_t]_j = [x_t]_j - \sum_i \sum_{k=2}^{t+1} [\Phi_x^{t+1-k,k}]_{j,i} [\hat{w}_{t+1-k}]_i \quad (16a)$$

$$[u_t]_j = \sum_i \sum_{k=1}^{t+1} [\Phi_u^{t+1-k,k}]_{j,i} [\hat{w}_{t+1-k}]_i. \quad (16b)$$

The above controller implementation can be put in the form of the communication model described in (2), with the following correspondence (3) in the SLS notation:

$$\begin{aligned} [m_t]_i &= \{[\phi_x^t]_i [\hat{w}_t]_i, [\phi_u^t]_i [\hat{w}_t]_i\} \\ [p_t]_i &= \{[\phi_x^t]_j [\hat{w}_t]_j, [\phi_u^t]_j [\hat{w}_t]_j\}_{j=1}^N. \end{aligned}$$

**Remark 2.** The above communication scheme expresses that every subsystem  $i$  decides on the columns  $[\phi_x^t]_i$ ,  $[\phi_u^t]_i$  and it shares it with all other subsystems and it can be verified that this is consistent with the original controller equations (16).

Recalling our original problem (6), notice that we can equivalently formulate the dropout constraint

$$[p_t]_i = [m_t]_{\mathcal{V}_t^{in}(i)}$$

as the following sparsity constraint on  $[\phi_x^t]_j$  and  $[\phi_u^t]_j$ :

$$[[\phi_x^t]_i]_{\mathcal{D}_t^{out}(i)} = 0 \Leftrightarrow [\phi_x^t]_i = [[\phi_x^t]_i]_{\mathcal{V}_t^{out}(i)} \quad (17a)$$

$$[[\phi_u^t]_i]_{\mathcal{D}_t^{out}(i)} = 0 \Leftrightarrow [\phi_u^t]_i = [[\phi_u^t]_i]_{\mathcal{V}_t^{out}(i)}, \quad (17b)$$

where  $[[\phi_x^t]_j]_{\mathcal{S}}$  stands for

$$[[\phi_x^t]_i]_{\mathcal{S}} := [\Phi_x^{t,:}]_{\mathcal{S},i}, \quad [[\phi_u^t]_i]_{\mathcal{S}} := [\Phi_u^{t,:}]_{\mathcal{S},i}.$$

With this in mind, together with equation (11), we can reformulate optimization (6) into a robust SLS problem<sup>2</sup> as:

$$\begin{aligned} \min_{\Phi_x, \Phi_u, \Delta} \quad & \mathbb{E}_{\mathcal{V}_t^{in} \sim f} \left\| [\mathbf{Q} \ \mathbf{R}] \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} (I + \Delta)^{-1} \right\|_{\mathcal{H}_2} \\ \text{s.t.} \quad & Z_{AB}^T \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = \Delta, \quad \|\Delta\|_* < 1, \\ & [\phi_x^t]_i = [[\phi_x^t]_i]_{\mathcal{V}_t^{out}(i)}, \quad [\phi_u^t]_i = [[\phi_u^t]_i]_{\mathcal{V}_t^{out}(i)} \\ & \forall t = 0, 1, \dots \end{aligned} \quad (18)$$

<sup>2</sup> Notice that, as shown in Wang et al. (2014), the localized LQR cost in (6) in the presence of additive-white Gaussian noise can be reformulated into the  $\mathcal{H}_2$  norm of the weighted transfer matrices  $\Phi_x$  and  $\Phi_u$

where  $Z_{AB}^T = [Z^T - A, -B]$ ,  $\mathbf{Q} := \text{blkdiag}(Q^{\frac{1}{2}}, Q^{\frac{1}{2}}, \dots)$ ,  $\mathbf{R} := \text{blkdiag}(R^{\frac{1}{2}}, R^{\frac{1}{2}}, \dots)$ , and  $*$  in  $\|\cdot\|_*$  is one of  $\mathcal{H}_\infty, \mathcal{L}_1, \varepsilon_1$ . Notice that the last constraint is using the notation  $\mathcal{V}_k^{out}(i)$ , which accounts for the presence of dropouts for each  $k < t \forall t$ .

#### 4.1 Relaxation of $\mathcal{H}_2$ Optimal Robust controller synthesis

Equation (18) presents a reformulation of the original problem (6) into one featuring the SLS formulation. This reformulation will allow to make this problem tractable given the transparency with which communication structure constraints can be imposed in SLS. However, one of the major limitations of (18) is that the objective function is non-convex in  $\Delta$ . Not only that, but it is not obvious how to perform the computation distributedly among subcontrollers. In what follows we present a way to solve a relaxation of (18) by means of a quasi-distributed convex optimization. In the following derivation we choose the  $\mathcal{L}_1$  norm as the robustness criterion for  $\|\cdot\|_*$  and give an appropriate bound on the  $\mathcal{H}_2$  norm.

We start by recalling a result on Neumann series of bounded linear operators from Lemma A.1 (Appendix A). It can be leveraged to obtain the following  $\mathcal{H}_2$  inequality for linear time-invariant operators in terms of the  $\|\cdot\|_{2\leftarrow 1}$ -norm as in (4.1).

**Lemma 4.1.** Let  $\Phi$  be a time-invariant linear operator with  $[\Phi^{t,k}]_{i,j}$  and  $[\phi^t]_i$  referring to the decompositions in (13), (14), (15). If  $\|\Phi\|_{\mathcal{H}_2} < \infty$ , then

$$\|\Phi\|_{2\leftarrow 1} := \sup_{\|w\|_1 \leq 1} \|\Phi w\|_2 = \max_{i,t} \sqrt{\|[\phi^t]_i\|_F^2}. \quad (19)$$

*Proof.* See Appendix A.  $\square$

**Lemma 4.2.** Assume  $\mathbf{G}, \Delta$  are linear time-invariant operators of dimension  $n$  with  $\|\Delta\|_{\mathcal{L}_1} \leq \lambda < 1$  and  $\|\mathbf{G}\|_{\mathcal{H}_2} < \infty$ . Then the following bound holds:

$$\|\mathbf{G}(\mathbf{I} - \Delta)^{-1}\|_{\mathcal{H}_2} \leq \|\mathbf{G}\|_{2\leftarrow 1} \frac{n}{1 - \lambda}.$$

*Proof.* Using Lemma A.1 we have  $\mathbf{G}(\mathbf{I} - \Delta)^{-1} = \sum_{k=0}^{\infty} \mathbf{G}\Delta^k$ ,

where  $\Delta^k = \underbrace{\Delta \circ \Delta \circ \cdots \circ \Delta}_k$ . First consider the  $\mathcal{H}_2$  norm of

the terms  $\mathbf{G}\Delta^k$ . We can write  $\|\mathbf{G}\Delta^k\|_{\mathcal{H}_2}^2 = \sum_{i=1}^n \|\mathbf{G}\Delta^k \mathbf{s}^{(i)}\|_2^2$

where  $\mathbf{s}^{(i)}$  is the sequence where  $s_0^{(i)} = e_i$ <sup>3</sup> and  $s_k^{(i)} = 0$  for all  $k > 0$ . Since,  $\|\mathbf{s}^{(i)}\|_1 = 1$  and  $\|\Delta^k\|_{\mathcal{L}_1} \leq \|\Delta\|_{\mathcal{L}_1}^k \leq \lambda^k$ , we have  $\|\Delta^k \mathbf{s}^{(i)}\|_1 \leq \lambda^k$ . This gives us the bound:

$$\|\mathbf{G}\Delta^k\|_{\mathcal{H}_2} \leq \sum_{i=1}^n \|\mathbf{G}\Delta^k \mathbf{s}^{(i)}\|_2 \leq n \|\mathbf{G}\|_{2\leftarrow 1} \lambda^k$$

and leads to the desired result:

$$\left\| \sum_{k=0}^{\infty} \mathbf{G}\Delta^k \right\|_{\mathcal{H}_2} \leq \sum_{k=0}^{\infty} n \|\mathbf{G}\|_{2\leftarrow 1} \lambda^k \leq \|\mathbf{G}\|_{2\leftarrow 1} \frac{n}{1 - \lambda}.$$

$\square$

Lemma 4.2 gives an upper bound on the objective function of problem (18). Hence, problem (18) can now be written as:

<sup>3</sup>  $e_i$  is the unit vector in the  $i^{\text{th}}$  entry.

$$\begin{aligned} \min_{\Phi_x, \Phi_u, \lambda} \quad & \mathbb{E}_{\mathcal{V}_t^{in} \sim f} \left\| \begin{bmatrix} \mathbf{Q} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} \right\|_{2 \leftarrow 1} \frac{n}{1-\lambda} \\ \text{s.t.} \quad & Z_{AB}^T \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} = \Delta, \quad \|\Delta\|_{\mathcal{L}_1} < \lambda, \\ & [\phi_x^t]_i = [[\phi_x^t]_i]_{\mathcal{V}_t^{out}(i)}, \quad [\phi_u^t]_i = [[\phi_u^t]_i]_{\mathcal{V}_t^{out}(i)} \\ & \forall t = 0, 1, \dots \end{aligned} \quad (20)$$

Problem (20) is a quasi-convex problem. Moreover, the  $\mathcal{L}_1$ -norm constraint can be decomposed in a convenient manner that facilitates distributed computation. In fact, applying our notational convention (13)-(15), on  $\Delta$ , we get the relation

$$[Z_{AB}^T] \begin{bmatrix} [\phi_x^t]_i \\ [\phi_u^t]_i \end{bmatrix} = [\delta^t]_i, \quad (21)$$

where  $[\Delta^{t,\cdot}]_{:,i} := [\delta^t]_i$ . Moreover, it can be verified that the term  $\|\Delta\|_{\mathcal{L}_1}$  can be equivalently written as

$$\|\Delta\|_{\mathcal{L}_1} = \max_{t,i} \|[\delta^t]_i\|_1, \quad (22)$$

which allows to write the robustness constraint (20) as

$$\max_{t,i} \|[\delta^t]_i\|_1 \leq \lambda.$$

Recalling, that the  $\|\cdot\|_{2 \leftarrow 1}$  norm also decomposes w.r.t.  $[\phi_x^t]_i$  and  $[\phi_u^t]_i$ , we can combine equation (19),(21) and optimization (20), to compute a relaxation of the robust optimal control problem (18) that decomposes. Furthermore, this allows for distributed computation, except for a global optimization over  $\lambda$  and  $c$ , which can be solved via bisection in a distributed way through a consensus algorithm. In summary, every subsystem  $i$  will solve the following optimization:

$$\begin{aligned} \min_{[\phi_x^t]_i, [\phi_u^t]_i, \lambda, c} \quad & c \\ \mathbb{E}_{\mathcal{V}_t^{in} \sim f} \quad & \sqrt{\left\| \begin{bmatrix} \mathbf{Q} \end{bmatrix}_i \begin{bmatrix} \mathbf{R} \end{bmatrix}_i \begin{bmatrix} [\phi_x^t]_i \\ [\phi_u^t]_i \end{bmatrix} \right\|_F^2} \frac{n}{1-\lambda} \leq c \\ \text{s.t.} \quad & \left\| \begin{bmatrix} Z_{AB}^T \end{bmatrix} \begin{bmatrix} [\phi_x^t]_i \\ [\phi_u^t]_i \end{bmatrix} \right\|_1 < \lambda, \\ & [\phi_x^t]_i = [[\phi_x^t]_i]_{\mathcal{V}_t^{out}(i)}, \quad [\phi_u^t]_i = [[\phi_u^t]_i]_{\mathcal{V}_t^{out}(i)} \\ & \forall t = 0, 1, \dots \end{aligned} \quad (23)$$

where,  $[\mathbf{Q}]_i$  and  $[\mathbf{R}]_i$  are the corresponding local blocks of subsystem  $i$  for  $Z_{AB}$ ,  $\mathbf{Q}$  and  $\mathbf{R}$  respectively.

#### 4.2 Stability guarantees for a time-varying implementation

It is important to note that the controller synthesis problem (23) corresponds to a LTI controller. However, the actual closed loop is time-varying due to the time-varying communication dropouts that can occur in the network. In the following lemma we show that column-wise switches<sup>4</sup> between any number of  $\mathcal{L}_1$  robust controllers do not render the closed-loop unstable.

**Lemma 4.3.** Consider the controller (16), where  $\{[\phi_x^t]_i, [\phi_u^t]_i\} \subset \{\{\psi_x^s, \psi_u^s\}\}_{s=1}^M$  where each  $\{\psi_x^s, \psi_u^s\}$  are partial system responses that satisfy

$$\left\| \begin{bmatrix} Z_{AB}^T \end{bmatrix} \begin{bmatrix} [\psi_x^s] \\ [\psi_u^s] \end{bmatrix} \right\|_1 < 1,$$

then the closed loop system is stable.

<sup>4</sup> A column-wise switch means that we can change one or more columns of the transfer matrices  $\Phi_x$  and  $\Phi_u$ .

*Proof.* This result follows directly from the relationship (22). Recall that  $\Delta$  can be written as

$$\|\Delta\|_{\mathcal{L}_1} = \max_{t,i} \|[\delta^t]_i\|_1,$$

but since  $\{[\phi_x^t]_i, [\phi_u^t]_i\} \in \{\{\psi_x^s, \psi_u^s\}\}_{s=1}^M$ , we have

$$\|\Delta\|_{\mathcal{L}_1} = \max_{t,i} \|[\delta^t]_i\|_1 = \max_s \left\| \begin{bmatrix} Z_{AB}^T \end{bmatrix} \begin{bmatrix} [\psi_x^s] \\ [\psi_u^s] \end{bmatrix} \right\|_1 < 1,$$

which proves stability per small gain theorem.  $\square$

Notice that this lemma is only valid under the assumption that we have  $\mathcal{L}_1$  robustness, i.e.  $\|\Delta\|_{\mathcal{L}_1} < 1$ , and given that the combinations are done column-wise. If this were not the case, an argument similar to the ones presented in (Ho and Doyle, 2019) would need to be made.

## 5. OFFLINE ROBUSTNESS TO DROPOUTS

In this section, we introduce a strategy to make the controller robust to communication dropouts. We take advantage of the formulation (23) and Lemma 4.3 and we propose an offline controller synthesis that accounts for communication dropouts, making the resulting controller intrinsically robust to communication dropouts. Since robustness is guaranteed by the synthesis process, we can implement this controller offline. The resulting controller is a very low-cost controller robust to package losses.

### 5.1 Offline controller synthesis

Let us start by considering that the general problem (18) with time invariant  $\Phi_x$  and  $\Phi_u$ . We need to guarantee that the resulting control will be robust to all the dropouts suffered by the communication network. To do that, we need to consider all the different sparsity patterns induced by all the possible dropouts, and guarantee satisfaction of the constraints for all of them. According to the definition,  $\mathcal{D}_t^{out}$  is associated with a given dropout at a certain time  $t$ . From equations (17), we have that each  $\mathcal{D}_t^{out}$  induces a sparsity pattern in  $\Phi_x$  and  $\Phi_u$ . We can characterize all the dropouts scenarios generated by the probability mass function  $f$ , under which the controller is required to be robust as the set of sets  $\mathcal{D}$ , so  $\mathcal{D}_t^{out} \in \mathcal{D}$  for each of the dropouts. Then, (18) can be written as:

$$\begin{aligned} \min_{\Phi_x, \Phi_u} \quad & \sum_{\mathcal{S} \in \mathcal{D}} f(\mathcal{S}) \left\| \begin{bmatrix} \mathbf{Q} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \Phi_x[\mathcal{S}] \\ \Phi_u[\mathcal{S}] \end{bmatrix} (I + \Delta)^{-1} \right\|_{\mathcal{H}_2} \\ \text{s.t.} \quad & \left\| \begin{bmatrix} Z_{AB}^T \end{bmatrix} \begin{bmatrix} \Phi_x[\mathcal{S}] \\ \Phi_u[\mathcal{S}] \end{bmatrix} \right\|_{\mathcal{L}_1} < 1, \forall \mathcal{S} \in \mathcal{D}, \\ & [[\Phi_x]_i]_{V/\bar{\mathcal{V}}^{out}} = 0, \quad [[\Phi_u]_i]_{V/\bar{\mathcal{V}}^{out}} = 0. \end{aligned} \quad (24)$$

Notice that the tractability of (24) depends on the size of  $\mathcal{D}$  and  $\bar{\mathcal{V}}^{in}$  and  $\bar{\mathcal{V}}^{out}$ . Even though this can lead to a combinatorial explosion, since the problem can be solved in a distributed manner via equation (23) and the size of  $\bar{\mathcal{V}}^{in}$  and  $\bar{\mathcal{V}}^{out}$  is much smaller than the size of the network in structured systems, this represents a feasible approach in practice.

### 5.2 Offline controller implementation

The controller implementation is distributed and can be described by equations (16), where the maps  $\Phi_x$  and  $\Phi_u$  are obtained from the distributed synthesis (23) with the additional robustness constraints as discussed in the previous subsection.

Notice that although the controller was restricted to be LTI, the implementation of such a controller will be time varying due

to the time varying nature of the dropouts. This controller is however guaranteed to be stable by virtue of Lemma 4.3.

## 6. ONLINE ROBUSTNESS TO DROPOUTS

Here we introduce a different strategy to make the controller robust to communication dropouts. Once again, we take advantage of the formulation (23) and Lemma 4.3 and we propose an online controller synthesis that is able to sense communication dropouts instantaneously. In this design performance is not hurt by the robustness, at the expense that the controller needs an online implementation and is therefore time-varying.

### 6.1 Online controller synthesis

We start by considering the general problem (18) where the controller is allowed to be time varying. The goal is for the controller to perform the optimal action given the current dropout. In order to do so, it is important to recall that the controller is able to sense instantaneously the dropout experienced. This is a reasonable assumption assuming a handshake protocol is in place and communication occurs at a local scale. Hence, the optimal strategy would be to use the optimal  $\Phi_x$  and  $\Phi_u$  for each dropout when it occurs, therefore switching between optimal controllers based on the dropout experienced. Under this premise, the synthesis reduces to solving offline optimization (18) for all dropout cases, which can be rewritten as:

$$\begin{aligned} \min_{\Phi_x^S, \Phi_u^S} & \left\| \begin{bmatrix} \mathbf{Q} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \Phi_x \\ \Phi_u \end{bmatrix} (I + \Delta)^{-1} \right\|_{\mathcal{H}_2} \\ \text{s.t.} & \left\| Z_{AB}^T \begin{bmatrix} \Phi_x^S \\ \Phi_u^S \end{bmatrix} \right\|_{\mathcal{L}_1} < 1, \end{aligned} \quad (25)$$

for each dropout pattern  $\mathcal{S} \in \mathcal{D}$ . The number of optimizations to solve depends on the size of  $\mathcal{D}$  and the different sparsity patterns that they generate in  $\Phi_x$  and  $\Phi_u$ . Notice that problem (25) can be solved offline in a distributed manner via (23).

### 6.2 Online controller implementation

Once the controllers have been synthesized offline, the online implementation easily follows. If at every time  $t$  dropout  $\mathcal{D}_t^{\text{out}}(i)$  is sensed, subsystem  $i$  implements the  $[\phi_x^{t+1}]_i$  and  $[\phi_u^{t+1}]_i$  that were synthesized according to the sparsity pattern induced by  $\mathcal{D}_t^{\text{out}}(i)$ . The corresponding  $[\phi_x^{t+1}]_i$  and  $[\phi_u^{t+1}]_i$  are implemented according to (16). This repeats for each  $t$ .

## 7. SIMULATION EXPERIMENTS

In this section, we present simulation experiments of the two strategies introduced in this paper to deal with dropouts and we compare their performance for different dropout scenarios. To perform these experiments we choose the following dynamical system consisting of 10 nodes:

$$[x]_i(t+1) = 1.2 \sum_{j=i-2}^{i+2} \alpha_{ij} [x]_j(t) + 1.2 [u]_i(t),$$

where  $\alpha_{ij} = .4$  for  $j = i \pm 1$  and  $\alpha_{ij} = .2$  otherwise. For  $i = 1, 10$   $\alpha_{ii} = .6$ . Negative  $j$  are not considered. The communication is described by the following adjacency matrix:

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & \dots \\ 1 & 1 & 1 & 0 & 0 & \dots \\ 0 & 1 & 1 & 1 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}^d,$$

and we consider  $d$  to be the dropout parameter. If no dropouts are present  $d = 5$ . A dropout changes the value of  $d$ , so

$d \in \mathcal{D} = \{2, 3, 4, 5\}$ . The probability distribution of  $d$  over  $\mathcal{D}$  is uniform. We define one  $[d]_i \in \mathcal{D}$  for each subsystem  $i$ , which represents the sparsity induced in the  $i^{\text{th}}$  column of  $(G)^{[d]_i}$ .

The cost at each time is computed as  $C(t) = x(t)^T x(t) + u(t)^T u(t)$ , and the total cost for each simulation is computed as  $C = \sum_{t=0}^T C(t)$  for  $T = 100$ . The simulations are run comparing the two strategies – offline and online – subject to the same random noise and the same dropout scenario. An illustrative example of the state, input and communication topology is introduced in Figure 1.

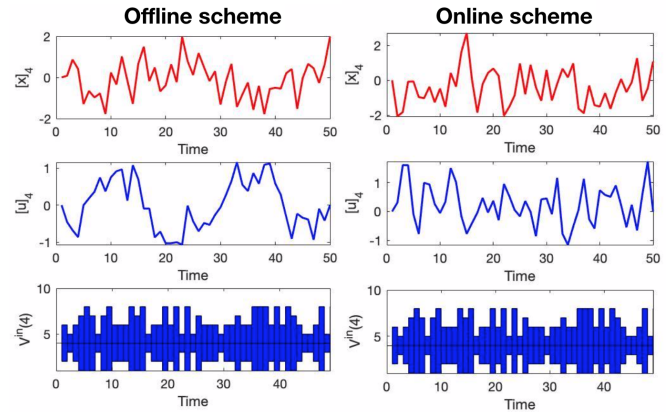


Fig. 1. On the left: state, input and communication topology for subsystem 4 using the offline strategy. On the right: identical representation for the online scheme.

Further, we compare the moving average defined as  $M = \frac{1}{T} \sum_{t=1}^T \mathbb{E}[C(t)]$ , where  $\mathbb{E}[C(t)]$  is by averaging the cost under 10 different Gaussian noise processes. We do so for different dropout scenarios. As illustrated by Figure 2, the online strategy performs better than the offline one. However, the gap is not dramatic, so the offline strategy is more cost efficient since it does not require an online implementation. Simulations suggest that both of these strategies are robust to communication dropouts while providing good performance, and the choice of using one versus the other is context dependent.

## 8. CONCLUSION

We presented two different strategies to deal with information packages dropouts in the communication network of distributed controllers by leveraging the SLS framework. The first strategy consists of an offline synthesis of a SLS controller constrained such that the resulting controller is inherently robust to communication dropouts. This controller can then be implemented in an offline fashion with the guarantee that it will be robust to dropouts. The second strategy consists of a synthesis of a collection of SLS controllers, each of them being optimal for a certain sparsity pattern. The implementation of the controller is carried out online, and each agent is able to choose a realization at each time step from the collection of synthesized controllers based on the current communication topology originated by the dropouts, which it can sense instantaneously. Notice that, although the controllers are synthesized as LTI, the controller implemented are time varying due to the time-varying nature of the communication network. We show in Lemma 4.3 that the controllers implemented are internally stabilizing. We also provide a relaxation to the robust version of SLS that allows for a distributed computation.

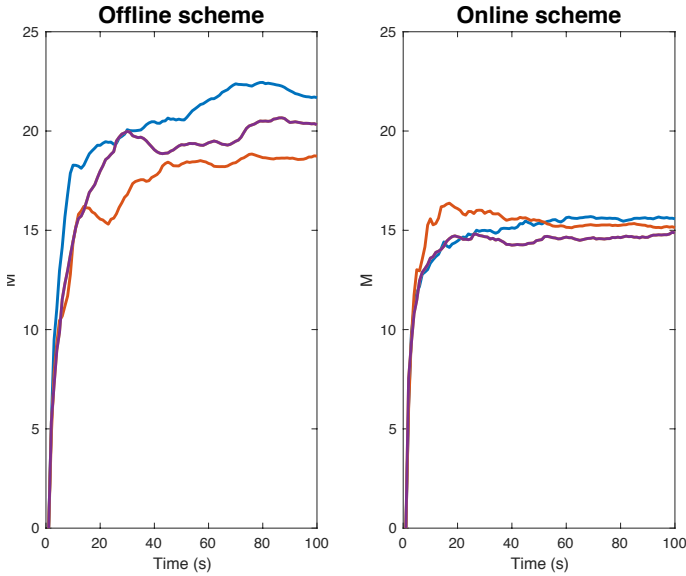


Fig. 2. On the left: moving average LQR cost computed over 10 disturbance processes for three different dropout scenarios for the offline scheme. On the right: identical representation for the online scheme.

This work represents a first step into the use of SLS as a tool to tackle communication problems in a networked control setting. Here we only discuss the extensions for the linear time-varying case, but remark that it is an interesting open problem whether the presented techniques can be extended to an even a broader setting, using the SLS approach for nonlinear systems introduced in (Ho (2020)). As for future work, it remains an open question how to tackle communication dropouts in the case where delay is present. Furthermore, we plan to exploit the connections with distributed and localized model predictive control (Amo Alonso and Matni, 2019), since communication in this scheme is key and drop of packages is likely in real-life applications. Another application could be the coalitional control framework (Fele et al., 2018), where cooperating local controllers are clustered in disjoint groups with intermittent communication to promote a trade-off between closed-loop performance and coordination overheads.

#### Appendix A. SUPPLEMENTARY PROOFS

**Lemma A.1.** Let  $\Delta$  be a linear bounded operator on  $\mathcal{L}_1$  and assume  $\|\Delta\|_{\mathcal{L}_1} < 1$ , then  $(\mathbf{I} - \Delta)^{-1}$  exists, can be written equivalently as

$$(\mathbf{I} - \Delta)^{-1} = \sum_{k=0}^{\infty} \Delta^k$$

and is bounded by  $\|(\mathbf{I} - \Delta)^{-1}\|_{\mathcal{L}_1} \leq \frac{1}{1-\lambda}$  where  $\Delta^k = \underbrace{\Delta \circ \Delta \circ \dots \circ \Delta}_k$ .

#### Proof of Lemma Lemma 4.1

*Proof.* Decompose  $\mathbf{w}$  into

$$\mathbf{w} = \sum_{i=1}^n \sum_{t=0}^{\infty} \mathbf{s}^{(i,t)} w_{i,t},$$

where  $\mathbf{s}^{(i,t)}$  are dirac sequences, i.e.  $s_{i,t}^{(i,t)} = 1$  and  $s_{j,k}^{(i,t)} = 0$  for all other times  $k$  and vector entries  $j$ . Now, due to linearity and triangle inequality we have

$$\|\Phi \mathbf{w}\|_2 = \left\| \sum_{i=1}^n \sum_{t=0}^{\infty} \Phi \mathbf{s}^{(i,t)} w_{i,t} \right\|_2 \leq \sum_{i=1}^n \sum_{t=0}^{\infty} \|\Phi \mathbf{s}^{(i,t)}\|_2 |w_{i,t}|,$$

where we dropped the  $t$  super-index due to time-invariance of  $\Phi$ . It follows the inequality

$$\begin{aligned} \|\Phi \mathbf{w}\|_2 &\leq \max_i \sqrt{\|[\phi^0]_i\|_F^2} \sum_{i=1}^n \sum_{t=0}^{\infty} |w_{i,t}| \\ &\leq \|\Phi\|_{2 \leftarrow 1} \|\mathbf{w}\|_1. \end{aligned}$$

The result follows, because the left-hand bound can always be achieved with an appropriately chosen dirac sequence  $\mathbf{w}$ .  $\square$

#### REFERENCES

- Amo Alonso, C. and Matni, N. (2019). Distributed and Localized Model Predictive Control via System Level Synthesis. *arXiv:1909.10074 [cs, eess, math]*. ArXiv: 1909.10074.
- Anderson, J., Doyle, J.C., Low, S., and Matni, N. (2019). System Level Synthesis. *arXiv:1904.01634 [cs, math]*. ArXiv: 1904.01634.
- Cetinkaya, A., Ishii, H., and Hayakawa, T. (2015). Event-triggered control over unreliable networks subject to jamming attacks. In *Proc. 54th IEEE Conference on Decision and Control (CDC)*, 4818–4823.
- Fele, F., Debada, E., Maestre, J.M., and Camacho, E.F. (2018). Coalitional Control for Self-Organizing Agents. *IEEE Transactions on Automatic Control*, 63(9), 2883–2897. doi:10.1109/TAC.2018.2792301.
- Ho, D. and Doyle, J.C. (2019). Scalable robust adaptive control from the system level perspective. In *2019 American Control Conference (ACC)*, 3683–3688.
- Ho, D. (2020). A system level approach to discrete-time nonlinear systems. *arXiv preprint arXiv:2004.08004*.
- Lun, Y.Z., D’Innocenzo, A., Smarra, F., Malavolta, I., and Di Benedetto, M.D. (2019). State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software*, 149, 174–216.
- Matni, N., Wang, Y.S., and Anderson, J. (2017). Scalable system level synthesis for virtually localizable systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 3473–3480. IEEE, Melbourne, Australia. doi:10.1109/CDC.2017.8264168.
- Mishra, P.K., Chatterjee, D., and Quevedo, D.E. (2018). Stabilizing stochastic predictive control under Bernoulli dropouts. *IEEE Transactions on Automatic Control*, 63(6), 1579–1590.
- Quevedo, D.E., Mishra, P.K., Findeisen, R., and Chatterjee, D. (2015). A stochastic model predictive controller for systems with unreliable communications. *IFAC-PapersOnLine*, 48(23), 57–64.
- Sandberg, H., Amin, S., and Johansson, K.H. (2015). Cyber-physical security in networked control systems: An introduction to the issue. *IEEE Control Systems Magazine*, 35(1), 20–23.
- Wang, Y.S. (2017). *A System Level Approach to Optimal Controller Design for Large-Scale Distributed Systems*. Ph.D. thesis, California Institute of Technology.
- Wang, Y.S., Matni, N., and Doyle, J.C. (2014). Localized LQR optimal control. In *53rd IEEE Conference on Decision and Control*, 1661–1668. IEEE, Los Angeles, CA, USA. doi: 10.1109/CDC.2014.7039638.
- Wang, Y.S., Matni, N., and Doyle, J.C. (2017). Separable and Localized System Level Synthesis for Large-Scale Systems. *arXiv:1701.05880 [cs, math]*. ArXiv: 1701.05880.