

# Localized LQR Control with Actuator Regularization

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

**Abstract**—This paper discusses the trade-off between actuator placement, controller complexity, and closed loop performance of a localized linear quadratic regulator (LLQR) control scheme on a state feedback localizable system. These are systems for which the effect of each local process disturbance can be localized in closed loop despite communication delays between controllers. Specifically, we formulate the LLQR control problem with actuator regularization to exploit the trade-off between actuator density and closed loop performance, given a fixed LLQR controller complexity. We further show that this regularization problem can be solved in a localized way using distributed optimization. This not only offers a scalable way to perform actuator placement on large network, but also guarantees that the controller for such network can be synthesized and implemented in a localized way.

## I. INTRODUCTION

One fundamental challenge in control system design is balancing the trade-off between closed loop performance and the complexity of controller, including its actuation and sensing interface. This issue becomes one that cannot be ignored for large-scale networked systems such as smart grid, Internet, wireless sensor networks, or biological networks. For these interconnected systems, simple controllers designed completely locally cannot guarantee global stability, whereas a centralized controller is neither scalable to compute nor practical to implement. Many research have addressed this fundamental difficulty and attempted to establish a new controller architecture for networked systems.

The field of distributed (decentralized) optimal control has emerged to incorporate the controller implementation constraint into the design process. The implementation constraints usually manifest themselves as information sharing constraints in the optimal control problem, restricting the controller to respect specific delay and sparsity patterns. It has been shown that a convex reformulation of this constrained optimal control problem exists if and only if the information sharing constraint is quadratically invariant (QI) with respect to the plant [1]–[4]. Following this key idea, various types of distributed constraints [5]–[9] are shown to be convex. For general non-convex implementation constraints, approximation methods also exist [10], [11]. Another framework that incorporates the controller implementation

constraint into design process is on large-scale localizable systems [12]–[14]. One special case is the localized linear quadratic regulator (LLQR) [13] control scheme on state feedback localizable systems. It has been shown that the LLQR controller can be implemented *and* synthesized in a localized and parallel way using only local plant model information.

However, all the methods mentioned above do not incorporate the design of actuation interface. In particular, stabilizable state space realization of the plant model is assumed prior to the synthesis procedure. For a networked system, the locations of the actuators need to be *designed* properly to guarantee stabilizability. This issue has been addressed by the community of network controllability (see [15] and the reference therein). However, most literature in this field ignore the implementation constraint on the distributed controller such as communication delay or sparsity. In addition, the algorithm for actuator placement may not be scalable to large systems.

In this paper, we restrict our discussion on LLQR controller structure and attempt to address the trade-off between actuator placement and closed loop performance. We adopt the regularization for design (RFD) framework [16] into the LLQR problem, and formulate the LLQR control problem with actuator regularization. It should be noted that the trade-off between closed loop performance and the sparsity of a static state feedback controller have been studied in [17], [18].

The rest of this paper is structured as follows. Section II introduces the interconnected system model and the preliminaries on LLQR control. The setup of the LLQR constraints is discussed in Section III. We formulate the LLQR problem with actuator regularization in Section IV, and use the technique of alternative direction method of multipliers (ADMM) [19] to solve this problem in a localized and iterative way. Finally, simulation results are shown in Section V, and conclusions and future research directions are summarized in Section VI.

## II. PRELIMINARIES AND PROBLEM STATEMENT

This section starts with the interconnected system model we considered in this paper. We then use a simple example to introduce the LLQR control scheme, and give the problem statement.

### A. Interconnected System Model

Consider a network of linear systems connected through a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{1, \dots, n\}$  is the set of subsystems, and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  encodes the interaction between

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.

these sub-systems. Each sub-system  $i$  is associated with a state vector  $x_i$  and a control action  $u_i$ . The dynamics of each sub-system  $i$  is given by

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

where  $A_{ii}, A_{ij}, B_{ii}$  are constant matrices with compatible dimensions,  $\delta x_i$  the perturbation on state, and  $\mathcal{N}_i$  the (up-stream) neighboring set of  $i$  defined by  $\mathcal{N}_i = \{j | (j, i) \in \mathcal{E}\}$ . The interconnected system model can be combined into the global plant model as

$$x[k+1] = Ax[k] + B_2u[k] + \delta_x[k] \quad (2)$$

where  $x$ ,  $u$ , and  $\delta_x$  are stacked vectors of local state, control, and process disturbance, respectively. The state space matrices  $(A, B_2)$  can be derived by combining the matrices in (1) in an appropriate way.

*Example 1:* Consider a chain of linear systems given by Fig. 1. The state  $x_i$  of each sub-system is assumed to be a scalar, and each sub-system is assumed to have an actuator. In this case, matrix  $A$  in (2) is a tridiagonal matrix and  $B_2 = I$ .

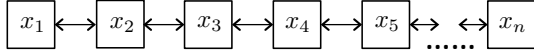


Fig. 1. Chain example

### B. Preliminaries on LLQR

Interconnected systems, even as simple as the chain model in Example 1, can be challenging for optimal controller design. Centralized method is not scalable to large  $n$ , whereas the completely local method cannot guarantee global stability when the effect of sub-system interaction  $A_{ij}$  cannot be ignored. The LLQR control scheme is developed to blend the simplicity of completely local scheme and the superior performance of (unrealistic) centralized scheme.

For a system described by (2), we denote  $R$  the closed loop transfer matrix from  $\delta_x$  to  $x$ ,  $M$  the closed loop transfer matrix from  $\delta_x$  to  $u$ . The LLQR problem is formulated as

$$\underset{\{R, M\}}{\text{minimize}} \quad \|[C_1 \ D_{12}] \begin{bmatrix} R \\ M \end{bmatrix}\|_{\mathcal{H}_2}^2 \quad (3)$$

$$\text{subject to} \quad [zI - A \ -B_2] \begin{bmatrix} R \\ M \end{bmatrix} = I \quad (4)$$

$$\begin{bmatrix} R \\ M \end{bmatrix} \in \mathcal{S} \cap \frac{1}{z} \mathcal{RH}_\infty \quad (5)$$

where  $z$  is the variable of  $z$ -transform,  $\mathcal{RH}_\infty$  the set of real rational stable proper transfer matrices,  $(C_1, D_{12})$  some constant matrices described the cost function, and  $\mathcal{S}$  a spatio-temporal constraint imposed on  $(R, M)$ . Usually, the constraint  $\mathcal{S}$  is imposed to force the closed loop transfer matrices  $(R, M)$  being localized (sparse) finite impulse response (FIR) transfer matrices. Without the constraint  $\mathcal{S}$ , problem (3) - (5) reduces to the traditional LQR scheme. If (3) - (5) is feasible with a localized FIR constraint  $\mathcal{S}$ , we show in [13],

[20], [21] that the LLQR controller achieves the following properties.

- 1) *Localized Closed Loop Response:* As  $(R, M)$  are localized FIRs, each local disturbance only affects the states within a localized region in finite time.
- 2) *Localized Implementation:* The controller achieving the desired closed loop response  $(R, M)$  can be implemented in a localized and scalable way using

$$\begin{aligned} u &= \tilde{M}w^e \\ x_r &= \tilde{R}w^e \\ w^e &= x - x_r \end{aligned} \quad (6)$$

where  $\tilde{R} = (zR - I)$  is a strictly proper FIR transfer matrix and  $\tilde{M} = zM$  a proper FIR transfer matrix.

- 3) *Localized Synthesis:* Problem (3) - (5) can be solved in a localized way using local plant model information.

For localized synthesis, we show in [13], [20] that (3) - (5) admits a column-wise decomposition. This implies that we can analyze the closed loop response for each local disturbance  $\delta x_i$  in an independent and parallel way. With the notion of localized region, each local optimization problem reduces to the form

$$\begin{aligned} \underset{\{x_\ell, u_\ell\}}{\text{minimize}} \quad & \|[C_{1\ell} \ D_{12\ell}] \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix}\|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & [zI - A_\ell \ -B_{2\ell}] \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} = e_\ell \\ & \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} \in \mathcal{S}_\ell \cap \frac{1}{z} \mathcal{RH}_\infty \end{aligned} \quad (7)$$

where  $(x_\ell, u_\ell)$  are local state and control,  $e_\ell$  the initial condition,  $(A_\ell, B_{2\ell})$  the local plant model, and  $(C_{1\ell}, D_{12\ell})$  the local cost matrices, all defined associated with a localized region. We refer this technique as *LLQR decomposition*.

*Example 2:* Consider the chain example with the tridiagonal sparsity constraint on  $R$  and pentadiagonal sparsity constraint on  $M$ . In this case, each local disturbance  $\delta x_i$  can only affect its neighboring states. In addition, (6) suggests that each local control action  $u_i$  can be computed by collecting the estimated disturbance  $w_j^e$  up to its two-hop neighbors, i.e.  $j = i - 2, \dots, i + 2$ . For localized synthesis, consider  $\delta x_1$  in Fig. 2. We define a localized region by  $\text{Region}(x_1) = \{x_1, x_2, x_3\}$  with boundary state  $x_3$ . As long as  $x_3$  is keeping zero for all time, there is no way for the disturbance  $\delta x_1$  to escape from the localized region. We can therefore solve (7) for all the local variables defined associated with this localized region, and reconstruct the first column of  $(R, M)$  in (3) - (5). The closed loop response for each  $\delta x_i$  can be analyzed in a parallel and independent way, even when their localized regions are *overlapping*.

### C. Problem Statement

The spatio-temporal constraint  $\mathcal{S}$  in (5) plays an important role in the LLQR control scheme – all the benefits on localized synthesis and localized implementation rely on the *existence* of a highly sparse  $\mathcal{S}$ . However, our previous works did not discuss on how to find a highly sparse  $\mathcal{S}$  to

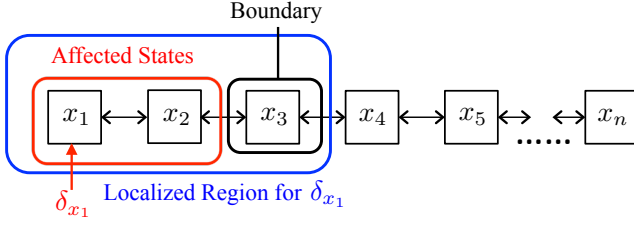


Fig. 2. Localized Region

ensure the feasibility of (4) - (5). In fact, the existence of a sparse  $S$  depends on the actuator placement  $B_2$ . We argue that the locations of actuators ( $B_2$ ) of a networked system should be properly *designed*. This argument also holds true for centralized LQR scheme since randomized actuator placement may not guarantee controllability or stabilizability. The main goal of this paper is to design the trade-off between number of actuators and closed loop performance, all the while maintaining the feasibility of the LLQR problem with a highly sparse  $S$ . In this way, the designed network system can be controlled using sparse actuation associated with a distributed controller that is scalable to implement.

### III. LLQR CONSTRAINT SETUP

In this section, we propose a general design procedure on actuator placement and the spatio-temporal constraint  $S$ . We then focus our discussion on the setup of the constraint  $S$  in the LLQR control scheme.

#### A. General Design Procedure

In this paper, we adopt to the following serial design procedure to design the spatio-temporal constraint  $S$  and the actuator placement  $B_2$ .

**Initialization:** Given a network system (2) with matrix  $A$ , place actuators on all states ( $B_2 = I$ ).

**LLQR Constraint Setup:** Given the communication delay constraint, determine the minimum localized region and the length of FIR such that the LLQR problem is feasible. Relax the localized region and the FIR constraint when necessary to allow more flexibility on actuator placement. This is the initial setup of the LLQR controller complexity ( $S$ ).

**Actuator Regularization:** Given the LLQR problem with the designed spatio-temporal constraint  $S$ , use regularization to exploit the trade-off between closed loop performance and the number of actuators (number of non-zero columns in  $B_2$ ).

This serial design procedure is proposed due to its simplicity. It can be extended to more complicated procedure, such as iteratively design the actuator placement ( $B_2$ ) and the controller complexity ( $S$ ). The setup of the LLQR constraint is discussed in the rest of this section, and the actuator regularization is discussed in Section IV. After finishing the design of spatio-temporal constraint  $S$  and actuator locations  $B_2$ , we can use the method in [13] to synthesize the LLQR controller.

Note that we assume full actuation  $B_2 = I$  at the initialization stage. This has two advantages: (i) the system

( $A, B$ ) is always controllable, and the LLQR problem is feasible if the communication speed is fast enough, and (ii) this gives more flexibility on actuator placement using regularization. We can also initialize  $B_2$  to other values if necessary, as long as the LLQR problem is feasible with a sparse  $S$ .

#### B. Setup of LLQR Constraint

Recall that  $S$  in the LLQR problem is a spatio-temporal constraint on the closed loop transfer matrix. We use the term *spatial localizability* to denote the spatial part of the constraint  $S$  (the sparsity of  $S$ ). On the other hand, the temporal part of the constraint  $S$  is just the common notion of controllability of ( $A, B_2$ ) [21].

Denote  $S_j$  the  $j$ -th (block) column of the constraint  $S$ . By definition,  $S_j$  is the spatio-temporal constraint imposed on the closed loop response from local disturbance  $\delta x_j$  to  $x$  and  $u$ . Therefore, the (block) row-wise sparsity pattern of  $S_j$  encodes the information of affected states, activated control action, and the localized region for local disturbance  $\delta x_j$ . The size of the minimum localized region for each local disturbance is primarily determined by the communication delay constraints. For an extreme case, if the communication speed is slower than the speed of disturbance propagation in the plant, then there is no way to localize the effect of a local disturbance. It is therefore unlikely to find a sparse feasible  $S$  for a plant with strongly connected topology.

The construction of a localized region depends on the topology of the network system. For  $B_2 = I$ , we can treat each single scalar state as a single sub-system. For a network system (2) with matrix  $A$ , we define the state transition graph as follows.

**Definition 1:** The state transition graph of a matrix  $A$  with dimension  $a$  is a directed graph  $\mathcal{G}_A = (\mathcal{V}_A, \mathcal{E}_A)$  with  $\mathcal{V}_A = \{v_1, \dots, v_a\}$ , and  $(v_i, v_j) \in \mathcal{E}_A$  if and only if  $A_{ji} \neq 0$ .<sup>1</sup>

The distance between two vertices  $v_i$  and  $v_j$  in a graph  $\mathcal{G}_A$  is the length of a shortest path from  $v_i$  to  $v_j$ , and is denoted by  $\text{dist}(v_i, v_j)$ . For the localized region, we assume that each local disturbance at  $v_i$  is only allowed to affect up to its  $d$ -hop (downstream) neighbors, i.e.  $v_i$  can affect  $\{v_j | \text{dist}(v_i, v_j) \leq d\}$  only. If such localized region constraint is feasible for each local disturbance, we say that the full actuated system is  $d$ -spatially localizable. For the delay patterns, we assume that each local controller takes  $t_s$  time to sense the state deviation,  $t_c$  time to transmit the information to its neighbor, and  $t_a$  time to actuate on the state. The parameter  $(t_s, t_c, t_a)$  are normalized with respect to the sampling time of the discrete time system, so they are not integers in general. If the information is received between two sampling interval, we use the information in the next sampling time. The relation between delay pattern and localized region is described by the following lemma.

**Lemma 1:** Assume that the communication network topology is the same as the plant topology  $\mathcal{G}_A$ , but with

<sup>1</sup>The  $A_{ji}$  here denotes the  $(j, i)$ -entry of  $A$ .

a faster communication speed, i.e.  $t_c < 1$ . A full actuated system is  $d$ -spatially localizable if  $\frac{t_s+t_a}{1-t_c} - 1 \leq d$ .

*Proof:* Consider a local disturbance at  $v_i$ . To make a system being  $d$ -spatially localizable, only the states in the set  $\{v_j | \text{dist}(v_i, v_j) \leq d\}$  can be affected. In other words, the states in the set  $\{v_j | \text{dist}(v_i, v_j) > d\}$  should be zero for all time. As each state can be directly actuated, we only need to ensure that the states in the boundary set  $\{v_j | \text{dist}(v_i, v_j) \leq d+1\} \setminus \{v_j | \text{dist}(v_i, v_j) \leq d\}$  are zero for all time. The effect of disturbance takes  $(d+1)$  time steps to reach the boundary through plant propagation. As the communication network topology follows the topology of the plant, information about the state deviation is transmitted to the boundary at time  $t_s + (d+1)t_c$ . If  $t_s + (d+1)t_c + t_a \leq d+1$ , then the action for the disturbance can be taken to cancel the effect of disturbance at the boundary. This argument holds for every local disturbance  $v_i$ . Therefore,  $d$ -spatial localizability is guaranteed. ■

*Remark 1:* This condition is similar but different from the delay interpretation of QI in [2]. Using the delay pattern  $(t_s, t_c, t_a)$ , the delay condition for QI is  $p * t_c \leq p + t_a + t_s$  where  $p$  is the distance between any pair of states. When  $t_c = 1$ , the delay pattern is QI, but the effect of disturbance cannot be spatially localized. Therefore, in terms of communication speed, our control scheme is more restricted than QI. However, LLQR only requires *local* communication when  $t_c < 1$ . On the contrary, QI scheme requires global communication for any  $t_c$  for a plant with strongly connected topology, which is not scalable.

In a common situation, we have  $t_s = 0$  and  $t_a = 1$ . Let  $h = \frac{1}{t_c}$  be the relative communication speed between neighboring controllers, the condition in Lemma 1 reduces to  $\frac{1}{h-1} \leq d$ . As we assume full actuation, each localized region is controllable. Using a similar argument to Lemma 1, we can show that the effect of each local disturbance can be eliminated within time step  $T \geq d+1$ .

We can use the information of localized region, communication delay, and the FIR constraint to construct a feasible  $\mathcal{S}$  for the LLQR problem. 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. More details on the operation of binary matrices are included in Appendix for completeness. Using the parameter  $(d, T, h)$ , a spatio-temporal constraint for closed loop transfer matrices  $(R, M)$  can be constructed by

$$\begin{aligned} \mathcal{S}_R &= \sum_{i=1}^T \frac{1}{z^i} \bar{\mathcal{A}}^{\min(d, \lfloor h(i-1) \rfloor)} \\ \mathcal{S}_M &= \sum_{i=1}^T \frac{1}{z^i} \bar{\mathcal{A}}^{\min(d+1, \lfloor h(i-1) \rfloor)}. \end{aligned} \quad (8)$$

For a full actuated system, the LLQR constraint  $\mathcal{S} = [\mathcal{S}_R; \mathcal{S}_M]$  is feasible for  $h > 1$ ,  $d \geq \frac{1}{h-1}$ , and  $T \geq d+1$ .

We can increase  $(d, T, h)$  to achieve a better closed loop performance. Usually, the communication speed  $h$  is limited by the physical constraint. As long as the speed is physically

implementable, we should set  $h$  larger to make  $d$  smaller. The parameter  $T$  dominates the trade-off between settling time and transient performance (including  $\mathcal{H}_2$  norm or maximum overshoot). It also affects the length of FIR in LLQR controller implementation (6). The parameter  $d$  should be kept as small as possible since it dominates the complexity of controller synthesis and implementation. To make LLQR effective, we usually increase  $d$  from its minimum by only 1 or 2 to allow some flexibility on actuator regularization.

#### IV. ACTUATOR REGULARIZATION

With a construction of a feasible spatio-temporal constraint  $\mathcal{S}$ , we can exploit the trade-off between closed loop performance and the number of actuators. This completes the second part of the general design procedure.

##### A. Problem Formulation

Recall that the LLQR controller is implemented by (6). If  $M$  has a zero row  $e_i^\top M = 0$ , then the control action  $u_i$  is zero for all time. We can then remove the actuator on  $u_i$  during implementation. If we want to construct a control rule with at most  $r$  control actions being activated, we can find a transfer matrix  $M$  with at most  $r$  nonzero rows. This leads to the following problem

$$\begin{aligned} &\underset{\{R, M\}}{\text{minimize}} && \| [C_1 \quad D_{12}] \begin{bmatrix} R \\ M \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ &\text{subject to} && (4) - (5) \\ &&& N_r(M) \leq r \end{aligned} \quad (9)$$

where  $N_r(M)$  represents the number of nonzero rows of  $M$ . Problem (9) is a combinatorial optimization problem due to the last constraint, and is generally very hard to solve. Therefore, we adopt the technique of regularization in hope of finding a row-wise sparse  $M$ . Specifically, we formulate the LLQR problem with actuator regularization as

$$\begin{aligned} &\underset{\{R, M\}}{\text{minimize}} && \| [C_1 \quad D_{12}] \begin{bmatrix} R \\ M \end{bmatrix} \|_{\mathcal{H}_2}^2 + \|M\|_{\mathcal{A}} \\ &\text{subject to} && (4) - (5) \end{aligned} \quad (10)$$

where  $\|\cdot\|_{\mathcal{A}}$  is the actuator norm introduced in [16]. One choice of actuator norm is given by

$$\|M\|_{\mathcal{A}} = \sum_{i=1}^{n_u} \lambda_i \|e_i^\top M\|_{\mathcal{H}_2}, \quad (11)$$

where  $\lambda_i$  is the relative price of each actuator. When  $\lambda_i = 1$  for all  $i$ , (11) is equivalent to the  $\ell_1/\ell_2$  norm (group lasso [22] in the statistical learning literature). Equation (10) is a convex optimization problem. Moreover, using the actuator regularizer (11), we may reconstruct a solution of (9) under some technical conditions. The parameter  $\{\lambda_i\}$  can be used to exploit the tradeoff between closed loop performance (the square of  $\mathcal{H}_2$  norm in (10)) and the number of actuator (approximated by the actuator norm in (10)), given a fixed controller complexity (the  $\mathcal{S}$  in the constraint).

Although the regularization problem (10) is convex, it is still a large-scale optimization problem. However, we can use

the ADMM algorithm to decompose (10) into several local optimization problems and solve them iteratively. This offers a scalable algorithm to do actuator placement and LLQR controller design for large-scale systems.

### B. ADMM Algorithm

Assume that the spatio-temporal constraint  $\mathcal{S}$  is a localized FIR, so LLQR decomposition described in Section II holds. We define the extended-real-value functions  $f(R, M_1)$ ,  $g(M_2)$  by

$$f(R, M_1) = \begin{cases} (3) & \text{if (4), (5)} \\ \infty & \text{otherwise} \end{cases}$$

$$g(M_2) = \begin{cases} \|M\|_{\mathcal{A}} & \text{if (5)} \\ \infty & \text{otherwise} \end{cases}$$

Problem (10) can then be equivalently formulated as

$$\begin{aligned} & \underset{\{R, M_1, M_2\}}{\text{minimize}} && f(R, M_1) + g(M_2) \\ & \text{subject to} && M_1 = M_2. \end{aligned} \quad (12)$$

Problem (12) can be solved via the standard ADMM approach [19] as

$$(R^{k+1}, M_1^{k+1}) = \underset{R, M_1}{\text{argmin}} \left( f(R, M_1) + \frac{\rho}{2} \|M_1 - M_2^k + \Lambda^k\|_{\mathcal{H}_2}^2 \right) \quad (13a)$$

$$M_2^{k+1} = \underset{M_2}{\text{argmin}} \left( g(M_2) + \frac{\rho}{2} \|M_2 - M_1^{k+1} - \Lambda^k\|_{\mathcal{H}_2}^2 \right) \quad (13b)$$

$$\Lambda^{k+1} = \Lambda^k + M_1^{k+1} - M_2^{k+1} \quad (13c)$$

where the square of  $\mathcal{H}_2$  norm of a transfer matrix can be calculated by summing the square of Frobenius norm over all its spectral components. Subproblem (13a) is a modified LLQR problem, which can also be solved via a column-wise decomposition on the objective and the constraint. For the actuator norm in (11), subproblem (13b) can be solved using a row-wise decomposition. Equation (13c) can be decomposed either column-wise or row-wise. The only variable that need to be shared globally before synthesis is  $\rho$  in (13a) - (13b). Other than this variable, (13a) - (13c) is a localized algorithm to solve the LLQR problem with actuator regularization using column-wise and row-wise decomposition alternatively and iteratively.

In the following, we show some convergence properties of (13a) - (13c), and design the stopping criteria for the algorithm. We then derive the analytic solution of (13a) and (13b). Specifically, the solution of (13a) can be written as an affine function of its reference solution. On the other hand, the solution of (13b) is given by a vectorial soft-thresholding [23] on its reference solution. In other words, we only need to solve the update rule in (13a) *once*, and then update (13a) - (13c) using the closed form solution. This offers a fast iterative algorithm solving LLQR problem with actuator regularization.

### C. Convergence and Stopping Criteria

Adding the actuator regularizer in (10) does not change the feasibility of the LLQR problem, so (10) has an optimal solution  $(R^*, M^*)$ . Assume that  $D_{12}^\top D_{12} > 0$ . In this case, the objective function in (10) is strongly convex with respect to  $M$ , and the optimal solution  $M^*$  is unique. As  $f$  and  $g$  are closed, proper, and convex, we have strong duality and (10) satisfies the convergence condition in [19]. From [19], the objective function in (10) converges to its optimal value. As the objective function in (10) is a continuous function of  $M$  and the optimal solution  $M^*$  is unique, we then have primal variable convergence  $M_1^k \rightarrow M^*$  and  $M_2^k \rightarrow M^*$ . For the convergence of  $R$ , consider the equation

$$(zI - A)(R^k - R^*) = B_2(M^k - M^*).$$

As the right-hand-side converges to zero for large  $k$ , the left-hand-side also converges to zero. Let  $\Delta^k = R^k - R^*$ , we then have  $\Delta^k[1] \rightarrow 0$  and  $\Delta^k[i+1] - A\Delta^k[i] \rightarrow 0$  for all  $i$ . In addition, the constraint in (13a) ensures that  $\Delta^k$  is stable, i.e.  $\Delta^k[i] \rightarrow 0$  for large  $i$  for all  $k$ . Combining these two facts, we can show that each spectral component of  $\Delta^k$  converges to 0 for large  $k$  using induction. This implies  $R^k \rightarrow R^*$ .

The stopping criteria is designed from [19], in which we use  $\|M_1^k - M_2^k\|_{\mathcal{H}_2}$  as primal infeasibility and  $\|M_2^k - M_2^{k-1}\|_{\mathcal{H}_2}$  as dual infeasibility. The algorithm (13a) - (13c) terminates when the primal and dual infeasibility are smaller than the feasibility tolerances  $\epsilon^{pri}$  and  $\epsilon^{dual}$ , respectively. Note that the square of primal and dual infeasibility can be calculated in a localized way as well.

Although the localized algorithm (13a) - (13c) always converges to an optimal solution of the regularization problem (10), the optimal solution of (10) may not necessary be row-wise sparse. However, under some technical conditions [16], (10) may recover a sparse solution in (9).

### D. Analytic Solution

In this subsection, we solve the analytic solution of (13a) and (13b). Problem (13a) is almost identical to the LLQR problem (3) - (5), except that we add a quadratic penalty on the distance between  $M_1$  and its reference  $M_2^k - \Lambda^k$ . We can then perform a column-wise LLQR decomposition on (13a) and use the notion of localized region to reduce the dimension of the problem. In the reduced dimension, the optimization problem has the form

$$\begin{aligned} & \underset{\{x_\ell, u_\ell\}}{\text{minimize}} && \| [C_{1\ell} \quad D_{12\ell}] \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} \|_{\mathcal{H}_2}^2 + \frac{\rho}{2} \|u_\ell - u_\ell^r\|_{\mathcal{H}_2}^2 \\ & \text{subject to} && \text{constraints in (7)} \end{aligned} \quad (14)$$

where  $u_\ell^r$  can be computed from  $M_2^k - \Lambda^k$  using the notion of (column-wise) localized region. Problem (14) is a finite dimensional affinely constrained quadratic program that can be solved analytically due to the FIR constraint. Specifically, the optimal solution can be written as an affine function of its reference as

$$\begin{bmatrix} \bar{x}_\ell^* \\ \bar{u}_\ell^* \end{bmatrix} = F_a \bar{u}_\ell^r + F_b \quad (15)$$

where the bar symbol represents the stacked vector of all its spectral components, i.e.  $\bar{u}_\ell^r = [u_\ell^r[1]^\top \dots u_\ell^r[T]^\top]^\top$ , etc. Once the constant update matrices  $F_a$  and  $F_b$  are computed, we can use (15) to update the solution in (13a), which accelerates the algorithm significantly.

For problem (13b), we can perform a row-wise decomposition on the objective function and the constraint. If (5) is a sparse FIR constraint, each row-wise decomposition of (13b) eventually leads to an unconstrained optimization problem

$$\underset{\bar{u}_\ell}{\text{minimize}} \quad \lambda_i \|\bar{u}_\ell\|_{\ell_2} + \frac{\rho}{2} \|\bar{u}_\ell - \bar{u}_\ell^r\|_{\ell_2}^2 \quad (16)$$

where  $\bar{u}_\ell^r = [u_\ell^r[1]^\top \dots u_\ell^r[T]^\top]^\top$ , and  $u_\ell^r$  can be derived from its reference solution  $M_1^{k+1} + \Lambda^k$  using (row-wise) localized region. The analytic solution to (16) can be expressed by a vectorial soft-thresholding [23] as

$$\bar{u}_\ell^* = (1 - \frac{\lambda_i/\rho}{\|\bar{u}_\ell^r\|_{\ell_2}})_+ \bar{u}_\ell^r \quad (17)$$

in which  $(c)_+ = \max(0, c)$  for any scalar  $c$ . If the actuator norm in (11) is chosen to be the  $\ell_1/\ell_\infty$  norm, efficient algorithm [24] also exists to calculate the solution.

We summarize the ADMM algorithm for LLQR with actuator regularization in Algorithm 1.

---

**Algorithm 1:** LLQR with Actuator Regularization

---

Given global plant model  $(A, B_2)$ , objective  $(C_1, D_{12})$ , and sparse FIR constraint  $\mathcal{S}$ ;

**for** each sub-system  $x_j$  **do**

Define column-wise localized region by  $\mathcal{S}_j$ ;  
Derive update matrices  $F_a$  and  $F_b$  within its localized region;

**if** All LLQR sub-problems are feasible **then**

**while** Stopping criteria not met **do**

**for** each column-wise localized region **do**

Compute reference  $u_\ell^r$ ;  
Update  $x_\ell, u_\ell$  using  $F_a, F_b, x_\ell^r, u_\ell^r$ ;  
Distribute  $x_\ell, u_\ell$  within its localized region;

**for** each row-wise localized region **do**

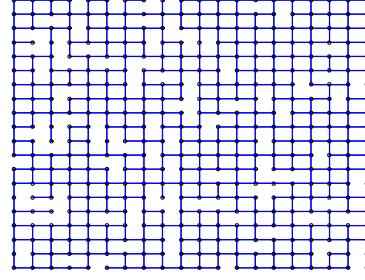
Compute reference  $u_\ell^r$ ;  
Update  $u_\ell$  using (17);  
Update  $\Lambda$  in (13c);  
Distribute  $u_\ell, \Lambda$  within its localized region;

---

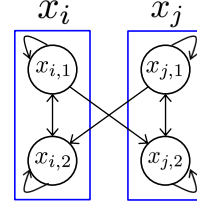
## V. SIMULATIONS

We start with a  $20 \times 20$  mesh topology representing the interconnection between sub-systems, and drop each edge with probability 0.15. The resulting interconnected topology is shown in Fig. 3(a). Each sub-system  $x_j$  is assumed to have two interacting states  $x_{j,1}$  and  $x_{j,2}$ . The interaction between two neighboring sub-systems is shown in Fig. 3(b), where state  $x_{i,1}$  can affect state  $x_{j,2}$  if  $i$  and  $j$  are neighbors.

The entries of the  $A$  matrix is randomly generated by the following rule. The diagonal entries are uniformly distributed



(a) Interconnected topology



(b) Interaction between neighboring sub-systems

Fig. 3. Simulation Example

in  $[0.4, 0.8]$ , and all other entries are uniformly distributed in  $[-0.4, -0.2] \cup [0.2, 0.4]$ . The instability of the plant can be characterized by the spectral radius of the matrix  $A$ , which is 1.2246 in our example. We set  $B_2 = I$  and  $[C_1 \ D_{12}] = I$ .

We assume that it takes one sampling time to transmit the measurement to a neighboring sub-system. For instance, sub-system  $u_i[t]$  can access  $x_j[\tau]$  for  $\tau \leq t - k$  if  $(i, j)$  are  $k$ -hop neighbors. As the disturbance takes two steps to propagate to its neighboring sub-systems, the communication speed is twice faster than the speed of disturbance propagation. Applying Lemma 1 with this delay constraint, we know that there exists a control rule such that each local disturbance only affects its direct neighbors in closed loop. To enhance flexibility on actuator regularization, we allow each local disturbance to affect up to its two-hop neighbors. This means that each sub-system needs to communicate up to its three-hop neighbors during implementation, and needs to use the plant model up to its three-hop neighbors for localized synthesis and actuator regularization. Lastly, we impose the FIR constraint with length  $T = 20$  on all closed loop transfer matrices.

When all states are actuated, the  $\mathcal{H}_2$  norm of the LLQR control is 34.59, while the centralized LQR scheme is 33.13. In terms of the  $\mathcal{H}_2$  norm, our control scheme has 4.4% degradation. We then perform actuator regularization with  $\lambda_i = 10$  for all  $i$ . After actuator regularization, there are 177 actuators with actuator norm smaller than 0.01. The locations of these actuators are shown in Fig. 4. The LLQR problem remains feasible when these actuators are removed. This means that we can still control this network using sparse actuation with a scalable controller implementation. The  $\mathcal{H}_2$  norm of the LLQR scheme for the new system is 35.91, and the norm of the centralized scheme is 34.67. The degradation between the centralized and the localized scheme is 3.6%.



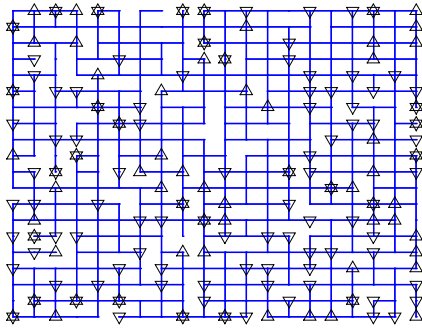


Fig. 4. The upward-pointing triangles represent the sub-systems in which the actuator on its first scalar state is removed. The downward-pointing triangles represent the sub-systems in which the actuator on its second scalar state is removed.

The actuator locations are determined by various factors such as the local dynamics, the cost function for each sub-system, and the price of actuator installation ( $\lambda_i$ ). Generally speaking, regularization tends to remove actuators at locations with stable local dynamics, smaller penalty on state deviation, and higher price on actuator installation. This coincides with our intuition to remove actuators that are less necessary. In this example, the cost function and the price of actuator installation are set to be the same among all sub-system. Therefore, the actuator locations are dominated by the effect of local dynamics (randomness in matrix  $A$ ).

## VI. CONCLUSION

In this paper, we discussed the trade-off between actuator density, controller complexity, and closed loop performance of the localized LQR (LLQR) control scheme. Specifically, we formulated the LLQR control problem with actuator regularization to deal with the trade-off between the number of actuators and closed loop performance, given a fixed LLQR controller complexity. We also shown that this algorithm can be solved in a localized way using distributed optimization. The complete procedure offers a scalable way to perform actuator placement on large network. In addition, the actuator placement also guarantees that the LLQR controller for such network can be synthesized and implemented in a localized way.

In the future, we aim to extend this framework to output feedback localizable system to perform both actuator and sensor placement.

## APPENDIX

Denote  $\cup$  the OR operator,  $\cap$  the AND operator on binary matrices. The addition of two binary matrices  $S_1, S_2 \in \{0, 1\}^{m \times n}$  is defined by

$$(S_1 + S_2)_{ij} = (S_1)_{ij} \cup (S_2)_{ij}.$$

The product  $S_1 = S_2 S_3$  with binary matrices of compatible dimension is defined by the rule

$$(S_1)_{ij} = \bigcup_k [(S_2)_{ik} \cap (S_3)_{kj}].$$

where the  $\cup$  operator is taken over all possible  $k$ . For a square binary matrix  $S_0$ , we define the power  $S_0^{i+1} := S_0^i S_0$  for all positive integer  $i$ , and  $S_0^0 = I$ .

The spatio-temporal constraint for a transfer function  $R$  is described by a constraint space  $\mathcal{S}_R := \sum_{i=0}^{\infty} \frac{1}{z^i} \mathcal{S}_R[i]$ , where  $\{\mathcal{S}_R[i]\}$  is an ordered set of binary matrices. We say that a transfer function  $R$  satisfies a spatio-temporal constraint  $\mathcal{S}_R$  if and only if  $\text{sp}(R[i]) \subseteq \mathcal{S}_R[i]$  for all  $i$ , which we denote by  $R \in \mathcal{S}_R$ .

## 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] 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. [Online]. Available: <http://dx.doi.org/10.1504/IJSCC.2010.031157>
- [3] 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>
- [4] 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.
- [5] A. Lamperski and J. C. Doyle, "Output feedback  $\mathcal{H}_2$  model matching for decentralized systems with delays," in *2013 IEEE American Control Conference (ACC)*, June 2013.
- [6] —, "The  $\mathcal{H}_2$  control problem for decentralized systems with delays," *arXiv preprint arXiv:1312.7724v2*, 2014.
- [7] A. Lamperski and L. Lessard, "Optimal decentralized state-feedback control with sparsity and delays," *arXiv preprint arXiv:1306.0036v2*, 2014.
- [8] P. Shah and P. A. Parrilo, "An optimal controller architecture for poset-causal systems," in *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.
- [9] L. Lessard and S. Lall, "Optimal controller synthesis for the decentralized two-player problem with output feedback," in *2012 IEEE American Control Conference (ACC)*, June 2012.
- [10] 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.
- [11] M. C. Rotkowitz, "Parametrization of stabilizing controllers subject to subspace constraints," in *2011 IEEE American Control Conference (ACC)*, June 2011.
- [12] 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.
- [13] 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>
- [14] 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.
- [15] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *Control of Network Systems, IEEE Transactions on*, vol. 1, no. 1, pp. 40–52, March 2014.
- [16] N. Matni and V. Chandrasekaran, "Regularization for design," *CoRR, 2014 IEEE Conference on Decision and Control*, vol. arXiv:1404.1972, 2014. [Online]. Available: <http://arxiv.org/abs/1404.1972v2.pdf>
- [17] 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.
- [18] 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.
- [19] 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.

- [20] Y.-S. Wang and N. Matni, "Localized LQG optimal control for large-scale systems," in *submitted to 2015 54th IEEE Conference on Decision and Control (CDC)*, 2015.
- [21] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized distributed control: An optimal control theory for large-scale systems," *In preparation*, 2015.
- [22] 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.
- [23] 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.
- [24] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the L1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. ACM, 2008, pp. 272–279.