Localized LQR Control with Actuator Regularization

Yuh-Shyang Wang, Nikolai Matni, and John C. Doyle

Abstract— **In previous work, we posed and solved the localized linear quadratic regulator (LLQR) problem – a LLQR controller is one that limits the propagation of dynamics to user-specified subsets of the global system. The advantages of taking this approach are tangible, as we show that this allows the controller to be synthesized and implemented in a scalable local manner. Implicit in this previous work was the existence of a feasible spatio-temporal constraint on the controller and closed loop response of the system that enforced these locality properties. This paper proposes and analyzes a procedure for designing such a spatio-temporal constraint, which can be interpreted as a measure of the implementation complexity of a controller, and a sparse actuation architecture that ensures that it is feasible. We show that the computational tasks involved can be suitably decomposed and solved using the alternating direction method of multipliers (ADMM), thus providing a scalable approach to designing a LLQR controller with a sparse actuation architecture.**

I. Introduction

A fundamental challenge in distributed control system design is balancing the tradeoff between the closed loop performance achieved by a controller, and the complexity required to implement it. The complexity of a distributed controller can be measured in terms of the scale of the synthesis problem, the amount of communication needed between sub-controllers to compute local control actions, and the density of the actuation and sensing architecture of the controller. Exploring this design space is highly non-trivial for large-scale distributed systems such as the smart grid, the Internet, wireless sensor networks, or biological networks. For example, although a centralized optimal controller achieves a globally optimal closed loop, it can be difficult to compute for large-scale systems, and impossible to implement due to communication constraints between sensors, actuators and sub-controllers.

In order to address some of these issues, the field of distributed (decentralized) optimal control has emerged, and allows for realistic communication constraints amongst local sensors, actuators, and sub-controllers to be explicitly incorporated into the design process. It has been shown that the distributed optimal control problem admits a convex reformulation in the Youla domain if [1] and only if [2] the information sharing constraint is quadratically invariant (QI) with respect to the plant. With the identification of quadratic invariance as a means of convexifying the distributed optimal control problem, tractable solutions for various types of distributed constraints and objectives have been developed [3]–[8]. When this property does not hold, approximation methods also exist [9]–[12] that attempt to approximately solve the resulting non-convex controller synthesis problem.

These approaches however have not addressed the issues of controller complexity and scalability. Motivated by this challenge, we introduced the localized optimal control problem in [13]–[16] – the intuition behind this approach is that if the actuation and sensing architectures of a controller are sufficiently dense, and if information can be exchanged between them sufficiently quickly, then the effect of each disturbance can be contained to a localized region. In [14], we pose and solve the localized linear quadratic regulator (LLQR) optimal control problem, which is a state-feedback localized control problem with an $\mathcal{H}_2$ performance metric. For this problem, we show that containing the effects of each disturbance to a local subset of the system allows for the LLQR controller to be implemented and synthesized in a scalable, parallel, and localized manner.

A key step in the localized optimal control framework is the identification of feasible *d-localized subspace constraints*, i.e., determining the local neighborhoods to which each disturbance is confined. This paper addresses this issue and proposes a general procedure for designing the locality constraints imposed on the closed loop system and the distributed controller, as well as the actuation architecture that ensures that these constraints result in a feasible localized optimal control problem. Specifically, we restrict our discussion to the LLQR optimal controller setting (i.e., to the state-feedback setting), and show that the relative communication speed between sub-controllers can be used to identify small feasible localized regions if the system is assumed to have full actuation. We then expand these small localized regions slightly and use the Regularization for Design (RFD) framework [17] to pose an actuator regularized LLQR optimal control problem, which allows for the identification of a sparse actuation architecture that nonetheless ensures that the expanded localized constraints remain feasible.

We emphasize that although the form of the alternating direction method of multipliers (ADMM) algorithms presented in this paper and our companion paper [16] are similar in form, these two papers address fundamentally different problems. Whereas [16] shows that the *output-feedback* localized optimal control problem can be solved

The authors are with the department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA ({yswang,nmatni,doyle}@caltech.edu).

This research was in part supported by NSF NetSE, AFOSR, the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office, and from MURIs "Scalable, Data-Driven, and Provably-Correct Analysis of Networks" (ONR) and "Tools for the Analysis and Design of Complex Multi-Scale Networks" (ARO). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

in a scalable way given a feasible localized constraint set, this paper addresses *how to design* such constraint sets and the controller architecture needed to ensure their feasibility.

The rest of this paper is structured as follows. Section II introduces the interconnected system model and recalls the LLQR optimal control problem. In Section III, we propose a sequential procedure for the design of the locality constraints of the control problem and a sparse actuation architecture that ensures that it is feasible. We then formulate the actuator regularized LLQR problem in Section IV, and the ADMM algorithm [18] can be used to decompose and solve this problem in a distributed, scalable and localized manner. Finally, we illustrate the effectiveness of our method with a numerical example in Section V, and end with conclusions and directions for future work in Section VI.

Notation: We use lower and upper case Latin letters such as x and M to denote vectors and matrices, respectively, and lower and upper case boldface Latin letters such as $\mathbf{x}$ and $\mathbf{M}$ to denote signals and transfer matrices, respectively. We use calligraphic letters such as $\mathcal{S}$ to denote subspaces. We use $\mathcal{RH}_\infty$ to denote the space of stable, proper real-rational transfer matrices. For a transfer matrix $\mathbf{G}$, we use $G[i]$ to denote its ith spectral component. Finally, we use $\mathcal{F}_T$ to denote the space of finite impulse response transfer matrices with horizon T, i.e., $\mathcal{F}_T := \{\mathbf{G} \in \mathcal{RH}_\infty \,|\, \mathbf{G} = \sum_{i=0}^{T} \frac{1}{z^i} G[i]\}$.

II. Preliminaries and Problem Statement

A. Interconnected System Model

We consider distributed systems described by linear time-invariant (LTI) dynamics and composed of a collection of subsystems that interact with each other according to a network topology specified by the interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V} = \{1, \dots, n\}$ denotes the set of subsystems: to each subsystem i we associate a state vector x_i and control action u_i. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes the interaction between these subsystems: an edge (i, j) is in $\mathcal{E}$ if and only if the state x_j of subsystem j directly affects the state x_i of subsystem i. Defining the (incoming) neighbor set $\mathcal{N}_i$ of subsystem i to be $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$, may then write the dynamics of subsystem i as

$$x_i[k+1] = A_{ii} x_i[k] + B_{ii} u_i[k] + \sum_{j \in \mathcal{N}_i} A_{ij} x_j[k] + w_i[k] \quad (1)$$

where A_{ii}, A_{ij}, B_{ii} are constant matrices with compatible dimensions and w_i is a disturbance affecting subsystem i. The subsystem dynamics (1) can be used to construct the global plant

$$x[k+1] = Ax[k] + B_2 u[k] + w[k] \quad (2)$$

where x, u, and w are stacked vectors of subsystem states, control actions, and disturbances, respectively. The state space matrices (A, B_2) are chosen such that the dynamics (2) are compatible with those described in (1).

Example 1: Consider a chain of linear systems as shown in Fig. 1. The state x_i of each subsystem is assumed to be a scalar, and each subsystem is assumed to have an actuator. In this case, the matrix A in (2) is tridiagonal and $B_2 = I$.

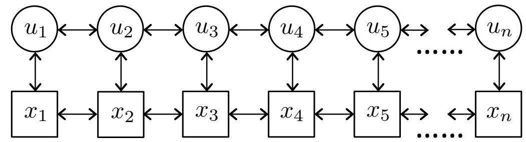


Fig. 1. Chain example

In the following, we assume that the disturbances w_i are drawn i.i.d. from a zero mean unit covariance Gaussian distribution – in particular the disturbances are assumed to not be correlated with each other. The control objective is to minimize the $\mathcal{H}_2$ norm of the regulated output $C_1\mathbf{x} + D_{12}\mathbf{u}$ for some user-specified matrices (C_1, D_{12}).

B. Preliminaries on LLQR

Distributed systems, even those as simple as the chain model of Example 1, can make the optimal controller synthesis task very challenging: (i) in general, the optimal control problem may be intractable if there is not sufficient communication between sub-controllers [19], and (ii) even if the problem is tractable, the resulting synthesis problem may not scale gracefully to large systems. In [14], we introduce the LLQR control scheme to allow distributed optimal controllers to be synthesized and implemented in a scalable manner.

For a system described by (2), we let $\mathbf{R}$ and $\mathbf{M}$ denote the closed loop transfer matrices mapping the disturbance $\mathbf{w}$ to the state $\mathbf{x}$ and control action $\mathbf{u}$, i.e. $\mathbf{x} = \mathbf{R}\mathbf{w}$ and $\mathbf{u} = \mathbf{M}\mathbf{w}$. In [14], we show that there exists a controller such that $\mathbf{x} = \mathbf{R}\mathbf{w}$ and $\mathbf{u} = \mathbf{M}\mathbf{w}$ if and only if

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I, \quad (3)$$

and use this observation to pose the LLQR optimal control problem as:

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} ||_{\mathcal{H}_2}^2 && (4a) \\ \text{subject to} \quad & \text{constraint (3)} && (4b) \\ & \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T && (4c) \end{aligned}$$

where $\mathcal{C}$ is a subspace encoding the information sharing constraints of the distributed controller, $\mathcal{L}$ is a localized (sparsity) constraint and $\mathcal{F}_T$ restricts the optimization variables to have finite impulse responses of horizon T. Without the last constraint (4c), problem (4a) - (4b) reduces to a traditional LQR problem with uncorrelated disturbances. Whereas the first subspace $\mathcal{C}$ is defined by the communication network interconnecting the sub-controllers, the subspaces $\mathcal{L}$ and $\mathcal{F}_T$ are design parameters to be specified by the control designer. Before we discuss the role of these latter two subspaces, we recall the following theorem from [14].

Theorem 1: Suppose that there exists a pair of transfer matrices $(\mathbf{R}, \mathbf{M})$ satisfying constraints (4b) and (4c). Then a controller yielding the desired closed loop response $\mathbf{x} = \mathbf{R}\mathbf{w}$ and $\mathbf{u} = \mathbf{M}\mathbf{w}$ can be implemented by

$$\begin{aligned} \hat{w}[k] &= x[k] - \hat{x}[k] \\ u[k] &= \sum_{\tau=0}^{T-1} M[\tau+1]\hat{w}[k-\tau] \\ \hat{x}[k+1] &= \sum_{\tau=0}^{T-2} R[\tau+2]\hat{w}[k-\tau]. \end{aligned} \tag{5}$$

In (5), the $\hat{w}$ term should be interpreted as an estimate of the disturbance, and is computed by taking the difference between the state measurement x and the reference trajectory $\hat{x}$. Similarly, the $\hat{x}$ term should be viewed as a reference trajectory, and is computed by the estimated disturbance $\hat{w}$.

Theorem 1 thus shows how the sparsity of the closed loop transfer matrices $(\mathbf{R}, \mathbf{M})$ translates into the implementation complexity of a controller as in (5). If the rows of these transfer matrices are suitably sparse, then each sub-controller only needs to collect a small number of subsystem estimated disturbances $\hat{w}_i$ to compute its control law and reference trajectory. The localized constraint $\mathcal{L}$ is the mechanism that we use to impose this sparsity.

We begin by defining the notion of the d-outgoing and incoming sets at a subsystem j. To do so, we let the distance $\mathrm{dist}\,(j \to i)$ from subsystem j to subsystem i be given by the length of the shortest path from node j to node i in the graph $\mathcal{G}$. We say that a set $\mathscr{C}$ of subsystems is of size d if $\mathrm{dist}\,(i \to j) \leq d$ for all $i, j \in \mathscr{C}$. We then define the d-outgoing set of subsystem j as $\mathrm{Out}_j(d) := \{i | \mathrm{dist}(j \to i) \leq d\}$, and the d-incoming set of subsystem j as $\mathrm{In}_j(d) := \{i | \mathrm{dist}(i \to j) \leq d\}$ – by definition, both of these sets are of size d. Our approach to making the controller synthesis task specified in optimization problem (4) scalable is to confine, or *localize* the effects of the state disturbance to a d-outgoing set at each subsystem j, for a size d much smaller than that of $\mathcal{V}$. As we make precise in the sequel, this implies that each sub-controller j only needs to collect data from subsystems i contained in its d-incoming set $\mathrm{In}_j(d)$.

The benefit of the formulation (4) is that it allows for the implementation complexity of the controller (5) to be specified via the subspace constraints $\mathcal{L}$ and $\mathcal{F}_T$ in (4c). In particular, the FIR subspace constraint $\mathcal{F}_T$ implies that the controller (5) can be implemented using FIR filter banks of length T. To illustrate the role of the localized constraint $\mathcal{L}$, we begin first with the following illustrative example.

Example 2: Consider the chain example in Fig. 1, and suppose that we want to impose that each sub-controller i is only allowed to collect information from its direct incoming neighbors (i.e., from subsystems i in $\mathrm{In}_i(1)$) to generate its control action u_i and reference trajectory $\hat{x}_i$. In order to do so, it suffices to construct a 1-localized subspace constraint $\mathcal{L}$ in (4c) such that $\mathbf{R}$ and $\mathbf{M}$ have a tridiagonal sparsity pattern. Likewise, if we allow each subsystem i to collect information from subsystems in $\mathrm{In}_i(2)$, then its suffices to construct a 2-localized constraint $\mathcal{L}$ such that $\mathbf{R}$ and $\mathbf{M}$ are pentadiagonal. Here one such admissible solution is given by $\mathbf{R} = \frac{1}{z}I + \frac{1}{z^2}A$ and $\mathbf{M} = -\frac{1}{z^2}A^2$.

Another benefit of the localized and FIR subspace constraints $\mathcal{L}$ and $\mathcal{F}_T$ is that they facilitate the synthesis of a distributed controller. In particular, the FIR constraint $\mathcal{F}_T$ makes optimization problem (4) finite dimensional, and choosing the constraint $\mathcal{L}$ to be a d-localized constraint allows for the LLQR optimal control problem (4) to be decomposed into subproblems of size specified by d. As we explain in detail in [14], optimization problem (4) admits a column-wise decomposition. Specifically, as we assume that the local disturbances w_i are uncorrelated and that the system is linear time invariant, we can analyze the closed loop response for each local disturbance w_i in an independent and parallel way, i.e., we can solve for the columns $\mathbf{R}_i$ and $\mathbf{M}_i$ of the closed loop response transfer matrices $\mathbf{R}$ and $\mathbf{M}$ in parallel. Further, when the subspace $\mathcal{L}$ is chosen to be a d-localized constraint, then the effect of each disturbance w_i is limited to the region $\mathrm{Out}_i(d)$, and hence a block row j of $\mathbf{R}_i$ and $\mathbf{M}_i$ is nonzero only if $j \in \mathrm{Out}_i(d)$, thus reducing the dimension of each optimization subproblem from that of the entire system to one specified by d. We defer the details of this decomposition and dimensionality reduction procedure to our companion paper [16], and instead consider here a simple illustrative example.

Example 3: Consider again the chain example, and assume that we have chosen the subspace $\mathcal{L}$ to be a 1-localized constraint on $\mathbf{R}$ and a 2-localized constraint on $\mathbf{M}$, i.e., we impose a tridiagonal sparsity constraint on $\mathbf{R}$ and a pentadiagonal sparsity constraint on $\mathbf{M}$. As the system is LTI and the noise is assumed to be uncorrelated, we can analyze the closed loop transfer matrices $\mathbf{R}_i$ and $\mathbf{M}_i$ for each w_i in an independent and parallel manner. Consider w_1 in Fig. 2: here the localized constraint $\mathcal{L}$ constrains the effects of w_1. In particular, only the states of subsystems 1 and 2 can be perturbed by disturbance w_1, i.e., the effect of w_1 is contained within $\mathrm{Out}_1(1)$. Notice that because of the sparsity constraint imposed by the 1-localized subspace $\mathcal{L}$, the effect of disturbance w_1 on the state of subsystem 3 is forced to be zero. Intuitively, this "boundary" state acts as a barrier preventing the disturbance from propagating further into the system. It is this containment of the effect of the disturbance w_1 to $\mathrm{Out}_1(1)$ that allows us to optimize the closed loop response $(\mathbf{R}_1, \mathbf{M}_1)$ to disturbance w_1 over this localized area, independent of the size of the entire system.

C. Problem Statement

Thus we have argued that the localized subspace constraint $\mathcal{L}$ in optimization problem (4) plays an important role in the LLQR control scheme – if $\mathcal{L}$ is d-localized, then the synthesis and implementation of an LLQR optimal controller can be done in a scalable way. However, in our previous work we do not provide a method for finding a small d such that optimization problem (4) is feasible for a d-localized subspace constraint $\mathcal{L}$. In this paper, we show that the existence of a feasible and sparse subspace constraint $\mathcal{L}$ is a function of the given communication constraints $\mathcal{C}$ and the

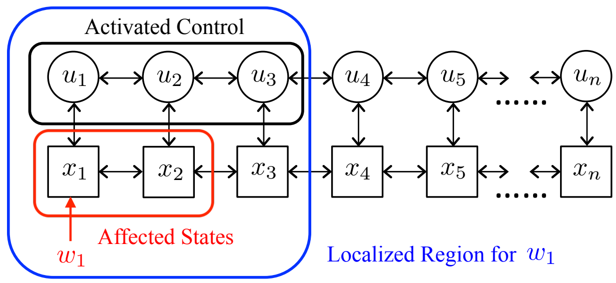


Fig. 2. Localized Region

actuator density of the system, as specified by the matrix B_2. While the communication constraint $\mathcal{C}$ is given in priori, in this paper we assume that the actuation architecture (i.e., the matrix B_2) of the distributed system (2) can be *designed*. The contribution of this paper is twofold. First, given an initial actuator placement $B_2 = I$ and a communication constraint $\mathcal{C}$ specified in terms of information sharing delays between sub-controllers, we derive a simple rule for designing a localized subspace $\mathcal{L}$ and a FIR subspace $\mathcal{F}_T$ such that (4) is feasible. Then, given the localized constraint $\mathcal{L}$ and the FIR constraint $\mathcal{F}_T$, we use the regularization for design [17] to explore the tradeoff between actuation density and closed loop performance. We further show that the actuator regularized LLQR optimal control problem can be solved in a localized and hence scalable manner. Thus these two main results provide a means for designing a localized optimal controller with sparse actuation that nonetheless achieves a desirable closed loop performance.

III. LLQR Constraint Setup

In this section, we propose a general procedure to design the actuation architecture B_2, d-localized subspace $\mathcal{L}$, and the FIR subspace $\mathcal{F}_T$ such that the LLQR synthesis problem (4) is feasible. We then focus our discussion on the subtleties that arise in designing the subspaces $\mathcal{L}$ and $\mathcal{F}_T$.

A. General Design Procedure

We begin by proposing a serial procedure that sequentially designs the localized subspace $\mathcal{L}$, the FIR subspace $\mathcal{F}_T$, and the actuation scheme B_2.

Initialization: Given a distributed plant (2), assume a completely dense actuation architecture, i.e., $B_2 = I$.

Subspaces $\mathcal{L}$ and $\mathcal{F}_T$: Given the communication subspace constraint $\mathcal{C}$ imposed on the distributed controller, determine the sparsest localized constraint $\mathcal{L}$, i.e., the smallest d-localized constraint $\mathcal{L}$, and the minimal length T of $\mathcal{F}_T$ such that the LLQR synthesis problem (4) is feasible. This task can be accomplished by beginning with d and T set to 1 and incrementing these values until feasibility is achieved.

Actuator Regularization: Given the LLQR synthesis problem (4) constrained by the designed subspaces $\mathcal{L}$ and $\mathcal{F}_T$, use regularization for design [17] to explore the tradeoff between closed loop performance and actuation density. If no acceptably sparse actuation architecture can be found, return to the previous step and increase either d or T.

This design procedure is certainly not unique, but is simple and intuitive: we assume that the system has full actuation and determine the "simplest" constraints such that a localized controller can still be synthesized. Then we attempt to remove actuators such that locality and closed loop performance are preserved: if this latter step cannot be satisfactorily completed, increase the complexity of the localized controller (by increasing either d or T) and repeat.

The rest of this section focusses on the subtleties of designing the subspaces $\mathcal{L}$ and $\mathcal{F}_T$, and we defer the actuator regularization task to §IV. Once the general design procedure is complete, the methods of [14] can be used to synthesize a LLQR controller that uses the designed actuation architecture and that respects the locality and FIR subspace constraints $\mathcal{L}$ and $\mathcal{F}_T$.

B. Designing $\mathcal{L}$ and $\mathcal{F}_T$

Use $\mathcal{L}_j$ to denote the jth (block) column of the yet to be designed localized constraint $\mathcal{L}$. It follows from the discussion of §II that $\mathcal{L}_j \cap \mathcal{F}_T$ is the spatio-temporal constraint imposed on the closed loop response from disturbance $\mathbf{w}_j$ to the global state and control action $\mathbf{x}$ and $\mathbf{u}$, respectively. It follows that the (block) row-wise sparsity pattern of $\mathcal{L}_j$ encodes which subsystems are allowed to be perturbed by disturbance $\mathbf{w}_j$ in closed loop, i.e., for a d-localized constraint $\mathcal{L}$, the ith (block) row of $\mathcal{L}_j$ is nonzero only if $i \in \text{Out}_j(d)$. The size d of the subset of subsystems that can be perturbed by a disturbance $\mathbf{w}_j$ is primarily determined by the communication delay constraints $\mathcal{C}$ that are imposed on the controller – if information can be exchanged between sub-controllers very quickly, then they can coordinate their actions to contain the effect of a disturbance in a small d-outgoing set. Conversely, if information is exchanged slowly then it may not be possible to localize a disturbance at all: as an extreme case, if communication between sub-controllers is slower than the propagation of disturbances through the plant, then there is no way to localize the effect of a local disturbance.

It should be clear that the feasibility of a d-localized constraint $\mathcal{L}$ depends on both the topology of the interaction graph $\mathcal{G}$ underlying the distributed system (2) and the information exchange constraint set $\mathcal{C}$ underlying the distributed controller. We now consider the case of $B_2 = I$ (dense actuation) as it allows us to treat each scalar state as a subsystem, and assume that the communication subspace $\mathcal{C}$ imposes delay constraints on the sub-controllers as follows: each sub-controller has a sensing delay of t_s, a communication delay of t_c (i.e., it takes time t_c for a sub-controller to transmit information to its neighbor), and actuation delay t_a. The delays (t_s, t_c, t_a) are normalized with respect to the sampling time of the discrete time system (2), and hence they may be non-integers in general. We adopt the following convention to handle fractional delays: if information is received by a sub-controller between two sampling times t and $t+1$, then it may be used by the sub-controller to compute its control action at time $t+1$. We can characterize a relationship between the delay parameters and the minimal size d such that a d-localized constraint set $\mathcal{L}$ leads to a feasible LLQR problem (4).

Lemma 1: Assume that the communication network topology mimics that of the interaction graph $\mathcal{G}$ underlying the plant (2), but allows for communication between sub-controller faster than disturbances propagate through the plant, i.e. $t_c < 1$. Then a system (2) with $B_2 = I$ is d-localizable if $\frac{t_s+t_a}{1-t_c} - 1 \leq d$.

Proof: Consider the disturbance $\mathbf{w}_i$ that affects subsystem i. If the LLQR problem (4) is feasible for $\mathcal{L}$ a d-localized constraint, only the states in the d-outgoing set $\text{Out}_i(d)$ may be perturbed in closed loop by disturbance $\mathbf{w}_i$. As each state can be directly actuated, we only need to ensure that the state of the "boundary" subsystems k satisfying $\text{dist}\,(i \rightarrow k) = d+1$ are not affected by disturbance $\mathbf{w}_i$. By definition, it takes the disturbance $(d+1)$ time steps to affect these "boundary" states via the dynamics (2). As the communication network topology mimics the topology of the plant, measurements of the disturbance $w_i[t]$ taken by subsystem i is transmitted to these boundary subsystems with a delay of $t_s+(d+1)t_c$. Thus if $t_s+(d+1)t_c+t_a \leq d+1$, then the boundary subsystems are given advanced warning of the disturbance that is propagating towards them: as actuation is assumed to be dense, the corresponding boundary sub-controllers can suppress disturbance such that the states of the boundary subsystems are not perturbed. As the initial disturbance was arbitrary, this shows that the delay condition of the lemma is sufficient to ensure the feasibility of a d-localized constraint $\mathcal{L}$. ■

Remark 1: This condition is reminiscent of the delay characterization of QI developed in [20]. Using our notation and setup, the delays (t_s, t_c, t_a) define a QI subspace constraint if $p(t_c) \leq p + t_a + t_s$, where p is the distance between *any* pair of states. Thus we see that the delay condition stated in Lemma 1 is more restrictive than the QI condition: for example, if $t_c = 1$, then the delays define a QI subspace constraint, but the system is not d-localizable for any d smaller than the diameter of the interaction graph $\mathcal{G}$. However, because the LLQR controller is by definition d-localized, communication between subsystems is limited to within a subset of size d if $t_c < 1$ – in contrast, the QI controller requires that local information be shared globally if the plant (2) has a strongly connected topology.

We further specialize the delay condition of Lemma 1 by assuming that $t_s = 0$ and $t_a = 1$, and let $h = \frac{1}{t_c}$ be the relative communication speed between neighboring controllers. The condition of Lemma 1 then reduces to $\frac{1}{h-1} \leq d$ – as we assume full actuation, every d-localized subset of the system is trivially controllable, and thus we can set the FIR horizon to be $T = d+1$.

Thus using Lemma 1 and the previous discussion, given a communication speed h, we can identify small values d and T to ensure that the subspace constraint $\mathcal{S} := \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T$ leads to a feasible LLQR problem (4). We now give an explicit construction for $\mathcal{S}$. Let $\text{sp}\,(\cdot) : \mathbb{R}^{m\times n} \rightarrow \{0,1\}^{m\times n}$ be the support operator, where $\{\text{sp}\,(A)\}_{ij} = 1$ if $A_{ij} \neq 0$ and $\{\text{sp}\,(A)\}_{ij} = 0$ otherwise. Let $\bar{\mathcal{A}} = \text{sp}\,(A) \cup \text{sp}\,(I)$, where $\cup$ is the OR operator on binary matrices. Using the parameter (d, T, h), a d-localized subspace $\mathcal{S}$ with horizon T for the closed loop transfer matrices $(\mathbf{R}, \mathbf{M})$ is specified by

$$\begin{aligned} \mathcal{S}_R &= \sum_{i=1}^{T} \frac{1}{z^i} \bar{\mathcal{A}}^{\min(d, \lfloor h(i-1) \rfloor)} \\ \mathcal{S}_M &= \sum_{i=1}^{T} \frac{1}{z^i} \bar{\mathcal{A}}^{\min(d+1, \lfloor h(i-1) \rfloor)}. \end{aligned} \tag{6}$$

According to Lemma 1, the LLQR problem (4) is feasible for a fully actuated system if $h > 1$, $d \geq \frac{1}{h-1}$, and $T \geq d+1$.

Usually, the communication speed h is imposed by the pre-specified communication constraint $\mathcal{C}$. In contrast, the parameters T and d can be chosen by the controller designer. The parameter T dominates the trade-off between settling time and transient performance (including $\mathcal{H}_2$ norm or maximum overshoot), and it also affects FIR horizon of the LLQR controller implementation (5) – thus T should chosen as small as possible while still leading to acceptable transient performance. The parameter d should also be kept as small as possible as it controls the complexity of the controller synthesis and implementation procedures. However, if sparse actuation is desired, then it is desirable to increase d from its minimum to allow some flexibility in the actuation architecture. Although these parameters typically need to be identified by trial and error, they are integer quantities that can be explored fairly easily. For instance, we later show that the locality size d often only needs to be increased by one or two to be able to design sparse actuation schemes that nonetheless achieve good closed loop performance.

IV. Actuator Regularization

In this section we show how to explore the tradeoff between closed loop performance and actuation density using the regularization for design [17] framework, given a feasible spatio-temporal constraint $\mathcal{S} = \mathcal{C} \cap o_{j,d} \cap \mathcal{F}_T$.

A. Problem Formulation

Recall that the LLQR controller is implemented using the equations (5) – it follows that if the ith row of the map $\mathbf{M}$ is zero, i.e., if $e_i^\top \mathbf{M} = 0$, then the control action u_i is identically zero, and the corresponding actuator at subsystem i can be discarded. If we want to construct a controller that uses at most r actuators, we can impose an additional row-cardinality constraint $N_r(\mathbf{M}) \leq r$ on the transfer matrix $\mathbf{M}$. If we incorporate this constraint into the LLQR problem (4), we can formulate the actuator constrained LLQR control problem as

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & \left\| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \right\|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & (3) \text{ and } (4c) \\ & N_r(\mathbf{M}) \leq r \end{aligned} \tag{7}$$

where $N_r(\mathbf{M})$ represents the number of nonzero rows of $\mathbf{M}$. Problem (7) is a combinatorial optimization problem due to the last constraint, and is generally computationally hard to solve. The regularization for design framework [17] allows such combinatorial controller architecture design problems to be solved in a tractable way based on convex relaxations of the combinatorial penalties. We use this technique to relax (7) and formulate the LLQR problem with actuator regularization as

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & \|\begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix}\|_{\mathcal{H}_2}^2 + \|\mathbf{M}\|_{\mathcal{A}} && \text{(8a)} \\ \text{subject to} \quad & \text{(3) and (4c)} && \text{(8b)} \end{aligned}$$

where $||.||_{\mathcal{A}}$ is the actuator norm introduced in [17]. In particular, the actuator norm is given by

$$||\mathbf{M}||_{\mathcal{A}} = \sum_{i=1}^{n_u} \lambda_i ||e_i^\top \mathbf{M}||_{\mathcal{H}_2}, \tag{9}$$

where λ_i is the relative price of each actuator. When $\lambda_i = 1$ for all i, (9) is equivalent to the ℓ_1/ℓ_2 norm (group lasso [21] in the statistical learning literature). The parameters $\{\lambda_i\}$ can be used to explore the tradeoff between closed loop performance (the square of $\mathcal{H}_2$ norm in (8a)) and the actuation architecture complexity (as measured by the actuator norm in (8a)), given a fixed controller spatio-temporal complexity (as specified by $\mathcal{L}$ and $\mathcal{F}_T$). Further, once an actuation architecture has been identified, a traditional LLQR optimization problem (4) can then be solved restricted to the designed actuation architecture, thus removing any bias or conservatism introduced by the actuator norm regularizer (cf. [17] for why this two step approach is appropriate).

Equation (8) is a convex optimization problem, and hence is tractable to solve – in the next subsection we show how it can be solved in a scalable way by exploiting the decomposability of the standard LLQR problem (4) and the ADMM optimization algorithm.

B. ADMM Algorithm

As shown in [14], the LLQR optimization problem (4) admits a (block) column-wise decomposition, resulting in a set of small subproblems that can be solved in parallel. In the actuator regularized LLQR problem (8), the actuator norm penalty decomposes in a (block) row-wise fashion, thus introducing a coupling term. A standard approach in the statistical inference and optimization literature to circumvent such coupling introduced by a regularizer is to use the ADMM algorithm, which introduces a redundant variable that shifts the coupling constraint to an easy to enforce equality constraint.

We begin by defining the extended-real-value functions $f(\mathbf{R}, \mathbf{M}_1)$, $g(\mathbf{M}_2)$ as

$$f(\mathbf{R}, \mathbf{M}_1) = \begin{cases} \text{(4a)} & \text{if (3), (4c)} \\ \infty & \text{otherwise} \end{cases}$$

$$g(\mathbf{M}_2) = \begin{cases} ||\mathbf{M}_2||_{\mathcal{A}} & \text{if (4c)} \\ \infty & \text{otherwise} \end{cases}$$

Problem (8) can then be equivalently formulated as

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}_1,\mathbf{M}_2\}}{\text{minimize}} \quad & f(\mathbf{R}, \mathbf{M}_1) + g(\mathbf{M}_2) \\ \text{subject to} \quad & \mathbf{M}_1 = \mathbf{M}_2. \end{aligned} \tag{10}$$

Problem (10) is precisely of the form needed by the ADMM algorithm [18], which is specified by the following iterate update equations

$$\begin{aligned} (\mathbf{R}^{k+1}, \mathbf{M}_1^{k+1}) = \underset{\mathbf{R},\mathbf{M}_1}{\text{argmin}} \Big(& f(\mathbf{R}, \mathbf{M}_1) \\ & + \frac{\rho}{2} ||\mathbf{M}_1 - \mathbf{M}_2^k + \mathbf{\Lambda}^k||_{\mathcal{H}_2}^2 \Big) \end{aligned} \tag{11a}$$

$$\mathbf{M}_2^{k+1} = \underset{\mathbf{M}_2}{\text{argmin}} \Big(g(\mathbf{M}_2) + \frac{\rho}{2} ||\mathbf{M}_2 - \mathbf{M}_1^{k+1} - \mathbf{\Lambda}^k||_{\mathcal{H}_2}^2 \Big) \tag{11b}$$

$$\mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k + \mathbf{M}_1^{k+1} - \mathbf{M}_2^{k+1}. \tag{11c}$$

Each of these subproblems is finite dimensional and can be solved by associating the FIR transfer matrices $\mathbf{R}$, $\mathbf{M}$ and $\mathbf{\Lambda}$ with their matrix representations. It is easy to see that subproblem (11a) has the same form as the LLQR problem (4), save for the additional $\frac{\rho}{2}||\mathbf{M}_1 - \mathbf{M}_2^k + \mathbf{\Lambda}^k||_{\mathcal{H}_2}^2$ term in the objective – fortunately, since this term is also an $\mathcal{H}_2$ penalty, it decomposes in a (block) column-wise fashion, allowing the LLQR decomposition to be applied. As described in [14] and our companion paper [16], the d-localized subspace constraint $\mathcal{L}$ allows the dimension of each (block) column subproblem of optimization problem (11a) to be reduced, as only those subsystems in the $(d+1)$-outgoing set $\text{Out}_j(d+1)$ of a subsystem j need to be considered when solving the jth (block) column subproblem.

Subproblem (11b) can similarly be shown to admit a (block) row-wise decomposition, as the actuator norm (9) and the quadratic penalty $\frac{\rho}{2}||\mathbf{M}_2 - \mathbf{M}_1^{k+1} - \mathbf{\Lambda}^k||_{\mathcal{H}_2}^2$ both decompose in a (block) row-wise fashion. Once again, the d-localized subspace constraint $\mathcal{L}$ allows for the dimensionality of each (block) row subproblem to be reduced. Finally, the lagrange multiplier update equation (11c) can be computed locally because matrix addition can be performed element-wise. Thus if the parameter ρ is agreed upon by all subsystems, the ADMM algorithm allows for the actuator regularized LLQR problem (8) to be solved in a distributed and scalable way.

We now focus on the iterate update problem (11a) and (11b), and show that they both admit analytic solutions. Specifically, the solution to (11a) can be written as an affine function of the problem data $(\mathbf{M}_2^k, \mathbf{\Lambda}^k)$. The solution of (11b) is shown to be given by a vectorial soft-thresholding [22] of the problem data $(\mathbf{M}_1^{k+1}, \mathbf{\Lambda}^k)$. It follows that once the parameters specifying the affine map of the update rule for (11a) are solved for, the iterates specified by (11a) - (11c) can all be computed using closed form solutions, thus offering a fast, distributed and scalable algorithm for solving the actuator regularized LLQR problem.

Analytic Solutions to Iterate Update Problems: We begin by recalling that subproblem (11a) is identical to the LLQR problem (4), save for the added quadratic penalty term on the gap between $\mathbf{M}_1$ and $\mathbf{M}_2^k - \mathbf{\Lambda}^k$. We can then perform a (block) column-wise LLQR decomposition on subproblem (11a) and use the method in [14], [16] to reduce the dimension of the problem. Let $\mathbf{R}_j$ and $\mathbf{M}_j$ be the jth block column of $\mathbf{R}$ and $\mathbf{M}$, respectively. For a d-localized constraint $\mathcal{L}$, let $\mathbf{R}_{o_{j,d}}$ and $\mathbf{M}_{o_{j,d}}$ be the restriction of the maps $\mathbf{R}_j$ and $\mathbf{M}_j$ to their block-rows i satisfying $i \in \text{Out}_j(d+1)$,[1] respectively. We can then define the local plant model $(A_{o_{j,d}}, B_{2o_{j,d}})$ by selecting the sub-matrices of (A, B_2) corresponding to the block-columns and block-rows specified by $\text{Out}_j(d+1)$, i.e., we only need to consider the state-space parameters of the subsystems contained within $\text{Out}_j(d+1)$, the $(d+1)$-outgoing region of subsystem j. Each iterate update (11a) can then be written in terms of these dimensionality reduced variables and constraints. The resulting problem is a least-squares problem subject to affine constraints: consequently, the optimal solution is specified as an affine function of the problem data $(\mathbf{M}_2^k, \mathbf{\Lambda}^k)_{o_{j,d}}$, and can be written

$$\begin{bmatrix} \mathbf{R}^{k+1} \\ \mathbf{M}_1^{k+1} \end{bmatrix}_{o_{j,d}} = F^a_{o_{j,d}}(\mathbf{M}_2^k, \mathbf{\Lambda}^k)_{o_{j,d}} + F^b_{o_{j,d}}, \qquad (12)$$

for a suitable linear map $F^a_{o_{j,d}}$ and affine term $F^b_{o_{j,d}}$ which can be computed using standard methods by exploiting the fact that $\|G\|_F^2 = \|\text{vec}(G)\|_2^2$. Once these parameters $(F^a_{o_{j,d}}, F^b_{o_{j,d}})$ are computed, we can use (12) to update the iterates of (11a), which accelerates the algorithm significantly.

For iterate update problem (11b), we can perform a (block) row-wise decomposition of the objective function and constraint. Let $\mathbf{M}_{\text{in}_{j,d}}$ be the sub-matrix of $\mathbf{M}$ by selecting the jth block row and ith block columns satisfying $i \in \text{In}_j(d+1)$. Using a similar argument as above, each block-row subproblem of update equation (11b) reduces to an unconstrained optimization problem of the form

$$\begin{aligned} \underset{(\mathbf{M}_2)_{\text{in}_{j,d}}}{\text{minimize}} \quad & \lambda_j ||(\mathbf{M}_2)_{\text{in}_{j,d}}||_{\mathcal{H}_2} \\ & + \frac{\rho}{2} ||(\mathbf{M}_2)_{\text{in}_{j,d}} - (\mathbf{M}_1^{k+1} + \mathbf{\Lambda}^k)_{\text{in}_{j,d}}||^2_{\mathcal{H}_2}. \qquad (13) \end{aligned}$$

The analytic solution to (13) is given by the vectorial soft-thresholding operator [22]:

$$\begin{aligned} \text{vec}(\mathbf{M}_2)^{k+1}_{\text{in}_{j,d}} = (1 - \frac{\lambda_j / \rho}{||(\mathbf{M}_1^{k+1} + \mathbf{\Lambda}^k)_{\text{in}_{j,d}}||_{\mathcal{H}_2}})_+ \cdot \\ \text{vec}(\mathbf{M}_1^{k+1} + \mathbf{\Lambda}^k)_{\text{in}_{j,d}} \qquad (14) \end{aligned}$$

in which $(c)_+ = \max(0, c)$ for any scalar c.

Thus using these analytic expressions to implement the iterate updates (11a) and (11b), the actuator regularized LLQR optimization problem (8) can be solved nearly as quickly as the state-feedback problem, as the most expensive step is to compute the update equation parameters $(F^a_{o_{j,d}}, F^b_{o_{j,d}})$ as defined in (12) – once this is done, each update can be implemented via a matrix multiplication or a soft-thresholding.

[1] If we only considered subsystems in the d-outgoing set, the localized constraint $\mathcal{L}$ would have no effect on the synthesized controllers, thus we extend the region by 1 to incorporate "boundary" subsystems that must have a zero response to the disturbance at subsystem j.

Convergence and Stopping Criteria: Assume that $D_{12}^\top D_{12} > 0$. Then the objective function of (8) is strongly convex with respect to $\mathbf{M}$, and the optimal solution $\mathbf{M}^*$ is unique. As f and g are closed, proper, and convex, we have strong duality and (8) satisfies the convergence condition in [18]. From [18], the objective function in (8) converges to its optimal value. As the objective function in (8) is a continuous function of $\mathbf{M}$ and the optimal solution $\mathbf{M}^*$ is unique, we then have primal variable convergence $\mathbf{M}_1^k \to M^*$ and $\mathbf{M}_2^k \to M^*$. For the convergence of $\mathbf{R}$, consider the equation

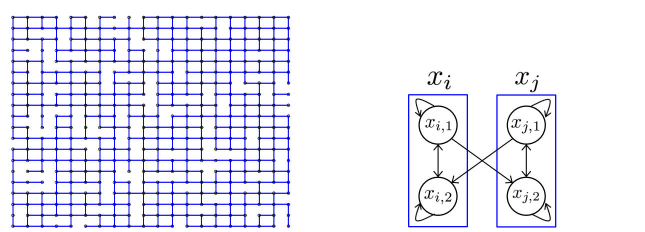
$$(zI - A)(\mathbf{R}^k - \mathbf{R}^*) = B_2(\mathbf{M}^k - \mathbf{M}^*).$$

As the right-hand-side converges to zero for large k, the left-hand-side also converges to zero. Let $\mathbf{\Delta}^k = \mathbf{R}^k - \mathbf{R}^*$: we then have that $\Delta^k[1] \to 0$ and $\Delta^k[i+1] - A\Delta^k[i] \to 0$ for all i. In addition, the constraint in (11a) ensures that $\mathbf{\Delta}^k$ is stable, i.e., $\Delta^k[i] \to 0$ for large i for all k. Combining these two facts, we can show that each spectral component of $\mathbf{\Delta}^k$ converges to 0 for large k using induction. This implies $\mathbf{R}^k \to \mathbf{R}^*$.

We refer the reader to [18] for how to design the stopping criteria for the optimization algorithm (11). We note that we suggest the use of $\|\mathbf{M}_1^k - \mathbf{M}_2^k\|_{\mathcal{H}_2}$ as a measure of primal infeasibility and $||\mathbf{M}_2^k - \mathbf{M}_2^{k-1}||_{\mathcal{H}_2}$ as measure of dual infeasibility, allowing for primal and dual infeasibility to verified in a localized way.

V. Simulations

We begin with a 20×20 mesh topology, which encodes the interconnection between subsystems, and then drop each edge with a probability of 0.15. The resulting interconnection topology is shown in Fig. 3(a) – we assume that all edges are undirected. Note that in general the network may not be strongly connected, but this does not affect the synthesis task. The dynamic interaction between two neighboring subsystems is shown in Fig. 3(b), in which state $x_{i,1}$ can affect state $x_{j,2}$ if i and j are neighbors. This interaction model is inspired from the second-order dynamics of a power network or mechanical system – see our companion paper [16] for a detailed discussion.



(a) Interconnected topology

(b) Interaction between neighboring subsystems

Fig. 3. Simulation Example

The diagonal and off-diagonal entries of A are drawn from a uniform distribution over $[0.4, 0.8]$ and $[-0.4, -0.2] \cup [0.2, 0.4]$, respectively. The open loop plant is unstable, with the spectral radius of A given by 1.2246 in the example that we present. We set $[C_1 \quad D_{12}] = I$, and initially consider the full actuation case of $B_2 = I$.

We assume that the communication constraint $\mathcal{C}$ is such that at time t, sub-controller i can receive $\hat{w}_j[\tau]$ for all $\tau \leq t-k$ if $\text{dist}\,(j \rightarrow i) = k$. The interaction between subsystems illustrated in Fig. 3(b) implies that it takes two time steps for a disturbance at subsystem j to propagate to its neighboring subsystems, and hence the communication speed is twice as fast as propagation speed of disturbances through the plant. Lemma 1 then implies that the LLQR problem (4) is feasible if we choose $\mathcal{L}$ to be a 1-localized constraint, and set the FIR horizon to $T = 2$. However, this only holds true if there is actuator at every subsystem – to allow some flexibility in the actuator regularization task, we expand $\mathcal{L}$ to be a 2-localized constraint and increase the FIR horizon to $T = 20$ – numerical experiments suggest that short FIR horizons T are much more detrimental on closed loop performance and actuator sparsity than small d-localized constraints $\mathcal{L}$.

The optimal LLQR controller implemented using a completely dense actuation architecture achieves a closed loop $\mathcal{H}_2$ norm of 34.59, whereas a centralized LQR controller achieves a closed loop norm of 33.13. Thus with full actuation, there is a 4.4% degradation in performance relative to that achieved by a centralized optimal controller. We then perform the actuator regularization procedure using algorithm (11) with $\lambda_i = 10$ for all i. This procedure identifies 177 actuators that can be removed without affecting the d-localized feasibility of the LLQR control problem (4): the locations of these actuators are shown in Fig. 4.[2] The $\mathcal{H}_2$ norm achieved by the sparsely actuated LLQR controller is 35.91, which is only a 3.8% degradation relative to that achieved by the fully actuated LLQR controller. We emphasize what has just been accomplished: using scalable convex optimization, a sparse actuation architecture has been identified that allows for a d-localized controller to be synthesized and implemented, and achieve near centralized performance.

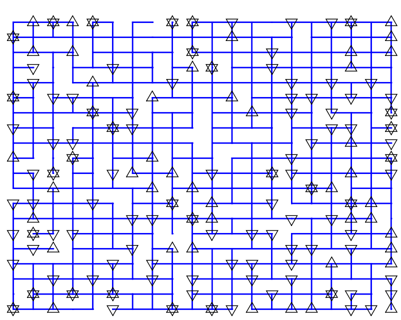


Fig. 4. The upward-pointing triangles represent the subsystems in which the actuator on its first scalar state is removed. The downward-pointing triangles represent the subsystems in which the actuator on its second scalar state is removed.

[2]We take the non-standard approach of showing which actuators are removed, rather than which actuators are selected, because the former is easier to visualize for the two state subsystems considered in this example.

VI. Conclusion

In this paper, we proposed a procedure to design the actuator architecture and spatio-temporal constraints $\mathcal{L} \cap \mathcal{F}_T$ of a LLQR controller. For an LTI distributed system with completely dense actuation and information sharing constraints $\mathcal{C}$ between sub-controllers specified by communication delays, we derive simple conditions on the size d of the d-localized subspace $\mathcal{L}$ and the horizon T of the FIR subspace $\mathcal{F}_T$ to ensure that the resulting LLQR optimal control problem is feasible. We then propose an actuator regularized LLQR control problem to allow for a principled exploration of the tradeoff between actuator density and closed loop performance of the system via convex optimization. We show that an ADMM algorithm can be used to solve this problem in a scalable, distributed and localized manner. Thus the complete procedure offers a scalable way to design a sparse actuation architecture that nonetheless allows for the closed loop system to be d-localized and achieve near-centralized performance. A direct extension of this work is to combine these ideas with those presented in our companion paper [16] to perform simultaneous actuation and sensing architecture design.

References

[1] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *Automatic Control, IEEE Transactions on*, vol. 51, no. 2, pp. 274–286, 2006.

[2] L. Lessard and S. Lall, "Convexity of decentralized controller synthesis," *submitted to IEEE Transaction on Automatic Control*, 2014. [Online]. Available: http://arxiv.org/pdf/1305.5859v2.pdf

[3] ——, "Optimal controller synthesis for the decentralized two-player problem with output feedback," in *2012 IEEE American Control Conference (ACC)*, June 2012.

[4] A. Lamperski and J. C. Doyle, "The $\mathcal{H}_2$ control problem for decentralized systems with delays," *arXiv preprint arXiv:1312.7724v2*, 2014.

[5] A. Lamperski and L. Lessard, "Optimal decentralized state-feedback control with sparsity and delays," *arXiv preprint arXiv:1306.0036v2*, 2014.

[6] Ş. Sabău and N. C. Martins, "Youla-like parametrizations subject to QI subspace constraints," *Automatic Control, IEEE Transactions on*, vol. 59, no. 6, pp. 1411–1422, 2014.

[7] N. Matni, "Distributed control subject to delays satisfying an $\mathcal{H}_\infty$ norm bound," in *2014 53th IEEE Conference on Decision and Control (CDC)*, 2014. [Online]. Available: http://arxiv.org/pdf/1402.1559.pdf

[8] L. Lessard, "State-space solution to a minimum-entropy $\mathcal{H}_\infty$-optimal control problem with a nested information constraint," in *2014 53th IEEE Conference on Decision and Control (CDC)*, 2014. [Online]. Available: http://arxiv.org/pdf/1403.5020v2.pdf

[9] G. Fazelnia, R. Madani, and J. Lavaei, "Convex relaxation for optimal distributed control problem," in *2014 53th IEEE Conference on Decision and Control (CDC)*, 2014.

[10] M. C. Rotkowitz, "Parametrization of stabilizing controllers subject to subspace constraints," in *2011 IEEE American Control Conference (ACC)*, June 2011.

[11] M. Fardad, F. Lin, and M. R. Jovanovic, "Sparsity-promoting optimal control for a class of distributed systems," in *2011 IEEE American Control Conference (ACC)*, June 2011.

[12] F. Lin, M. Fardad, and M. R. Jovanovic, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *Automatic Control, IEEE Transactions on*, vol. 58, no. 9, pp. 2426–2431, 2013.

[13] Y.-S. Wang, N. Matni, S. You, and J. C. Doyle, "Localized distributed state feedback control with communication delays," in *2014 IEEE American Control Conference (ACC)*, June 2014.

[14] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized LQR optimal control," in *2014 53th IEEE Conference on Decision and Control (CDC)*, 2014. [Online]. Available: http://arxiv.org/pdf/1409.6404.pdf

[15] Y.-S. Wang and N. Matni, "Localized distributed optimal control with output feedback and communication delays," in *IEEE 52nd Annual Allerton Conference on Communication, Control, and Computing*, 2014.

[16] ——, "Localized LQG optimal control for large-scale systems," in *submitted to 2016 IEEE American Control Conference (ACC)*, 2016.

[17] N. Matni and V. Chandrasekaran, "Regularization for design," *Submitted to IEEE TAC*, vol. arXiv:1404.1972, 2015.

[18] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

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

[20] M. Rotkowitz, R. Cogill, and S. Lall, "Convexity of optimal control over networks with delays and arbitrary topology," *Int. J. Syst., Control Commun.*, vol. 2, no. 1/2/3, pp. 30–54, Jan. 2010.

[21] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[22] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, July 2009.

[23] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized LQR control with actuator regularization," *Extended report*, 2015. [Online]. Available: cds.caltech.edu/~nmatni/ACC16_LLQRX.pdf